



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**APLICACIÓN DE LA METODOLOGÍA DE LA EMPRESA TATA
CONSULTANCY SERVICES A LAS ÁREAS DE BASES DE
DATOS, INTELIGENCIA ARTIFICIAL E INNOVACIÓN DE LA
CARRERA DE CIENCIAS Y SISTEMAS, DE LA FACULTAD DE
INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE
GUATEMALA**

Framklym Adolfo Algaba Gutiérrez

Asesorado por el Ing. Jorge Armín Mazariegos

Guatemala, noviembre de 2009.

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**APLICACIÓN DE LA METODOLOGÍA DE LA EMPRESA TATA
CONSULTANCY SERVICES A LAS ÁREAS DE BASES DE DATOS,
INTELIGENCIA ARTIFICIAL E INNOVACIÓN DE LA CARRERA DE
CIENCIAS Y SISTEMAS, DE LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN DE EPS

PRESENTADO A LA JUNTA DIRECTIVA
DE LA FACULTAD DE INGENIERÍA

POR

FRAMKLYM ADOLFO ALGABA GUTIÉRREZ
ASESORADO POR EL ING. JORGE ARMÍN MAZARIEGOS
AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2009.

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. José Milton De León Bran
VOCAL V	Br. Isaac Sultán Mejía
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADORA	Inga. Floriza Felipa Ávila Pesquera de Medinilla
EXAMINADOR	Ing. Marlon Antonio Pérez Turk
EXAMINADORA	Inga. Sonia Yolanda Castañeda Ramírez
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de Ejercicio Profesional Supervisado (EPS) titulado:

**APLICACIÓN DE LA METODOLOGÍA DE LA EMPRESA TATA
CONSULTANCY SERVICES A LAS ÁREAS DE BASES DE DATOS,
INTELIGENCIA ARTIFICIAL E INNOVACIÓN DE LA CARRERA DE
CIENCIAS Y SISTEMAS, DE LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA,**

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, en agosto 2008.

Framklym Adolfo Algaba Gutiérrez.



Guatemala, 11 de septiembre 2009

Ingeniera
Norma Ileana Sarmiento
Directora Unidad EPS
Dirección de EPS
Facultad de Ingeniería
USAC

Respetable Ingeniera Sarmiento:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **FRANKLYM ADOLFO ALGABA GUTIÉRREZ**, titulado: **"Aplicación de la metodología de la empresa TATA CONSULTANCY SERVICES a las áreas de Bases de Datos, Inteligencia Artificial e Innovación de la carrera de Ingeniería en Ciencias y Sistemas, de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala"**, y a mi criterio, el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Agradeciendo su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Jorge Armin Mazariegos
Catedrático
Asesor del Proyecto
Escuela de Ciencias y Sistemas

Jorge Armin Mazariegos Rufanales
ING. EN CIENCIAS Y SISTEMAS
M.C. ADMINISTRADOR DE TECNOLOGIA
COLEGIO 33-17

Universidad de San Carlos de Guatemala
Facultad de Ingeniería



UNIDAD DE E.P.S.

Guatemala, 29 de septiembre de 2009.
Ref.EPS.DOC.1380.09.09.

Inga. Norma Ileana Sarmiento Zeceña de Serrano
Directora Unidad de EPS
Facultad de Ingeniería
Presente

Estimada Ingeniera Sarmiento Zeceña.

Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) del estudiante universitario de la Carrera de Ingeniería en Ciencias y Sistemas, **Framklym Adolfo Algaba Gutiérrez** Carné No. **200320723** procedí a revisar el informe final, cuyo título es **“APLICACION DE LA METODOLOGÍA DE LA EMPRESA TATA CONSULTANCY SERVICES A LAS ÁREAS DE BASES DE DATOS, INTELIGENCIA ARTIFICIAL E INNOVACIÓN DE LA CARRERA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**.

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

“Id y Enseñad a Todos”

Inga. Floriza Felipa Ávila Pesquera de Medinilla
Supervisora de EPS
Área de Ingeniería en Ciencias y Sistemas

FFAPdM/RA



Edificio de E.P.S., Facultad de Ingeniería, Universidad de San Carlos de Guatemala
Ciudad Universitaria zona 12, teléfono directo: 2442-3509

Universidad de San Carlos de Guatemala
Facultad de Ingeniería



UNIDAD DE E.P.S.

Guatemala, 29 de septiembre de 2009.
Ref.EPS.D.625.09.09.

Ing. Marlon Antonio Pérez Turck
Director Escuela de Ingeniería Ciencias y Sistemas
Facultad de Ingeniería
Presente

Estimado Ingeniero Perez Turck.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **“APLICACION DE LA METODOLOGÍA DE LA EMPRESA TATA CONSULTANCY SERVICES A LAS ÁREAS DE BASES DE DATOS, INTELIGENCIA ARTIFICIAL E INNOVACIÓN DE LA CARRERA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, que fue desarrollado por el estudiante universitario **Franklym Adolfo Algaba Gutiérrez** Carné No. **200320723** quien fue debidamente asesorado por el Ing. **Jorge Armin Mazariegos** y supervisado por la Inga. **Floriza Felipa Ávila Pesquera de Medinilla**

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor y de la Supervisora de EPS, en mi calidad de Directora apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

“Id y Enseñad a Todas”


Inga. Norma Ileana Zecena
Directora Unidad de EPS
Unidad de Prácticas de Ingeniería y EPS
Facultad de Ingeniería

Edificio de E.P.S., Facultad de Ingeniería, Universidad de San Carlos de Guatemala
Ciudad Universitaria zona 12, teléfono directo: 2442-3509



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 07 de Octubre de 2009

Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **FRANKLYM ADOLFO ALGABA GUTIERREZ**, titulado: **"APLICACIÓN DE LA METODOLOGIA DE LA EMPRESA TATA CONSULTANCY SERVICES A LAS AREAS DE BASES DE DATOS, INTELIGENCIA ARTIFICIAL E INNOVACION DE LA CARRERA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



E
S
C
U
E
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado "APLICACIÓN DE LA METODOLOGÍA DE LA EMPRESA TATA CONSULTANCY SERVICES A LAS ÁREAS DE BASES DE DATOS, INTELIGENCIA ARTIFICIAL E INNOVACIÓN DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA", presentado por el estudiante FRANKLYM ADOLFO ALGABA GUTIERREZ, aprueba el presente trabajo y solicita la autorización del mismo.

"ID Y ENSEÑAD A TODOS"



Ing. Marlon Antonio Pérez Turk
Director, Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 16 de noviembre 2009



Ref. DTG.509.09

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **APLICACIÓN DE LA METODOLOGÍA DE LA EMPRESA TATA CONSULTANCY SERVICES A LAS ÁREAS DE BASES DE DATOS, INTELIGENCIA ARTIFICIAL E INNOVACIÓN DE LA CARRERA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por el estudiante universitario **Franklym Adolfo Algaba Gutiérrez**, autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olimpo Paiz Recinos
DECANO



Guatemala, noviembre de 2009

/cc
c.c. archivo.

ACTO QUE DEDICO A:

Mi familia:

Ivania Gutiérrez, Teresa Gutiérrez y Fernando Algaba, mis más sinceras gracias por su apoyo incondicional durante el desarrollo de mi carrera; este logro está dedicado a ustedes que lo hicieron posible.

AGRADECIMIENTOS A:

Mi asesor:

Ing. Armín Mazariegos, por el tiempo dedicado y los consejos aportados al proyecto.

La Universidad San Carlos de Guatemala, Facultad de Ingeniería:

Por haber mantenido durante toda mi carrera un alto nivel de educación.

Colaboradores:

A todas las personas que hicieron posible la realización de este proyecto, al proveer sus consejos o al participar en el proyecto.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO.....	IX
RESUMEN.....	XI
OBJETIVOS.....	XIII
INTRODUCCIÓN.....	XV
1. MARCO TEÓRICO	
1.1 Principios de Pedagogía.....	01
1.2 Proceso de enseñanza – aprendizaje.....	03
1.3 Método de clases magistrales.....	05
1.4 Aprendizaje práctico por medio de Laboratorios.....	07
2. GUÍA DEL INSTRUCTOR	
2.1 Definición de una guía del instructor.....	09
2.2 Objetivos de una guía del instructor.....	10
2.3 Elementos de la guía del instructor.....	11
2.4 Proceso de construcción de la guía del instructor.....	12
2.5 Modo de uso de la guía del instructor.....	15
3. GUÍA DEL INSTRUCTOR DEL LABORATORIO DE SISTEMAS DE BASES DE DATOS 1	
3.1 Datos generales.....	17
3.2 Distribución del curso.....	18
3.3 Evaluación	18

3.4	Detalle de sesiones	19
3.5	Distribución de sesiones.....	19
3.6	Detalle de <i>lab assignments</i>	30
3.7	Detalle de proyectos.....	33
3.8	Detalle de exámenes.....	35
3.9	Examen de ejemplo.....	36
3.10	Bibliografía sugerida.....	37
3.11	<i>Lab handbook</i>	37

4. GUÍA DEL INSTRUCTOR DEL LABORATORIO DE SISTEMAS DE BASES DE DATOS 2

4.1	Datos generales.....	45
4.2	Distribución del curso.....	46
4.3	Evaluación.....	46
4.4	Detalle de sesiones.....	47
4.5	Distribución de sesiones.....	48
4.6	Detalle de <i>lab assignments</i>	56
4.7	Detalle de proyectos.....	60
4.8	Detalle de exámenes.....	61
4.9	Examen de ejemplo.....	61
4.10	Bibliografía sugerida.....	62
4.11	<i>Lab handbook</i>	63

5. GUÍA DEL INSTRUCTOR DE ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 2

5.1	Datos generales.....	71
5.2	Distribución del curso.....	72
5.3	Evaluación.....	72
5.4	Detalle de sesiones.....	73
5.5	Distribución de sesiones.....	74
5.6	Detalle de <i>lab assignments</i>	79

5.7	Detalle del proyecto.....	80
5.8	Detalle del examen.....	81
5.9	Examen de ejemplo.....	82
5.10	Bibliografía sugerida.....	82
5.11	<i>Lab handbook</i>	83
6. DOCUMENTACIÓN DE APOYO DEL CURSO DE BASES DE DATOS 2		
6.1	Datos generales.....	85
6.2	Transacciones y control de concurrencia.....	88
6.3	<i>Backup</i> y recuperación	92
6.4	Seguridad	97
6.5	Optimización y alto rendimiento.....	100
6.6	Sistemas distribuidos de bases de datos.....	103
6.7	Análisis multidimensional y <i>Data Warehouse</i>	107
7. DOCUMENTACIÓN DE APOYO DEL CURSO DE INTELIGENCIA ARTIFICIAL		
7.1	Datos generales.....	115
7.2	Historia de la inteligencia artificial.....	119
7.3	Algoritmos genéticos.....	121
7.4	La <i>Web</i> semántica	123
7.5	<i>Machine Learning</i>	125
7.6	Redes neuronales.....	130
7.7	Agentes.....	139
8. DOCUMENTACIÓN DE APOYO DEL CURSO DE SEMINARIO DE SISTEMAS 2		
8.1	Datos generales.....	151
8.2	<i>Data Warehouse</i>	152
8.3	Diseño.....	156
8.4	ETL y DSS	158

8.5 <i>Data Mining</i>	162
CONCLUSIONES.....	169
RECOMENDACIONES	171
BIBLIOGRAFÍA.....	173
ANEXOS.....	175

ÍNDICE DE ILUSTRACIONES

FIGURAS

1. Examen ejemplo de Bases de Datos 1	36
2. Examen ejemplo de Bases de Datos 2.....	62
3. Examen ejemplo de Ensambladores 2	82
4. Páginas 1-4 “Bases de Datos 2”	86
5. Páginas 5-8 “Bases de Datos 2”	87
6. Páginas 9-12 “Bases de Datos 2”	88
7. Páginas 13-16 “Bases de Datos 2”	89
8. Páginas 17-20 “Bases de Datos 2”	90
9. Páginas 21-24 “Bases de Datos 2”	91
10. Páginas 25-28 “Bases de Datos 2”	92
11. Páginas 29-32 “Bases de Datos 2”	93
12. Páginas 33-36 “Bases de Datos 2”	94
13. Páginas 37-40 “Bases de Datos 2”	95
14. Páginas 41-44 “Bases de Datos 2”	96
15. Páginas 45-48 “Bases de Datos 2”	97
16. Páginas 49-52 “Bases de Datos 2”	98
17. Páginas 53-56 “Bases de Datos 2”	99
18. Páginas 57-60 “Bases de Datos 2”	100
19. Páginas 61-64 “Bases de Datos 2”	101
20. Páginas 65-68 “Bases de Datos 2”	102
21. Páginas 69-72 “Bases de Datos 2”	103
22. Páginas 73-76 “Bases de Datos 2”	104
23. Páginas 77-80 “Bases de Datos 2”	105
24. Páginas 81-84 “Bases de Datos 2”	106
25. Páginas 85-88 “Bases de Datos 2”	107
26. Páginas 89-92 “Bases de Datos 2”	108
27. Páginas 93-96 “Bases de Datos 2”	109

28. Páginas 97-100 “Bases de Datos 2”	110
29. Páginas 101-104 “Bases de Datos 2”	111
30. Páginas 105-108 “Bases de Datos 2”	112
31. Páginas 109-112 “Bases de Datos 2”	113
32. Páginas 113-114 “Bases de Datos 2”	114
33. Páginas 1-4 “Inteligencia Artificial”	117
34. Páginas 5-8 “Inteligencia Artificial”	118
35. Páginas 9-12 “Inteligencia Artificial”	119
36. Páginas 13-16 “Inteligencia Artificial”	120
37. Páginas 17-20 “Inteligencia Artificial”	121
38. Páginas 21-24 “Inteligencia Artificial”	122
39. Páginas 25-28 “Inteligencia Artificial”	123
40. Páginas 29-32 “Inteligencia Artificial”	124
41. Páginas 33-36 “Inteligencia Artificial”	125
42. Páginas 37-40 “Inteligencia Artificial”	126
43. Páginas 41-44 “Inteligencia Artificial”	127
44. Páginas 45-48 “Inteligencia Artificial”	128
45. Páginas 49-52 “Inteligencia Artificial”	129
46. Páginas 53-56 “Inteligencia Artificial”	130
47. Páginas 57-60 “Inteligencia Artificial”	131
48. Páginas 61-64 “Inteligencia Artificial”	132
49. Páginas 65-68 “Inteligencia Artificial”	133
50. Páginas 69-72 “Inteligencia Artificial”	134
51. Páginas 73-76 “Inteligencia Artificial”	135
52. Páginas 77-80 “Inteligencia Artificial”	136
53. Páginas 81-84 “Inteligencia Artificial”	137
54. Páginas 85-88 “Inteligencia Artificial”	138
55. Páginas 89-92 “Inteligencia Artificial”	139
56. Páginas 93-96 “Inteligencia Artificial”	140
57. Páginas 97-100 “Inteligencia Artificial”	141
58. Páginas 101-104 “Inteligencia Artificial”	142
59. Páginas 105-108 “Inteligencia Artificial”	143
60. Páginas 109-112 “Inteligencia Artificial”	144
61. Páginas 113-116 “Inteligencia Artificial”	145
62. Páginas 117-120 “Inteligencia Artificial”	146
63. Páginas 121-124 “Inteligencia Artificial”	147
64. Páginas 125-128 “Inteligencia Artificial”	148
65. Páginas 129-132 “Inteligencia Artificial”	149

66. Páginas 1-4 “ <i>Data Warehouse</i> ”	152
67. Páginas 5-8 “ <i>Data Warehouse</i> ”	153
68. Páginas 9-12 “ <i>Data Warehouse</i> ”	154
69. Páginas 13-16 “ <i>Data Warehouse</i> ”	155
70. Páginas 17-20 “ <i>Data Warehouse</i> ”	156
71. Páginas 21-24 “ <i>Data Warehouse</i> ”	157
72. Páginas 25-28 “ <i>Data Warehouse</i> ”	158
73. Páginas 29-32 “ <i>Data Warehouse</i> ”	159
74. Páginas 33-36 “ <i>Data Warehouse</i> ”	160
75. Páginas 37-40 “ <i>Data Warehouse</i> ”	161
76. Páginas 41-44 “ <i>Data Warehouse</i> ”	162
77. Páginas 45-48 “ <i>Data Warehouse</i> ”	163
78. Páginas 49-52 “ <i>Data Warehouse</i> ”	164
79. Páginas 53-56 “ <i>Data Warehouse</i> ”	165
80. Páginas 57-60 “ <i>Data Warehouse</i> ”	166
81. Páginas 61-62 “ <i>Data Warehouse</i> ”	167

TABLAS

I. Distribución de horas y actividades.	12
II. Forma de evaluación	13
III. Detalle de sesiones	13
IV. Distribución curso Sistemas de Bases de Datos 1	18
V. Evaluación curso Sistemas de Bases de Datos 1	18
VI. Sesiones curso Sistemas de Bases de Datos 1	19
VII. Distribución de sesiones de Bases de Datos 1	19
VIII: Prácticas curso Sistemas de Bases de Datos.....	31
IX: Proyectos curso Sistemas de Bases de Datos 1.....	34
X: Exámenes curso Sistemas de Bases de Datos 1.....	35
XI: Distribución curso Sistemas de Bases de Datos 2.....	46
XII: Evaluación curso Sistemas de Bases de Datos 2	46
XIII: Sesiones curso Sistemas de Bases de Datos 2	47

XIV: Distribución de sesiones de Bases de Datos 2.....	48
XV. Prácticas curso Sistemas de Bases de Datos 2	56
XVI. Proyecto curso Sistemas de Bases de Datos 2	60
XVII. Distribución curso Arquitectura de Computadores y Ensambladores 2	72
XVIII. Evaluación Arquitectura de Computadores y Ensambladores 2	72
XIX. Sesiones curso Arquitectura de Computadores y Ensambladores 2	73
XX. Distribución de sesiones de Ensambladores 2	74
XXI. Prácticas curso Arquitectura de Computadores y Ensambladores 2	79
XXII. Proyectos curso Arquitectura de Computadores y Ensambladores 2	81

GLOSARIO

<i>Instructor Guideline</i>	Es una guía estructurada que tiene como fin estandarizar un módulo y establecer los elementos que se utilizan para impartirlo.
EPS	Ejercicio Profesional Supervisado
Módulo	Unidad de medida por medio de la cual se agrupan todas las actividades relacionadas para poder impartir un curso.
Sesión	Se define como un conjunto de temas que se agrupan con el fin de ser impartidos por un instructor, en una unidad determinada de tiempo.

RESUMEN

Los cambios en las tecnologías de comunicación e información están en constante evolución y la única manera de mantener su paso es actualizando los contenidos que se imparten en la Escuela de Ciencias y Sistemas. Así mismo, es necesario velar por la calidad de la enseñanza impartida por esta institución.

Todo esto nos lleva a la necesidad de crear un sistema para definir de manera clara y detallada la estructura que debe llevar cada curso de la carrera de Ciencias y Sistemas. Para este Ejercicio Profesional Supervisado (EPS), se busca crear dicha estructura por medio de una documentación detallada.

El primer paso para crear una plataforma sobre la cual se pueda crear una guía para todos los cursos fue la creación de los *instructor guidelines*, documentos que son utilizados por el Centro Tecnológico de la India para manejar e impartir los cursos que ofrecen. El personal del Centro Tecnológico aportó sus valiosos conocimientos y participó en la supervisión de la creación de dichos documentos.

Se impartió un curso intensivo en la elaboración de *instructor guidelines* durante cuatro meses, en el período de enero a abril de 2008. Con base en este conocimiento se elaboraron *instructor guidelines* para los cursos de Sistemas de Bases de Datos 1, Sistemas de Bases de Datos 2 y Arquitectura de Computadores y Ensambladores 2.

Posteriormente a la creación de estos documentos se crearon guías para las clases de Sistemas de Bases de Datos 2, Inteligencia Artificial y Seminario de Sistemas 2, por medio de un libro de texto que contiene las unidades más importantes de cada curso y otros materiales de apoyo.

Los documentos descritos en los párrafos anteriores fueron elaborados en el presente Ejercicio Profesional Supervisado, sobre los cuales se aplicó la metodología enseñada por el Centro Tecnológico de la India.

OBJETIVOS

GENERAL

Crear una guía en la cual se pueda definir la estructura completa de tres cursos impartidos por la Escuela de Ciencias y Sistemas, aplicando la metodología de los documentos denominados *instructor guidelines*, con el fin de estandarizar su contenido. También proporcionar material de apoyo para tres cursos en formato de un libro de texto, conteniendo las unidades más importantes de cada uno.

ESPECÍFICOS

1. Elaborar un *instructor guideline*, que por medio de módulos pueda estructurar el contenido y las sesiones de un curso, también definir la forma en que se evalúa e imparten los siguientes:
 - Sistemas de Bases de Datos 1
 - Sistemas de Bases de Datos 2
 - Arquitectura de Computadores y Ensambladores 2
2. Elaborar un *lab handbook*, como documentación complementaria para cada *instructor guideline* respectivo, el cual contiene detalles sobre ejercicios, tareas y exámenes.
3. Proporcionar material gráfico para detallar cada sesión en los *instructor guidelines* y su contenido.

4. Desarrollar documentación de apoyo proporcionando al lector un texto de referencia que contiene las unidades más importantes de cada curso, este material se desarrolló para los siguientes cursos:
 - Sistemas de Bases de Datos 2
 - Seminario de Sistemas 2
 - Inteligencia Artificial
5. Documentar una serie de artículos relacionados en cada unidad del curso, para el material de apoyo de clase.

INTRODUCCIÓN

Mantener el nivel educativo de los cursos impartidos en la Escuela de Ciencias y Sistemas es crucial para garantizar un nivel de enseñanza de alto nivel como el que ha caracterizado a la Facultad de Ingeniería. De aquí surge la necesidad de estandarizar el contenido de los cursos para asegurar su nivel de calidad.

Es muy importante también mantener actualizados los contenidos de los cursos periódicamente, especialmente en una carrera que cambia constantemente, como sucede en la carrera de ciencias y sistemas.

Observando los problemas que la escuela ha tenido para coordinar diferentes secciones de un mismo curso y el contenido impartido tanto en la clase como en los laboratorios, se ve en la necesidad de crear un sistema por medio del cual se pueda estandarizar el contenido de los mismos.

Como solución a este problema, la Escuela de Ciencias y Sistemas trabajando en conjunto con la empresa *TATA Consultancy Services* y por medio del proyecto *India-Guatemala IT Education Centre of Excellence*, creó un curso impartido a los estudiantes para poder realizar *instructor guidelines*. En este curso se imparte la metodología que el *IT Education Centre of Excellence* ha utilizado para impartir sus propios cursos, mostrando la manera en que organizan y estructuran cada módulo.

Se le indicó a cada persona trabajando en este proyecto que iba a tomar tres cursos del área de Ciencias y Sistemas y que iba a aplicar esta nueva metodología. Cada *instructor guideline* fue creado de acuerdo al curso impartido por el señor Mrutunjaya Panda, que junto a catedráticos de cada curso ayudó al personal del proyecto a crear los documentos.

Este documento comienza explicando en qué consiste esta metodología y en los capítulos posteriores se aplica la metodología a los cursos de Sistemas de Bases de Datos 1, Sistemas de Bases de Datos 2 y Arquitectura de Computadores y Ensambladores 2. También se muestra el contenido desarrollado para las guías de clase de los cursos Sistemas de Bases de Datos 2, Inteligencia Artificial y Seminario de Sistemas 2.

1. MARCO TEÓRICO

1.1. Principios de Pedagogía

De la definición de los objetivos principales de este proyecto surge la necesidad de definir algunos conceptos básicos de Pedagogía.

La Pedagogía, es la rama de la ciencia que se puede resumir en la frase “el arte de enseñar”. Esta ciencia busca tanto definir como aplicar las técnicas y prácticas que se realizan con el fin de facilitar el proceso de aprendizaje. Se busca optimizar el proceso de aprendizaje con el objetivo de beneficiar a ambos: el educador y el objetivo a ser educado.

Ya que mediante esta ciencia estamos llegando al pensamiento humano, es imposible definir esta disciplina sin mencionar a la Psicología. La Psicología, como ciencia individual busca estudiar los procesos mentales; mediante el estudio del comportamiento humano se pueden crear programas con el fin de optimizarlo.

De acuerdo a Chickerin y Gamson, los principios de la Pedagogía son siete.

- a. Propiciar el contacto entre estudiantes y profesores. Aquí vemos el elemento fundamental de la Psicología en la definición de la Pedagogía. Es muy importante que el profesor logre crear un vínculo

con sus estudiantes, la dinámica que se practica aquí varía dependiendo del número de estudiantes que se tiene. Si el estudiante se siente cómodo se logra crear un ambiente positivo en el cual es más fácil impartir el conocimiento.

- b. Fomentar la cooperación entre los estudiantes. Nunca se debe de olvidar que la interacción entre los estudiantes es muy importante en el proceso educativo. El profesor siempre debe incentivar el trabajo en equipo
- c. Propiciar el aprendizaje activo. El aprendizaje activo se refiere al grado de participación del estudiante. Este objetivo se puede derivar del primer y segundo objetivo. Si el estudiante siente un vínculo con el profesor y se ha creado un ambiente cooperativo entre los alumnos la participación del alumno viene de manera natural. La participación tiene como objetivo tanto ayudar al alumno como al profesor. De esta manera se logra una retroalimentación que se obtiene del alumno, tanto las opiniones como las vivencias pueden complementar el programa de estudios base en una clase.
- d. Proporcionar retroalimentación a tiempo. Es importante saber manejar la retroalimentación que se obtiene del estudiante y saber en qué casos es necesaria una respuesta inmediata. De esta manera se incentiva este tipo de comportamiento mejorando la sinergia del grupo completo.
- e. Enfatizar el uso apropiado del tiempo. La planeación del tiempo es crucial en el proceso de aprendizaje, los beneficios que obtenemos de una buena planeación van desde impartir el contenido deseado hasta aliviar la carga de trabajo en los alumnos. Tanto el alumno como el profesor se benefician de un programa bien definido en términos de tiempo. También es muy importante el manejo del tiempo en cada una de las sesiones de clase.

- f. Propiciar altas expectativas al estudiante. Al incentivar al estudiante a participar e intentar obtener un poco más de lo que se le pide normalmente a un alumno estamos creando metas más altas para él. Es importante hacerle saber al alumno que estas metas pueden alcanzarse y que los beneficios que se obtienen al lograr estas metas son muchos.
- g. Respetar los diferentes estilos de aprendizaje. Este principio afirma que un programa de estudio, o un método de estudio no puede mantenerse estático. Es importante considerar la necesidad de cambiar e incluso aceptar otros estilos de aprendizaje. Mediante la observación activa podemos incorporar al programa básico de estudios ideas que podrían beneficiarlo.

1.2. Proceso enseñanza – aprendizaje

- a. Educación. Es un proceso social en el cual se busca que el alumno desarrolle sus capacidades mentales, técnicas o físicas para aumentar su habilidad en determinado campo o tema.
- b. Enseñanza. El sistema de enseñanza se basa en el modelo Estudiante-Profesor. El elemento llamado profesor tiene como objetivo impartir el conocimiento que posee sobre cierta área, la manera en que imparte dicho conocimiento demanda de él creatividad y habilidad. El segundo elemento en este modelo es llamado estudiante y es el sujeto al cual se le busca impartir conocimiento. Para que este sistema funcione se necesita del compromiso de ambos componentes.
- c. Aprendizaje. A su nivel más bajo se define como el proceso de adquirir conocimientos. La naturaleza de este conocimiento abarca tanto

aspectos intelectuales como físicos y emocionales. Si el alumno logra aumentar dicho nivel de conocimiento podemos afirmar que se está logrando llegar al aprendizaje.

El tipo de aprendizaje que se busca en este proyecto es el aprendizaje intelectual, el objetivo que se busca es impartir conocimiento. Bajo este principio definimos lo que es el proceso de enseñanza-aprendizaje, en este proceso tenemos tres elementos principales. El primer elemento es el estudiante y el segundo es el alumno. El tercer elemento es lo que llamamos cuerpo de conocimiento, que se puede definir como el resultado del proceso de aprendizaje. El cuerpo de conocimiento en sí es el conocimiento adquirido después de impartirse una clase, conocimiento que aumenta su capacidad mental o física.

Podemos ver que el proceso de enseñanza-aprendizaje es bastante complejo e involucra una gran cantidad de variables. Es por esa razón que resulta crucial tener una guía sobre la cual se pueda basar un plan para impartir conocimiento.

De los tres elementos principales descritos en el proceso enseñanza-aprendizaje existe tres interacciones directas que deben ser detalladas:

- a. Interacción profesor – conocimiento. El elemento profesor debe tener claro cuáles son los recursos que tiene a su disposición para impartir conocimiento.
- b. Interacción alumno – conocimiento. El alumno es un elemento crucial en este sistema y sin su colaboración no se puede completar el proceso de aprendizaje.

- c. Interacción profesor – alumno. Es muy importante que el profesor pueda crear un vínculo con el estudiante para involucrarlo en el proceso y despertar en él un deseo de aprender.

En el proceso de enseñanza-aprendizaje tenemos dos elementos que están íntimamente relacionados, en este proceso si definimos la acción por medio de la cual se imparte el conocimiento estamos describiendo a la acción didáctica. Para ejecutar la acción didáctica debemos conocer sus tres componentes principales:

- a. Planteamiento. En esta etapa se definen los objetivos que se buscan lograr, a su vez definimos el conjunto de actividades que se deben realizar para orientarnos hacia dichos objetivos.
- b. Ejecución. Basándonos en el plan descrito anteriormente y en los recursos que el profesor tiene a su disposición, se comienza el proceso de impartir conocimiento.
- c. Evaluación. En esta etapa observamos si se han cumplido los objetivos propuestos. En base a esta observación podemos realizar cambios a nuestro planteamiento inicial.

1.3. Método de clases magistrales

Existen diferentes maneras en que un profesor puede impartir una clase, podemos definir dos métodos principales. El primer método es lo que se llama una clase activa, en este tipo de método se incentiva al estudiante a participar en la clase. Esto se logra haciendo directamente preguntas a los estudiantes o incentivando la participación directa.

El segundo método se llama clase magistral, aquí se tiene un sistema diferente a la clase activa ya que lo que se busca es que el profesor imparta una clase sin ser interrumpido por los estudiantes. El profesor comienza a dar la clase y los estudiantes pueden tomar nota de los elementos más importantes, pero su participación es pasiva ya que no se involucra directamente en la conversación.

Los beneficios principales de una clase magistral son la facilidad de administrar el tiempo y la capacidad de impartir mayor cantidad de material en una unidad de tiempo determinada. Es por esta razón que este método se utiliza con mayor frecuencia. Con este método es más fácil para el profesor preparar las sesiones a impartir.

Los elementos principales que debemos manejar durante una clase magistral son:

- La voz. Debemos tomar en cuenta el lugar en el cual se va a impartir la clase, si el lugar es muy grande es posible que tengamos necesidad de un micrófono. Por otro lado si el lugar que utilizamos es pequeño o la audiencia es de tamaño reducido tener un micrófono solamente molestaría al público.
- El cuerpo. El uso adecuado del cuerpo al realizar una exposición es muy importante. Es más fácil para el público seguir a una persona que está en constante movimiento ya que evita que la exposición se vuelva aburrida. Por otro lado no debemos exagerar nuestros movimientos ya que el público pierde el interés en el contenido.

- Las Imágenes visuales. El uso de diferentes ayudas visuales nos permite captar la atención del público con mayor facilidad y en muchas ocasiones facilita el entendimiento de algunos temas.
- Invitados. Traer como invitado a un experto en cierta materia puede ser beneficioso para una exposición.
- El plan de exposición. Es importante mostrarle al público al inicio cuál es el plan completo a seguir durante la exposición, ya que de esta manera puede saber de antemano que temas son de mayor interés para él.
- El texto de la clase. En algunas ocasiones puede ser de gran ayuda proveer al público con material escrito para que pueda seguir con mayor facilidad la exposición.
- Resumen final. Nos ayuda a repetir elementos importantes de la exposición.

1.4. Aprendizaje práctico por medio de laboratorios

Existen ciertos temas que no pueden impartirse únicamente como una clase teórica, temas en los cuáles la parte práctica es tan importante o en algunos casos más importante. En este tipo de casos es una buena idea pensar en la utilización de un laboratorio, es una manera sencilla de involucrar directamente al estudiante y ayudarlo a comprender conceptos teóricos.

El lado negativo de este método es su alto costo en ciertos casos, si no se tienen los recursos adecuados y se tiene un grupo numeroso de estudiantes el laboratorio no es efectivo. Por esta razón se debe realizar un estudio detallado

de quien será el grupo de estudiantes al cual el laboratorio está destinado y también se debe saber con qué tipo de equipo se puede contar.

2. GUÍA DEL INSTRUCTOR

2.1. Definición de una guía del instructor

Un *instructor guideline* es un documento que tiene como objetivo estructurar, estandarizar y detallar las diferentes actividades que se realizan al impartir una clase. Buscan también definir de manera clara el contenido que se busca impartir y el tiempo en el cual se debe hacer.

El proceso de enseñanza pide de nosotros una planeación del tiempo bien definida ya que es el elemento principal que se busca optimizar. Debemos saber qué tipo de actividades se deben realizar y con qué frecuencia se debe realizar cada una.

La planificación se detalla en un documento escrito, y dicha planeación se realiza en base a una serie de módulos bien definidos. La estructura de este documento y la manera en la cual se trabaja ha sido probada y utilizada por varios años por la empresa TATA.

Cada *instructor guideline* va dirigido directamente a la persona que está encargada de impartir la clase, ya sea un auxiliar o un catedrático. Cada una de las secciones de este documento provee una herramienta importante cuya finalidad es facilitar la labor de impartir clases.

La planificación del proceso de enseñanza debe poseer las siguientes características:

- Flexibilidad. Debemos tener en cuenta que los planes que se hacen pueden ser adaptados a los cambios que sean necesarios.
- Realidad. Se debe saber exactamente con que recursos se cuenta en el lugar que se imparte una clase, a partir de este conocimiento se puede crear un plan efectivo para lograr los objetivos planteados para cada módulo.
- Precisión. Cada guía debe definir cada elemento de manera clara y precisa. Si se da lugar a la ambigüedad se puede llegar a tener problemas de entendimiento y por consecuencia la guía se aleja de los objetivos planteados.

2.2. Objetivos de una guía del instructor

Los objetivos principales del *instructor guideline* son estructurar, ordenar, estandarizar y especificar el contenido que se utiliza para impartir una clase o un laboratorio.

Esta guía permite llevar el control con más facilidad de un curso cuando se tiene diferentes secciones u horarios. Evita la necesidad de verificar constantemente entre los diferentes catedráticos sobre el contenido y el ritmo impartido en el curso. Al tener un punto de referencia claro sobre el cual respaldarse es más fácil llevar el control del curso y evitar problemas de tiempo y de contenido debido a la planeación.

2.3. Elementos de la guía del instructor

Un *instructor guideline* está compuesto por varios elementos, estos son los componentes principales:

- a. Información general del curso.
- b. Distribución de horas y actividades.
- c. Forma de evaluación.
- d. Detalle de sesiones.
- e. Distribución de sesiones.
- f. Detalle de tareas asignadas
- g. Detalle de exámenes.
- h. Ejemplo de exámenes.
- i. Bibliografía.

Adicionalmente se cuenta con un documento adicional llamado *lab handbook*, en este documento se muestran detalles sobre ejercicios, exámenes y tareas. Dicho contenido no se incluye en el documento principal para no sobrecargarlo. Con este documento se logra entrar en detalle sobre varios aspectos y se da una serie de ejemplos para comprender mejor los objetivos de cada módulo

2.4. Proceso de construcción de la guía del instructor

En la información general del curso se describe la idea general e introductoria del curso, esta debe contener:

- Nombre del curso. Según el pensum, incluye el código del curso.
- Prerrequisitos. Los cursos previos que son requisitos obligatorios.
- Objetivo. La meta a alcanzar para el curso.
- Módulos. Componentes del curso con diferente finalidad.

En la sección de distribución de horas y actividades se especifica cuantos módulos se trabajan por curso y se detalla para cada uno las diferentes actividades que se realizan y la cantidad en horas de tiempo que se tarda el instructor en impartirla.

Las actividades que se detallan son: tareas, exámenes y proyectos. En la siguiente tabla muestra un ejemplo de la distribución de horas y actividades:

Tabla I: Distribución de horas y actividades.

MODULO	TEORÍA	PRÁCTICA	TOTAL	EXAMEN	PRACTICA DE LAB.	PROYECTOS
Nombre del módulo	X horas	X horas	X horas	1	10	1
			X horas			

En la forma de evaluación se le asigna un valor numérico a cada actividad que se realizará para llegar a un 100% de la nota total. Aquí se tiene una tabla de ejemplo:

Tabla II. Forma de evaluación

	NUMERO	PORCENTAJE	TOTAL
TUTORIAL	3	2.5%	7.50 %
TAREAS	10	4.0%	40.00%
PROYECTO	1	30.0%	30.00%
EXAMEN	1	20.0%	20.00%
		TOTAL	100.00%

En el detalle de sesiones se define el número de sesiones en las cuales se divide cada módulo del curso, se muestra también la duración, definiendo si existirá alguna tarea en cada sesión. También se detalla cuando exista un proyecto pendiente.

Tabla III. Detalle de sesiones

NO.	TEORIA	NO.	PRACTICA	TAREA
1	MÓDULO – SESIÓN 1			
2	MÓDULO – SESIÓN 2	1	MÓDULO – LAB ASSIGNMENT 1	TAREA 1
3	MÓDULO – SESIÓN 3	2	MÓDULO – LAB ASSIGNMENT 2	TAREA 2

En la distribución de sesiones se debe considerar que previo a la realización de una sesión, en un módulo se deben tomar en cuenta una serie de factores diferentes. El primer factor es la naturaleza del contenido que se debe incluir, para considerar esto se debe saber que cursos ha llevado anteriormente el estudiante para establecer su base de conocimiento. Otro factor importante es el tiempo de cada sesión, cada sesión está programada para impartirse en dos horas, de manera que se debe estimar el grado de dificultad del contenido o si

habrá necesidad de mostrar varios ejemplos para explicar bien el tema. Finalmente se especifica en cada sesión si se estará haciendo un examen y cuál debe ser su duración.

La distribución de sesiones se realiza dividiendo el módulo completo en una cantidad fija de sesiones, en el documento debemos identificar el nombre del curso, el módulo al que pertenece, el número de sesión, el objetivo, la distribución de temas y el objetivo del módulo.

Para el detalle de tareas asignadas se dividen de manera similar a la distribución de sesiones las diferentes prácticas que se deben realizar. Se detalla aquí: el módulo al que pertenece, el objetivo que se busca, la forma de evaluación, el material a entregar y finalmente en qué momento se debe hacer.

En el detalle de exámenes se define la manera en que se debe hacer un examen para el módulo. Se detalla aquí: el módulo al que pertenece, cuando se debe realizar, el contenido a evaluar, la duración y el modo de evaluación.

En ejemplo de exámenes se logra tener una mejor idea de los objetivos que se buscan al realizar una prueba se muestra en esta sección un ejemplo completo de un examen. Por medio de un ejemplo es más fácil apreciar de qué manera se debe crear la evaluación.

La bibliografía es el conjunto de material didáctico que se utiliza como base para creación del contenido incluido en cada sesión. También se detalla para tener un texto de referencia en caso de necesitar ampliar la información dada.

Un *lab handbook* es un documento anexo a la guía principal en el cual se detallan aspectos importantes de la guía, se realizan aquí para no sobrecargar

la guía principal y para tener mejor estructurado el contenido. Está dividido en las siguientes secciones:

1. Ejercicios de laboratorio. Son actividades que se realizan durante una sesión de clases, es decir se detiene la clase un momento para darle la oportunidad a los estudiantes de realizar un ejercicio rápido.
2. *Lab assignments*. Son las actividades que los estudiantes realizan fuera del horario de las sesiones de clase.
3. Tutoriales. Son documentos realizados por parte del estudiante en los cuales se detallan los pasos que se deben seguir para realizar una actividad.
4. Proyectos. Tienen como objetivo evaluar de una manera integral el contenido teórico y práctico impartido en cada sesión de clase. Normalmente tiene una duración de dos semanas a un mes. Se da este tiempo ya que usualmente el problema que se debe resolver es de un alto grado de dificultad.

2.5. Forma de uso de la guía del instructor

Cada *instructor guideline* tiene como objetivo ayudar a la persona que imparte las sesiones de cada módulo. Esta ayuda abarca varios aspectos entre ellos: administración del tiempo al detallar cada sesión, el contenido a abarcar en cada sesión, el número de ejercicios y evaluaciones que se deben realizar, la manera en que se deben calificar y los objetivos que se buscan para cada módulo.

Adicionalmente se provee del *lab handbook* que nos permite explicar en detalle cada ejercicio y evaluación que se realiza, en este documento no solamente muestra un ejemplo de cada actividad sino que también se detalla cuáles son sus objetivos.

3. GUÍA DEL INSTRUCTOR DE SISTEMAS DE BASES DE DATOS 1

A continuación se presenta el desarrollo de la Guía del curso de Sistemas de Bases de Datos 1, de acuerdo al formato sugerido en el curso de “Estructuración de laboratorios”, impartido por el señor. Mrutunjaya Panda.

3.1. Datos generales

NOMBRE DE CURSO:	Sistemas de Bases de Datos 1 (774)
PRE- REQUISITOS:	Manejo e Implementación de archivos (773)
POST – REQUISITOS:	Sistemas de Bases de Datos 2 (775) Análisis y Diseño de Sistemas 1 (283)
OBJETIVO:	Al finalizar el curso, el estudiante podrá diseñar de una manera eficiente un modelo complejo de bases de datos, también será capaz de utilizar el lenguaje SQL para realizar cualquier tipo de consulta.

3.2. Distribución del curso

Tabla IV: Distribución curso Sistemas de Bases de Datos 1

MODULO	TEORÍA	PRÁCTICA	TOTAL	EXAMEN	PRACTICA DE LAB.	PROYECTOS
Modelado de Bases de Datos y SQL	30 horas	20 horas	50 horas	2	10	2
			50 horas			

3.3. Evaluación

Ponderación de tareas, prácticas y proyectos.

Tabla V: Evaluación curso Sistemas De Bases de Datos 1

ACTIVIDAD	NUMERO	PORCENTAJE	TOTAL
PRESENTACION	1	8.0 %	8.00 %
PRACTICA DE LABORATORIO	10	1.0%	10.00%
EXAMEN	2	8.0 %	16.00 %
ASISTENCIA	10	0.2 %	2.00 %
PROYECTO	2	32.0 %	64.00 %
		TOTAL	100.00 %

3.4. Detalle de sesiones

Tabla VI: Sesiones curso Sistemas de Bases de Datos1

No	TEORIA	No	PRACTICA	PRACTICA DE LAB / PROYECTO
1	Modelado de BD y SQL 1	1	Modelado de BD y SQL 1	Práctica de Laboratorio 1
2	Modelado de BD y SQL 2	2	Modelado de BD y SQL 2	Práctica de Laboratorio 2
3	Modelado de BD y SQL 3	3	Modelado de BD y SQL 3	Práctica de Laboratorio 3
4	Modelado de BD y SQL 4	4	Modelado de BD y SQL 4	Práctica de Laboratorio 4
5	Modelado de BD y SQL 5			
6	Modelado de BD y SQL 6			
7	Modelado de BD y SQL 7	5	Modelado de BD y SQL 5	Práctica de Laboratorio 5
8	Modelado de BD y SQL 8			
9	Modelado de BD y SQL 9	6	Modelado de BD y SQL 6	Práctica de Laboratorio 6
10	Modelado de BD y SQL 10	7	Modelado de BD y SQL 7	Primer Proyecto/ Práctica de Laboratorio 7
11	Modelado de BD y SQL 11	8	Modelado de BD y SQL 8	Práctica de Laboratorio 8
12	Modelado de BD y SQL 12	9	Modelado de BD y SQL 9	Práctica de Laboratorio 9
13	Modelado de BD y SQL 13	10	Modelado de BD y SQL 10	Segundo Proyecto / Práctica de Laboratorio 10
14	Modelado de BD y SQL 14			
15	Modelado de BD y SQL 15			

3.5. Distribución de sesiones

Tabla VII: Distribución de sesiones de Bases de Datos 1

SESIÓN NO. 1:	Desarrollo de un modelo
OBJETIVO:	Mostrar al estudiante la manera de realizar un modelo de datos a partir de un enunciado o problema.
CONTENIDO:	<ol style="list-style-type: none"> 1. Realización de un modelado de dificultad media de base de datos en clase. <ol style="list-style-type: none"> 1.1 Lectura del problema, y escritura de puntos clave. 1.2 Análisis preliminar del problema. 1.3 Identificación de entidades clave. 1.4 Creación de las entidades principales 1.5 Identificación de campos principales 1.6 Identificación de identificadores únicos 1.7 Identificación de relaciones entre entidades 1.8 Creación del resto de las entidades del problema 1.9 Revisión de relaciones y entidades del problema completo.
SESIÓN NO. 2:	Desarrollo de un modelo 2
OBJETIVO:	Mostrar al estudiante la manera de realizar un modelo de datos a partir de un enunciado o problema.
CONTENIDO:	<ol style="list-style-type: none"> 1. Realización de un modelado de dificultad media de base de datos en clase.

	<ul style="list-style-type: none"> 1.1 Lectura del problema, y escritura de puntos clave. 1.2 Análisis preliminar del problema. 1.3 Identificación de entidades clave. 1.4 Creación de las entidades principales 1.5 Identificación de campos principales 1.6 Identificación de identificadores únicos 1.7 Identificación de relaciones entre entidades 1.8 Creación del resto de las entidades del problema 1.9 Revisión de relaciones y entidades del problema completo.
SESIÓN NO. 3:	Desarrollo de un modelo 3
OBJETIVO:	Mostrar al estudiante la manera de realizar un modelo de datos a partir de un enunciado o problema.
CONTENIDO:	<ul style="list-style-type: none"> 1. Realización de un modelado de dificultad media de base de datos en clase. 1.1 Lectura del problema, y escritura de puntos clave. 1.2 Análisis preliminar del problema. 1.3 Identificación de entidades clave. 1.4 Creación de las entidades principales 1.5 Identificación de campos principales 1.6 Identificación de identificadores únicos 1.7 Identificación de relaciones entre entidades 1.8 Creación del resto de las entidades del problema

	1.9 Revisión de relaciones y entidades del problema completo.
SESIÓN NO. 4:	Modelado de entidades complejas
OBJETIVO:	Mostrar la manera de realizar entidades complejas y su respectivo mapeo conceptual.
CONTENIDO:	<ol style="list-style-type: none"> 1. Modelado de entidades complejas <ol style="list-style-type: none"> 1.1 Súper tipos y subtipos 1.2 Subtipos anidados 1.3 Arcos 1.4 Jerarquías y relaciones recursivas 1.5 Roles 1.6 Modelado en el tiempo
SESIÓN NO. 5:	Mapeo de entidades complejas y examen no. 1
OBJETIVO:	Consolidar y evaluar las habilidades adquiridas para realizar modelos complejos de bases de datos.
CONTENIDO:	<ol style="list-style-type: none"> 1. Mapeo de entidades complejas <ol style="list-style-type: none"> 1.1 Súper tipos y subtipos 1.2 Arcos 1.3 Relaciones recursivas 2. Examen de Laboratorio # 1: realización de un modelo de dificultad

	alta.
SESIÓN NO. 6:	Tablespaces e Índices
OBJETIVO:	Lograr que el estudiante maneje e utilice de manera eficiente tablespaces e índices.
CONTENIDO:	<ol style="list-style-type: none"> 1. <i>Tablespaces</i> <ol style="list-style-type: none"> 1.1 <i>Tablespace</i> 1.2 Tipos de <i>Tablespace</i> 1.3 Ejemplo de <i>Tablespace</i> 1.4 Modificar <i>Tablespace</i> 2. Índices <ol style="list-style-type: none"> 2.1 Índices 2.2 Recomendaciones 2.3 Creación
SESIÓN NO. 7:	Creación de la base de datos ejemplo
OBJETIVO:	Crear una base de datos utilizando <i>tablespaces</i> , e índices. Conocer cuáles son los archivos del sistema de la base de datos usando Oracle.
CONTENIDO:	<ol style="list-style-type: none"> 1. Creación de la base de datos ejemplo

	<ul style="list-style-type: none"> 1.1 Crear <i>tablespaces</i> 1.2 Crear tablas 1.3 Crear índices 2. Archivos de sistema <ul style="list-style-type: none"> 2.1 Data files 2.2 Redo log files 2.3 Control files
SESIÓN NO. 8:	Creación de la base de datos ejemplo 2
OBJETIVO:	<p>Crear una base de datos utilizando un <i>schema</i> e índices. Conocer cuáles son los archivos del sistema de la base de datos usando SQL.</p>
CONTENIDO:	<ul style="list-style-type: none"> 1. Creación de la base de datos ejemplo <ul style="list-style-type: none"> 1.1 Crear <i>schema</i> 1.2 Crear tablas 1.3 Crear llaves foráneas 1.4 Crear índices 2. Archivos de sistema <ul style="list-style-type: none"> 2.1 Estructura física 2.2 Data files 2.3 Log files

SESIÓN NO. 9:	Funciones y restricciones
OBJETIVO:	Uso de cláusulas para presentación y ordenamientos así como instrucciones de restricción avanzadas.
CONTENIDO:	<ol style="list-style-type: none"> 1. Funciones <ol style="list-style-type: none"> 1.1 Uso de funciones en Oracle (SUM, AVG, COUNT, MAX, MIN) 1.2 Funciones usando group by (SUM, AVG, COUNT, MAX, MIN) 1.3 <i>Group by</i> complejo 1.4 Cuando y porque se debe usar <i>group by</i>. 2. Restricciones avanzadas <ol style="list-style-type: none"> 2.1 Cláusula <i>having</i> 2.2 Ejemplo de cláusula <i>having</i> 2.3 Cláusula <i>IN, NOT IN</i> 2.4 Cláusula <i>BETWEEN</i>
SESIÓN NO. 10:	Campos calculados y sub consultas
OBJETIVO:	Mostrar al estudiante el uso de campos calculados y sub consultas.
CONTENIDO:	<ol style="list-style-type: none"> 1. Campos calculados

	<ul style="list-style-type: none"> 1.1 Campos calculados 1.2 Campos calculados usando funciones 1.3 Campos estáticos 2. Sub consultas <ul style="list-style-type: none"> 2.1 Uso de sub consultas dentro de la cláusula WHERE 2.2 Uso de sub consultas dentro de la cláusula FROM. 2.3 Sub consultas complejas realizadas en el WHERE y FROM. 2.4 Sub consultas Anidadas.
SESIÓN NO. 11:	SQL avanzado y vistas
OBJETIVO:	<p>Uso de <i>EXISTS</i> y <i>UNION</i> para SQL, creación de vistas y presentaciones por parte de los alumnos para el DBMS Oracle, como soporte para el segundo proyecto.</p>
CONTENIDO:	<ul style="list-style-type: none"> 1. SQL Avanzado <ul style="list-style-type: none"> 1.1 <i>EXISTS, NOT EXISTS</i> 1.2 <i>UNION</i> 1.3 <i>INTERSECT, MINUS</i> 2. Vistas <ul style="list-style-type: none"> 2.1 Que es una vista 2.2 Como crearlas 2.3 Manipular una vista

SESIÓN NO. 12:	DBMS SQL y segundo examen de laboratorio
OBJETIVO:	Presentaciones del DBMS SQL como soporte para el segundo proyecto así como la evaluación numero dos del laboratorio.
CONTENIDO:	<ol style="list-style-type: none"> 1. Presentaciones SQL <ol style="list-style-type: none"> 1.1 Instalación. 1.2 Características importantes 1.3 Creación de tablas 1.4 Ambiente para realizar consultas 2. Realización del segundo examen de laboratorio.
SESIÓN NO. 13:	Operaciones con cadenas y el DBMS Oracle
OBJETIVO:	Poder realizar operaciones durante consultas para cadenas. Realizar presentaciones por parte de los estudiantes para el DBMS Oracle.
CONTENIDO:	<ol style="list-style-type: none"> 1. Operaciones con cadenas para Oracle <ol style="list-style-type: none"> 1.1 <i>LIKE</i> con % y con _ 1.2 <i>UPPER, LOWER, INITCAP</i>

	<ul style="list-style-type: none"> 1.3 CONCAT, 1.4 SUBSTR 2. Operaciones con cadenas para SQL <ul style="list-style-type: none"> 2.1 LIKE con % y con _ 2.2 UPPER, LOWER 2.3 + 2.4 SUBSTRING, LEFT, RIGHT 3. Presentaciones Oracle <ul style="list-style-type: none"> 3.1 Instalación 3.2 Características importantes 3.3 Creación de tablas 3.4 Ambiente para realizar consultas
SESIÓN NO. 14:	Operaciones numéricas y Postgres
OBJETIVO:	Poder realizar operaciones durante consultas para fechas y números. Realizar presentaciones por parte de los estudiantes para el DBMS Oracle.
CONTENIDO:	<ul style="list-style-type: none"> 1. Operaciones para campos de fecha en Oracle <ul style="list-style-type: none"> 1.1 TO_CHAR aplicado a fechas 1.2 SYSDATE 1.3 MONTHS_BETWEEN, NEXT_DAY 2. Operaciones para campos de fecha en SQL

	<p>2.1 <i>GETDATE</i></p> <p>2.2 <i>YEAR, MONTH, DAY</i></p> <p>2.3 <i>DATEPART</i></p> <p>2.4 <i>DATEDIFF</i></p> <p>3. Operaciones para campos numéricos en Oracle</p> <p>3.1 Cálculos numéricos</p> <p>3.2 SQRT</p> <p>3.3 POWER.</p> <p>4. Operaciones para campos numéricos en SQL</p> <p>4.1 Cálculos numéricos</p> <p>4.2 LOG10</p> <p>4.3 COS</p> <p>5. Presentaciones para el DBMS Postgres</p> <p>5.1 Instalación</p> <p>5.2 Características importantes</p> <p>5.3 Creación de tablas</p> <p>5.4 Ambiente para realizar consultas</p>
SESIÓN NO. 15:	Presentaciones DB2 y MySQL
OBJETIVO:	Realizar presentaciones para los dos DBMS restantes por parte de los estudiantes.
CONTENIDO:	

	<p>1. Presentaciones para el DBMS DB2</p> <p>1.1 Instalación</p> <p>1.2 Características importantes</p> <p>1.3 Creación de tablas</p> <p>1.4 Ambiente para realizar consultas</p> <p>2. Presentaciones para el DBMS MySQL</p> <p>2.1 Instalación</p> <p>2.2 Características importantes</p> <p>2.3 Creación de tablas</p> <p>2.4 Ambiente para realizar consultas</p>
--	---

3.6. Detalle de *lab assignments*

Practica:

Criterio de Evaluación:

- Completar el trabajo: 20
- Corrección del trabajo: 80

Documentación a entregar: SI

- Solución del problema
- Imágenes de la solución

Módulo: Modelado de Bases de Datos y SQL.

Tabla VIII: Prácticas curso Sistemas de Bases de Datos 1

LAB ASSIGNMENT 1:	Evaluar la Practica 1 como Practica de Laboratorio
OBJETIVO:	Repasar de forma práctica conceptos de diseño de bases de datos
Iniciada en sesión	1
Entregada en sesión	2
Viva / Demostración	NO
LAB ASSIGNMENT 2:	Evaluar la Practica 2 como Practica de Laboratorio
OBJETIVO:	Repasar de forma práctica conceptos de diseño de bases de datos
Iniciada en sesión	2
Entregada en sesión	3
Viva / Demostración	NO
LAB ASSIGNMENT 3:	Evaluar la Practica 3 como Practica de Laboratorio
OBJETIVO:	Repasar de forma práctica conceptos de diseño de bases de
Iniciada en sesión	3
Entregada en sesión	4
Viva / Demostración	NO
LAB ASSIGNMENT 4:	Evaluar la Practica 4 como Practica de Laboratorio
OBJETIVO:	Repasar de forma práctica conceptos de diseño de bases de datos que se ven en el laboratorio de forma teórica

Iniciada en sesión	4
Entregada en sesión	5
Viva / Demostración	NO
LAB ASSIGNMENT 5:	Evaluar la Practica 5 como Practica de Laboratorio
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL
Iniciada en sesión	7
Entregada en sesión	8
Viva / Demostración	NO
LAB ASSIGNMENT 6:	Evaluar la Practica 6 como Practica de Laboratorio
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL
Iniciada en sesión	9
Entregada en sesión	10
Viva / Demostración	NO
LAB ASSIGNMENT 7:	Evaluar la Practica 7 como Practica de Laboratorio
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL
Iniciada en sesión	10
Entregada en sesión	11
Viva / Demostración	NO
LAB ASSIGNMENT 8:	Evaluar la Practica 8 como Practica de Laboratorio

OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL
Iniciada en sesión	11
Entregada en sesión	12
Viva / Demostración	NO
LAB ASSIGNMENT 9:	Evaluar la Practica 9 como Practica de Laboratorio
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL
Iniciada en sesión	12
Entregada en sesión	13
Viva / Demostración	NO
LAB ASSIGNMENT 10:	Evaluar la Practica 10 como Practica de Laboratorio
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL
Iniciada en sesión	13
Entregada en sesión	14
Viva / Demostración	NO

3.7. Detalle de proyectos

PROYECTO:

Criterio de evaluación:

- Funcionamiento: 70
- Conocimiento Técnico: 10
- Documentación: 20

Documentación a entregar: SI

- Enunciado

Módulo: Modelado de bases de datos y SQL

Tabla IX: Proyectos curso Sistemas de Bases de Datos 1

PROYECTO 1:	
OBJETIVO:	Realizar un modelo de datos sobre el cual se realizarán una serie de consultas en SQL.
Iniciada en sesión	10
Entregada en sesión	12
Viva / Demostración	SI
PROYECTO 2:	
OBJETIVO:	Realizar un modelo de datos sobre el cual se realizarán una serie de consultas en SQL.
Iniciada en sesión	13
Entregada en sesión	15
Viva / Demostración	SI

3.8. Detalle de exámenes

Número total de exámenes: 2

Examen 1. Desarrollo de modelos de datos y modelado de entidades complejas.

- Entregado en sesión: 5
- Duración: 2 horas.
- Criterios de Evaluación: Subjetivo 100 %.

Examen 2. Lenguaje de consultas SQL y teoría sobre bases de datos.

- Entregado en sesión: 12
- Duración: 2 horas.
- Criterios de Evaluación: Objetivo 40 %, Subjetivo 60 %.

Tabla X: Exámenes curso Sistemas de Bases de Datos 1

EXAMEN 1:	
Sesión	5
Temas	Desarrollo de modelos de datos y Modelado de entidades complejas.
Duración	2 horas
Criterios Evaluación	Subjetivo 100%
EXAMEN 2:	

Sesión	12
Temas	Lenguaje de consultas SQL y teoría sobre bases de datos.
Duración	15
Criterios Evaluación	Objetivo 40%, Subjetivo 60%

3.9. Examen de ejemplo

Figura 1. Examen ejemplo de Bases de Datos 1

SERIE I: Indique la respuesta que crea correcta que corresponda al concepto expuesto.

Clausula conocida como proyección que permite la selección de cierto número de atributos de la tabla.																								
Lenguaje de base de datos normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos o sobre la estructura de los mismos.																								
Función de agregado utilizada para calcular el promedio de los valores de un campo determinado.																								
Restricción utilizada para indicar el dominio o conjunto de datos y valores que puede tomar un atributo de una tabla de la base de datos.																								
Función de agregado que permite obtener cierto número de tuplas que se encuentran al inicio o final de un rango especificado por una clausula ORDER BY.																								
Restricción que especifica los valores de datos que son aceptables para actualizar una columna y que están basados en valores de datos provenientes de una o más columna de otras tablas.																								
Esta restricción previene la duplicación de valores de columnas que tienen valor único y que no son llave primaria, pero que pueden ser una llave alternativa o candidata.																								
Lenguaje que permite la creación y definición de nuevas bases de datos, tablas, campos, índices y otros objetos de base de datos.																								
Función de agregado que permite obtener el valor más bajo de un atributo especificado.																								
Sentencia utilizada para la modificación de la estructura de las tablas las cuales generalmente se encuentran en producción y contienen datos.																								
<table border="0"> <tr> <td>1. COUNT</td> <td>7. DML</td> <td>13. TOP</td> <td>19. UNIQUE</td> </tr> <tr> <td>2. DDL</td> <td>8. SUM</td> <td>14. CONSTRAINT</td> <td>20. FOREIGN KEY</td> </tr> <tr> <td>3. AVG</td> <td>9. SELECT</td> <td>15. PRIMARY KEY</td> <td>21. CHECK</td> </tr> <tr> <td>4. ETL</td> <td>10. ALTER TABLE</td> <td>16. SQL</td> <td>22. MAX</td> </tr> <tr> <td>5. WHERE</td> <td>11. DOMAIN</td> <td>17. MIN</td> <td>23. DEFAULT</td> </tr> <tr> <td>6. FROM</td> <td>12. CREATE</td> <td>18. DELETE</td> <td>24. DROP</td> </tr> </table>	1. COUNT	7. DML	13. TOP	19. UNIQUE	2. DDL	8. SUM	14. CONSTRAINT	20. FOREIGN KEY	3. AVG	9. SELECT	15. PRIMARY KEY	21. CHECK	4. ETL	10. ALTER TABLE	16. SQL	22. MAX	5. WHERE	11. DOMAIN	17. MIN	23. DEFAULT	6. FROM	12. CREATE	18. DELETE	24. DROP
1. COUNT	7. DML	13. TOP	19. UNIQUE																					
2. DDL	8. SUM	14. CONSTRAINT	20. FOREIGN KEY																					
3. AVG	9. SELECT	15. PRIMARY KEY	21. CHECK																					
4. ETL	10. ALTER TABLE	16. SQL	22. MAX																					
5. WHERE	11. DOMAIN	17. MIN	23. DEFAULT																					
6. FROM	12. CREATE	18. DELETE	24. DROP																					

3.10. Bibliografía sugerida

1. Date, C.J. **Introducción a los sistemas de bases de datos.**
2. Elmasri y Navathe. **Sistemas de Bases de datos.**
3. Ron Soukup, **Microsoft SQL Server a Fondo.**
4. Andy Opper. ***Databases Demystified.***
5. Kevin Loney. ***Oracle Database 10g: the complete reference.***

3.11. Lab handbook

Lab assignment 1. Hacer el diagrama de bases de datos para 2 enunciados diferentes.

Especificaciones:

- Ya que estos son los modelos de datos iniciales su grado de dificultad debe ser a lo sumo intermedio, (aproximadamente 6 entidades).
- Se debe poner atención especial a la manera en que se crean las llaves primarias.
- No se deben aceptar relaciones muchos a muchos, estas tienen que ser disueltas.
- No deben haber relaciones redundantes.
- Cada entidad debe tener una razón de ser.
- Se debe tener un cuidado especial en la opcionalidad de las relaciones, en ambos lados.
- Finalmente se debe revisar que tipo de propiedades hay en cada entidad.

Lab assignment 2. Hacer el diagrama de bases de datos para un enunciado específico.

Especificaciones:

- Para este modelo de datos el grado de dificultad debe incrementarse considerablemente al ejercicio anterior.
- El modelo debe de ser de aproximadamente 10 entidades.
- No se deben aceptar relaciones muchos a muchos.
- No deben haber relaciones redundantes ni entidades sin propósito.
- La opcionalidad de las relaciones es importante.
- La definición del problema debe ser bastante específico, dejando al mínimo el número de entidades sin especificar.
- Y en general se busca llegar a un nivel alto de funcionalidad con el modelo de datos.

Lab assignment 3. Hacer el diagrama de bases de datos para un enunciado específico.

Especificaciones:

- Para este modelo de datos el grado de dificultad debe incrementarse ligeramente al ejercicio anterior.
- El modelo debe de ser de aproximadamente 15 entidades.
- No se deben aceptar relaciones muchos a muchos.
- No deben haber relaciones redundantes ni entidades sin propósito.
- La opcionalidad de las relaciones es importante.
- La definición del problema debe especificarse en la mayor parte del problema, por lo menos en la estructura principal.

- Y en general se busca llegar a un nivel alto de funcionalidad con el modelo de datos.

Lab assignment 4. Hacer el diagrama de bases de datos para un enunciado específico.

Especificaciones:

- No se deben aceptar relaciones muchos a muchos.
- No deben haber relaciones redundantes ni entidades sin propósito.
- La opcionalidad de las relaciones es importante.
- La definición del problema no tiene que ser muy específica, para que el alumno se vea forzado a pensar en las posibles entidades que involucran un negocio de este tipo, para este ejemplo.
- Y en general se busca llegar a un nivel alto de funcionalidad con el modelo de datos.

Lab assignment 5. Realizar un modelo de datos, luego realizar una consulta sobre este modelo.

Especificaciones:

- Se busca que el estudiante tome un modelo de datos y lo cree físicamente en la base de datos de su elección.
- Se debe de utilizar de manera eficiente *tablespaces* e índices.
- Se provee al estudiante también una lista de datos para cada tabla y el estudiante debe poblar sus tablas creadas.
- Luego que tiene su diseño funcionando y los datos cargados el estudiante debe realizar una consulta.
- Esta consulta se deja como un repaso de lo visto en el curso inmediato anterior, se busca hacer un *join* involucrando 2 o 3 tablas.

- El objetivo detrás de dejar consultas sencillas en esta práctica es únicamente asegurarnos que la base de datos está funcionando bien con los datos cargados.
- El modelo que se crea al principio del semestre se utiliza a lo largo de las prácticas de laboratorio, por lo que cada semestre se debe crear un modelo nuevo.

Lab assignment 6. Realizar 2 consultas sobre el modelo de datos creado a principio de semestre.

Especificaciones:

- Se busca que el estudiante tenga claro cuando usar *group by* y porque se debe usar.
- En la primera consulta se busca que el estudiante utilice una de las funciones básicas y tenga que agrupar un campo.
- En la segunda consulta se buscan que el estudiante utilice una de las funciones básicas pero esta vez tenga que agrupar dos o más campos.

Lab assignment 7. Realizar 2 consultas sobre el modelo de datos creado a principio de semestre.

Especificaciones:

- Se busca que el estudiante tenga claro el concepto de sub consultas.
- En la primera consulta se busca que el estudiante comprenda que se pueden hacer restricciones tanto en el *where* como en el *from*, esta consulta se resuelve de ambas formas.

- En la segunda consulta se busca hacer una sub consulta compleja o una sub consulta anidada, también se busca crear restricciones complejas donde se tengan que usar operadores como AND y OR.

Lab assignment 8. Realizar 2 consultas sobre el modelo de datos creado a principio de semestre.

Especificaciones:

- Se busca que el estudiante use las cláusulas *HAVING* y cláusulas de comparación de rangos como *BETWEEN*, también que sea capaz de hacer ordenamientos.
- En la primera consulta se busca que el estudiante vea la necesidad de usar la cláusula *HAVING* al tener que usarla en esta consulta, luego se pide que se haga otra vez pero sin usar *HAVING* para ver la diferencia.
- En la segunda consulta se busca usar uno de los operadores siguientes: *IN*, *NOT IN*, *BETWEEN*, *NOT BETWEEN* y también que comience a ordenar los resultados mediante *ORDER BY*.

Lab assignment 9. Realizar 2 consultas sobre el modelo de datos creado a principio de semestre.

Especificaciones:

- Se busca que el estudiante logre usar las cláusulas *EXISTS* y *UNION*.
- En la primera consulta se busca que el estudiante use *EXISTS*.
- En la segunda consulta se busca que el estudiante use *NOT EXISTS*.

Lab Assignment 10. Realizar 2 consultas sobre el modelo de datos creado a principio de semestre.

Especificaciones:

- Esta consulta sirve como repaso para todos los conceptos vistos en clase.
- En la primera consulta se busca realizar una consulta compleja y que luego se unan los valores usando *UNION*.
- En esta consulta se busca también que el estudiante use campos estáticos.
- En la segunda consulta solamente se busca que el estudiante utilice funciones usando *group by* y aparte de esto ordene los datos, el estudiante debe usar la cláusula *having* a pesar de no decírselo.

Proyecto 1. El proyecto consiste en realizar un modelo de datos a partir de un enunciado, luego realizar una serie de consultas sobre dicho modelo.

Especificaciones:

- El grado de dificultad del modelo no es tan alto por lo tanto las consultas se facilitan un poco.
- El ambiente de trabajo es más amigable (SQL) esto facilita un poco las cosas.
- En general el grado de las restricciones no es tan alto.
- El número y complejidad de sub consultas a utilizar es intermedio.
- En varias consultas incluso no es necesario usar sub consultas.
- El grado de dificultad de algunas consultas radica en poder utilizar bien funciones para manejo de cadenas y fechas.

Proyecto 2. El proyecto consiste en realizar un modelo de datos a partir de un enunciado, luego realizar una serie de consultas sobre dicho modelo.

Especificaciones:

- El grado de dificultad sube al tener un modelo más complejo sobre el cual se tiene que trabajar.
- Se tiene un ambiente de trabajo más difícil para este proyecto (Oracle).
- En general el grado de las restricciones se vuelve más específico y complicado.
- El número y complejidad de sub consultas a utilizar se incrementa.
- En algunos casos es necesario usar *UNION*.

4. GUÍA DEL INSTRUCTOR DE SISTEMAS DE BASES DE DATOS 2

A continuación se presenta el desarrollo de la Guía del curso de Sistemas de Bases de Datos 2, de acuerdo al formato sugerido en el curso de “Estructuración de laboratorios”, impartido por el señor. Mrutunjaya Panda.

4.1. Datos generales

NOMBRE DE CURSO:	Sistemas de Bases de Datos 2 (775)
PRE- REQUISITOS:	Sistemas Operativos 1 (281) Sistemas de Bases de Datos 1 (774)
POST – REQUISITOS:	Bases de Datos Avanzadas (738)
OBJETIVO:	Al finalizar el curso, el estudiante podrá conocer la manera en que funciona internamente un DBMS, y realizar funciones de mantenimiento y administración de una base de datos. También conocerá el lenguaje PL/SQL y T-SQL.
MODULOS:	Bases de Datos

4.2. Distribución del curso

Tabla XI: Distribución curso Sistemas de Bases de Datos 2

MODULO	TEORÍA	PRACTICA	TOTAL	EXAMEN	LAB ASSIGNMENTS	PROYECTOS
BASES DE DATOS	34 horas	26 horas	60 horas	1	13	2
			60 horas			

4.3. Evaluación

Ponderación de tareas, prácticas y proyectos.

Tabla XII: Evaluación curso Sistemas de Bases de Datos 2

ACTIVIDAD	NUMERO	PORCENTAJE	TOTAL
PRACTICA DE LABORATORIO	13	1.5 %	19.50 %
EXAMEN	1	9.5 %	9.50 %
ASISTENCIA	16	0.0625 %	1.00 %
PROYECTO	2	35.0 %	70.00 %
TOTAL			100.00 %

4.4. Detalle de sesiones

Tabla XIII: Sesiones curso Sistemas de Bases de Datos 2

No	TEORIA	No	PRACTICA	TUTORIAL / PROYECTO / LAB ASSIGNMENT
1	Bases de Datos – 1	1	Modelado de BD y SQL 1	Práctica de Laboratorio 1
2	Bases de Datos – 2			
3	Bases de Datos – 3	2	Modelado de BD y SQL 2	Práctica de Laboratorio 2
4	Bases de Datos – 4	3	Modelado de BD y SQL 3	Práctica de Laboratorio 3
5	Bases de Datos – 5			
6	Bases de Datos – 6	4	Modelado de BD y SQL 4	Práctica de Laboratorio 4
7	Bases de Datos – 7	5	Modelado de BD y SQL 5	Primer Proyecto / Práctica de Laboratorio 5
8	Bases de Datos – 8	6	Modelado de BD y SQL 6	Práctica de Laboratorio 6
9	Bases de Datos – 9	7	Modelado de BD y SQL 7	Práctica de Laboratorio 7
10	Bases de Datos – 10	8	Modelado de BD y SQL 8	Práctica de Laboratorio 8
11	Bases de Datos - 11	9	Modelado de BD y SQL 9	Segundo Proyecto / Práctica de Laboratorio 9
12	Bases de Datos – 12	10	Modelado de BD y SQL 10	Práctica de Laboratorio 10
13	Bases de Datos – 13	11	Modelado de BD y SQL 11	Práctica de Laboratorio 11
14	Bases de Datos – 14	12	Modelado de BD y SQL 12	Práctica de Laboratorio 12
15	Bases de Datos – 15			
16	Bases de Datos – 16	13	Modelado de BD y SQL 13	Práctica de Laboratorio 13
17	Bases de Datos – 17			

4.5. Distribución de sesiones

Tabla XIV: Distribución de sesiones de Bases de Datos 2

SESIÓN NO. 1:	Gestión de Memoria Dinámica
OBJETIVO:	Mostrar al estudiante la manera en que un DBMS funciona internamente, la manera en que se optimizan los procesos por medio del uso de memoria dinámica.
CONTENIDO:	<ol style="list-style-type: none">1. Gestión de memoria dinámica.<ol style="list-style-type: none">1.1 ODM, SQLOS y <i>File manager</i>1.2 Funcionamiento del <i>buffer</i>1.3 Diferentes Listas1.4 <i>Buffer</i> y bloques1.5 Características importantes1.6 <i>Buffer Manager, Lazy Writer</i>
SESIÓN NO. 2:	Optimización
OBJETIVO:	Mostrar la manera en que el DBMS procesa las instrucciones y como afecta el desempeño.
CONTENIDO:	<ol style="list-style-type: none">1. Optimización<ol style="list-style-type: none">1.1 Fase de <i>Parser</i>

	<p>1.2 Fase de Ejecución</p> <p>1.3 Fase de <i>Fetch</i></p> <p>1.4 Ejemplo de plan de ejecución</p> <p>1.5 Optimizador por regla</p> <p>1.6 Mayores factores en el desempeño</p>
SESIÓN NO. 3:	Transacciones y el DBMS
OBJETIVO:	Mostrar al estudiante como se manejan las transacciones en diferentes lenguajes de programación y como los maneja el DBMS.
CONTENIDO:	<p>1. Transacciones y el DBMS</p> <p>1.1 Transacciones</p> <p>1.2 <i>Commit</i></p> <p>1.3 <i>Rollback</i></p> <p>1.4 <i>Savepoint</i></p> <p>1.5 Bitácora</p>
SESIÓN NO. 4:	Seguridad
OBJETIVO:	Introducir al estudiante al manejo de la seguridad al crear una base de datos.
CONTENIDO:	<p>1. Seguridad</p> <p>1.1 Seguridad</p> <p>1.2 Conceptos de Seguridad</p>

	<p>1.3 Seguridad a nivel de Base de Datos</p> <p>1.4 Esquema: Permisos</p> <p>1.5 Esquema: Roles</p> <p>1.6 Esquema: Usuarios</p>
SESIÓN NO. 5:	Introducción a PL/SQL
OBJETIVO:	Introducir al estudiante al lenguaje PL/SQL.
CONTENIDO:	<p>1. Introducción a PL/SQL</p> <p>1.1 PL/SQL</p> <p>1.2 Variables</p> <p>1.3 Operadores</p> <p>1.4 <i>IF, THEN ,ELSE</i></p> <p>1.5 <i>GOTO</i></p> <p>1.6 <i>LOOP, WHILE, FOR</i></p>
SESIÓN NO. 6:	Bloques en PL/SQL
OBJETIVO:	Explicar al estudiante las estructuras de bloques que se manejan para crear procedimientos, funciones, <i>triggers</i> , etc.
CONTENIDO:	<p>1. Bloques en PL/SQL</p> <p>1.1 Bloques en PL/SQL</p> <p>1.2 Procedimientos Almacenados</p> <p>1.3 Funciones</p>

	1.4 Crear procedimientos y funciones
SESIÓN NO. 7:	Cursores en PL/SQL
OBJETIVO:	Explicar al estudiante la manera en que se maneja la información por medio de cursores.
CONTENIDO:	<ol style="list-style-type: none"> 1. Cursores en PL/SQL <ol style="list-style-type: none"> 1.1 Cursores 1.2 Cursores Implícitos 1.3 Cursores Explícitos. 1.4 Navegando un Cursor
SESIÓN NO. 8:	Mas sobre bloques
OBJETIVO:	Mostramos el resto de las estructuras de bloques que existen y el manejo de parámetros y errores.
CONTENIDO:	<ol style="list-style-type: none"> 1. Excepciones <ol style="list-style-type: none"> 1.1 Manejo de excepciones 1.2 Levantar errores 2. <i>Triggers</i> <ol style="list-style-type: none"> 2.1 <i>Triggers</i> 2.2 <i>Triggers</i> OLD y NEW 3. Parámetros <ol style="list-style-type: none"> 3.1 Parámetros IN OUT

	<p>3.2 IN</p> <p>3.3 OUT</p> <p>3.4 IN OUT</p>
SESIÓN NO. 9:	Otros tipos de variables
OBJETIVO:	Se le muestra el estudiante el uso de variables más complejas
CONTENIDO:	<p>1.Otros tipos de variables</p> <p>1.1 Registros</p> <p>1.2 Tablas</p> <p>1.3 <i>Varrays</i></p> <p>1.4 <i>Packages</i></p>
SESIÓN NO. 10:	Introducción T-SQL
OBJETIVO:	Introducir al estudiante al lenguaje T-SQL.
CONTENIDO:	<p>1. Introducción T-SQL</p> <p>1.1 T-SQL</p> <p>1.2 Variables y Operadores</p> <p>1.3 Estructuras de control de flujo</p> <p>1.4 Manejo de errores</p> <p>1.5 Procedimientos</p>
SESIÓN NO. 11:	T-SQL 2

OBJETIVO:	Definir estructuras más complejas para la utilización del lenguaje T-SQL.
CONTENIDO:	<ol style="list-style-type: none"> 1. T-SQL 2 <ol style="list-style-type: none"> 1.1 Funciones 1.2 Funciones escalares, en línea 1.3 <i>Triggers</i> 1.4 Cursores
SESIÓN NO. 12:	<i>Backup y Recuperación</i>
OBJETIVO:	Se introduce al estudiante a la teoría básica detrás de la realización de <i>backups</i> y recuperaciones.
CONTENIDO:	<ol style="list-style-type: none"> 1. <i>Backup y Recuperación</i> <ol style="list-style-type: none"> 1.1 <i>Backups</i> 1.2 <i>Backup del Sistema Operativo</i> 1.3 <i>Backup en frio y en caliente</i> 1.4 <i>Backup lógico usando Import y Export</i> 1.5 Tipos de fallos 1.6 RMAN
SESIÓN NO. 13:	Recuperación y estrategias
OBJETIVO:	Mostramos al estudiante la manera en que se realizan recuperaciones de <i>backup</i> y mostramos estrategias de <i>backup</i> .
CONTENIDO:	<ol style="list-style-type: none"> 1. Recuperación

	<ul style="list-style-type: none"> 1.1 Recuperación 1.2 En caliente sin RMAN 1.3 En caliente con RMAN 1.4 En frio sin RMAN 1.5 En frio con RMAN 2. Estrategias de <i>backup</i> <ul style="list-style-type: none"> 2.1 <i>Backup</i> completo 2.2 <i>Backup</i> incremental 2.3 <i>Backup</i> diferencial
SESIÓN NO. 14:	<i>Backup</i> y recuperación en SQL
OBJETIVO:	Finalizar el concepto de <i>backup</i> y recuperación realizando dicho proceso en el DBMS SQL.
CONTENIDO:	<ul style="list-style-type: none"> 1. Introducción T-SQL <ul style="list-style-type: none"> 1.1 T-SQL 1.2 Variables y Operadores 1.3 Estructuras de control de flujo 1.4 Manejo de errores 1.5 Procedimientos
SESIÓN NO. 15:	Bases de datos multidimensionales
OBJETIVO:	Se introduce al estudiante a las bases de datos multidimensionales

CONTENIDO:	<p>1. Bases de datos multidimensionales</p> <p>1.1 Bases de datos multidimensionales</p> <p>1.2 <i>Business Intelligence</i></p> <p>1.3 Diseño de un <i>Data Warehouse</i></p> <p>1.4 ETL</p> <p>1.5 Dimensiones, Niveles y Métricas</p>
SESIÓN NO. 16:	Ejemplo de <i>Data Warehouse</i>
OBJETIVO:	Se muestra al estudiante un ejemplo práctico de la realización de un modelo completo de <i>Data Warehouse</i> .
CONTENIDO:	<p>1. Ejemplo práctico de <i>Data Warehouse</i></p> <p>1.1 Bases de Datos Multidimensionales</p> <p>1.2 <i>Business Intelligence</i></p> <p>1.3 Diseño de un <i>Data Warehouse</i></p> <p>1.4 ETL</p> <p>1.5 Dimensiones, Niveles y Métricas</p>
SESIÓN NO. 17:	Examen de Laboratorio
OBJETIVO:	Realizar el único examen de laboratorio teórico para el curso
CONTENIDO:	1. Examen de laboratorio.

4.6. Detalle de *lab assignments*

PRÁCTICAS:

Criterio de Evaluación:

- Completar el trabajo: 20
- Corrección del trabajo: 80

Documentación a entregar: SI

- Solución del problema
- Imágenes de la solución

Módulo: Bases de Datos

Tabla XV: Prácticas curso Sistemas de Bases de Datos 2

LAB ASSIGNMENT 1:	Repaso Base de Datos 1
OBJETIVO:	Repasar de forma práctica conceptos básicos de diseño de base de datos.
Iniciada en sesión	1
Entregada en sesión	2
Viva / Demostración	NO
LAB ASSIGNMENT 2:	Transacciones

OBJETIVO:	Aplicar de manera Práctica los conocimientos sobre transacciones en diferentes lenguajes de programación.
Iniciada en sesión	3
Entregada en sesión	4
Viva / Demostración	NO
LAB ASSIGNMENT 3:	Seguridad
OBJETIVO:	Repasar de forma práctica conceptos teóricos de seguridad vistos en clase por medio de la realización de un esquema completo.
Iniciada en sesión	4
Entregada en sesión	5
Viva / Demostración	NO
LAB ASSIGNMENT 4:	PL/SQL 1
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL.
Iniciada en sesión	6
Entregada en sesión	7
Viva / Demostración	NO
LAB ASSIGNMENT 5:	Modelado PL/SQL 2
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL.
Iniciada en sesión	7
Entregada en sesión	8

Viva / Demostración	NO
LAB ASSIGNMENT 6:	PL/SQL 3
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL.
Iniciada en sesión	8
Entregada en sesión	9
Viva / Demostración	NO
LAB ASSIGNMENT 7:	PL/SQL 4
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL.
Iniciada en sesión	9
Entregada en sesión	10
Viva / Demostración	NO
LAB ASSIGNMENT 8:	T-SQL
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL.
Iniciada en sesión	10
Entregada en sesión	11
Viva / Demostración	NO
LAB ASSIGNMENT 9:	T-SQL 2
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL.
Iniciada en sesión	11

Entregada en sesión	12
Viva / Demostración	NO
LAB ASSIGNMENT 10:	<i>Backup</i>
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL.
Iniciada en sesión	12
Entregada en sesión	13
Viva / Demostración	NO
LAB ASSIGNMENT 11:	Recuperación
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL.
Iniciada en sesión	13
Entregada en sesión	14
Viva / Demostración	NO
LAB ASSIGNMENT 12:	<i>Backup SQL</i>
OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL.
Iniciada en sesión	14
Entregada en sesión	15
Viva / Demostración	NO
LAB ASSIGNMENT 13:	Bases de Datos Multidimensionales

OBJETIVO:	Repasar de forma práctica conceptos teóricos vistos en clase para SQL.
Iniciada en sesión	16
Entregada en sesión	17
Viva / Demostración	NO

4.7. Detalle de proyectos

PROYECTO 1:

Criterio de Evaluación:

- Funcionamiento: 70
- Conocimiento Técnico: 10
- Documentación: 20

Documentación a entregar: SI

- Enunciado detallado en *lab handbook*

Módulo: Bases de Datos

Tabla XVI: Proyecto curso Sistemas de Bases de Datos 2

PROYECTO 1:	
OBJETIVO:	Realizar una aplicación completa en la cual se haga un uso pesado de la base de datos. Se busca que el estudiante haga un análisis completo de la aplicación.

Iniciada en sesión	7
Entregada en sesión	10
Viva / Demostración	SI
PROYECTO 2:	
OBJETIVO:	Realizar una serie de configuraciones complicadas sobre la base de datos. Se busca comprobar las habilidades del estudiante en cuanto a configuraciones complejas sobre la base de datos.
Iniciada en sesión	11
Entregada en sesión	14
Viva / Demostración	SI

4.8. Detalle de exámenes

Examen 1. Teoría del módulo completo de bases de datos.

- Entregado en sesión: 13
- Duración: 1 hora.
- Criterio de Evaluación: objetivo 60%, subjetivo 40%.

4.9. Examen de ejemplo

Figura 2: Examen ejemplo de Bases de Datos 2

Instrucciones: Responda lo que se le solicita, o indique a que corresponde la definición que se presenta. Puede ser posible que la respuesta no aplique.

1. Propiedad ACID que indica que una vez una transacción es confirmada, sus actualizaciones sobreviven aun cuando hayan caídas posteriores del sistema:
2. Unidad de Paralelismo e Integridad:
3. Restaura una base de datos a partir de un vaciado previo y luego sin usar la bitácora recupera el trabajo que se hizo desde el vaciado.
4. Propiedad ACID que no obedece a todas las restricciones de integridad:
5. Esta correctamente escrita el statement: `SELECT employee_name FROM employees WHERE employee_id = 123 FOR UPDATE OF employee_name?`
6. Unidad de Trabajo Lógica:
7. El atributo del extend de Oracle que permite la compresión del archivo de respaldo como un archivo .zip:
8. Modos de backup que se puede realizar con export/import en Oracle:
9. Es recomendable hacer un export de una partición Linux a una partición NTFS?
10. Un export puede ser incremental, acumulativo y completo?

4.10. Bibliografía sugerida

1. C.J. Date, **Introducción a los sistemas de bases de datos.**
2. Andy Oppel, ***Databases demystified a self teaching guide.***
3. Elmasri y Navathe. **Sistemas de Bases de datos.**
4. Ron Soukup, **Microsoft SQL Server a Fondo.**
5. Kevin Loney. ***Oracle Database 10g: the complete reference.***

6. Almeida, Ishikawa, Reinschmidt y Roeber. ***Getting started with Data Warehouse and Business Intelligence.***
7. Fon Silvers, ***Building and maintaining a Data Warehouse.***

4.11. Lab handbook

Lab assignment 1. Realizar un modelo de datos complejo.

Especificaciones:

- El modelo de datos no debe ser especificado mucho, el objetivo es dejar al estudiante una labor de investigación.
- Únicamente se busca ver la habilidad del estudiante para realizar modelos a este nivel.
- El estudiante debe ser capaz de manejar eficientemente *tablespaces* y tener un usuario para conectarse a consola.

Lab assignment 2. Crear una base de datos y realizar un modelo de datos sencillo de 2 o más entidades.

Especificaciones:

- El modelo sobre el cual se debe trabajar puede ser especificado o dejar a discreción del estudiante, no es importante.
- Se busca que el estudiante experimente con los procesos de *commit* y *rollback* de una transacción bajo diferentes lenguajes.
- Se recomienda hacer el ejercicio en 2 lenguajes diferentes.
- El DBMS que se recomienda usar es Oracle.

Lab assignment 3. Tomar como base el modelo de datos realizado en el ejercicio 1 y sobre este realizar un esquema de seguridad.

Especificaciones:

- Se busca tener un modelo de datos grande sobre el cual trabajar.
- Sobre este modelo de datos se busca que el estudiante realice un esquema de seguridad completo.
- Lo más importante es que defina bien que roles existen y que permisos tienen asignados.

Lab assignment 4. Crear un procedimiento almacenado y una función.

Especificaciones:

- Se busca crear un procedimiento almacenado y una función en PL/SQL.
- En este momento todavía no se debe involucrar una base de datos.
- Es importante ver tanto el código del procedimiento como un *screenshot* del mismo funcionando, ya que el objetivo de la practica es lograr que el estudiante realice un procedimiento y sea capaz de ejecutarlo.
- Las operaciones que deben realizarse deben de ser de dificultad fácil a intermedia.
- Se busca que el estudiante use estructuras como *IF*, *CASE*, *LOOP*.
- Debe mandar *screenshots* del procedimiento ejecutándose y el script del ejercicio completo.

Lab assignment 5. Crear un procedimiento almacenado y una función.

Especificaciones:

- En la primera parte se busca crear una función que deba recorrer una tabla del modelo provisto y regresar un valor.
- En la segunda parte el objetivo es hacer que el estudiante haga una serie de búsquedas en una o más tablas, guarde los valores temporalmente y con esto cree un nuevo registro en otra tabla.
- Todo debe ser realizado por medio de procedimientos almacenados.
- Se busca también que el estudiante use funciones internas como la fecha del sistema y una secuencia para generar el número de factura automático.
- El modelo no tiene que ser muy complicado para enfocarse en la programación en PL.
- Debe mandar *screenshots* del procedimiento ejecutándose y el script del ejercicio completo.

Lab assignment 6. Realizar una serie de ajustes y verificaciones sobre procedimientos ya hechos.

Especificaciones:

- Se busca que el estudiante sea capaz de manejar errores eficazmente y también pueda levantar errores para mantener el control del programa.
- Realizar un *trigger* sobre cualquier tabla en cualquier evento, se busca que se realice cualquier tipo de acción pero que se estén utilizando los datos del evento.

- Debe mandar *screenshots* del procedimiento ejecutándose y el script del ejercicio completo.

Lab assignment 7. Crear un *package* en PL/SQL y realizar una serie de funciones sobre este.

Especificaciones:

- Se busca que el estudiante sea capaz de crear un *package* y pueda crear tanto variables como procedimientos.
- Debe incluir dentro del *package* todas las funciones y procedimientos que se han creado durante los ejercicios para PL.
- Se debe hacer también un procedimiento extra en los cuales se tenga que usar parámetros de tipo *IN*, *OUT*, *IN/OUT*.
- En este procedimiento para buscar los datos se debe usar una de las estructuras especiales vistas en esta sesión.
- Debe mandar *screenshots* del procedimiento ejecutándose y el script del ejercicio completo.

Lab assignment 8. Crear dos procedimientos en T-SQL.

Especificaciones:

- Se busca que el estudiante sea capaz de crear manejar diferentes tipos de variables y operadores básicos
- Se busca también que utilice de manera eficiente las diferentes estructuras de flujo.
- Se busca que el estudiante sea capaz de crear procedimientos sencillos y que pueda mostrar respuestas en consola.
- Se busca que pueda manejar errores de una manera eficiente.

- Debe mandar *screenshots* del procedimiento ejecutándose y el script del ejercicio completo.

Lab assignment 9. Crear dos procedimientos en T-SQL.

Especificaciones:

- Se busca que el estudiante pueda manejar los diferentes tipos de funciones que existen y que también sea capaz de utilizar cursores para navegar a través de los datos.
- Se busca que pueda manejar estructuras más complejas como triggers y que sea capaz de manipular los datos anteriores y los datos posteriores.
- Debe mandar *screenshots* del procedimiento ejecutándose y el script del ejercicio completo.

Lab assignment 10. Realizar 4 diferentes tipos de *backups* en Oracle.

Especificaciones:

- Se busca que el estudiante comprenda y sea capaz de realizar diferentes tipos de *backup*.
- La base de datos sobre la cual se realizan los *backups* debe ser cualquier modelo creado en Oracle.

Lab assignment 11. Realizar 4 tipos diferentes de recuperaciones en Oracle.

Especificaciones:

- Se busca que el estudiante comprenda y sea capaz de realizar diferentes tipos de recuperaciones a partir de un *backup* específico.

- La base de datos sobre la cual se realizan los *backups* debe ser cualquier modelo creado en Oracle.
- Se recomienda que como parte de la calificación del primer proyecto, se exija al estudiante que realice un *backup* y una recuperación de su elección durante la calificación.

Lab assignment No. 12. Realizar diferentes tipos de *backup* y recuperaciones sobre SQL Server.

Especificaciones:

- Cimentar los principios sobre *backups* y recuperación vistos anteriormente.
- Que cada estudiante experimente personalmente como realizar los diferentes tipos de *backups* que existen, por medio de esta investigación con los ejemplos incluidos.

Lab assignment No. 13. Realizar una investigación sobre las bases de datos multidimensionales.

Especificaciones:

- Cimentar los conocimientos y conceptos dados sobre el tema.

Proyecto 1. Implementar una aplicación compleja que tenga un uso intensivo de la base de datos.

Especificaciones:

- Aquí estamos tomando un problema complejo y debemos crear la solución desde los cimientos.
- Se debe crear un modelo de datos para soportar el problema.

- Se debe crear una aplicación *web* para el problema y utilizar la base de datos.
- Es necesario utilizar PL/SQL para facilitar los requerimientos del problema.
- Se debe hacer un esquema de seguridad riguroso e implantarlo para la solución del problema.
- También se deben realizar consultas complejas en algunos procedimientos almacenados, aquí se verifica la efectividad del modelo de datos.
- Es importante tener la capacidad de recuperar los datos del sistema por medio de un *backup* y una recuperación sobre la base de datos.
- Se da un mes para la realización del proyecto.

Proyecto 2. Implementar un ambiente de alta disponibilidad sobre una plataforma específica.

Especificaciones:

- El objetivo de este proyecto es implantar ambiente de alta disponibilidad para el DBMS Oracle bajo el sistema operativo Linux de la elección del estudiante.
- Se busca lograr configurar de manera exitosa en el sistema operativo todos los requerimientos descritos arriba.
- Se estará implantando un *cluster* y debe poderse realizar un procedimiento de balanceo de carga para la base de datos y un procedimiento de *fail over*.
- Este proyecto toma conocimientos teóricos adquiridos en la clase y los aplica a un proyecto orientado a configurar la base de datos.
- Se da como tiempo aproximadamente un mes para realizar el proyecto.

5. GUÍA DEL INSTRUCTOR DE ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 2

A continuación se presenta el desarrollo de la Guía del curso de Arquitectura de Computadores y Ensambladores 2, de acuerdo al formato sugerido en el curso de “Estructuración de laboratorios”, impartido por el señor. Mrutunjaya Panda.

5.1. Datos generales

NOMBRE DE CURSO: **Arquitectura de computadores y Ensambladores 2 (779)**

PRE- REQUISITOS: Arquitectura de computadores y Ensambladores 1 (778)

POST – REQUISITOS: Ninguno

OBJETIVO: Introducir al estudiante a la administración de proyectos complejos de tecnología y adquirir los conocimientos necesarios para la correcta administración de un proyecto en el cual se debe integrar componentes de software con hardware.

MÓDULOS: Administración de Tecnología e Innovación

5.2. Distribución del curso

Tabla XVII: Distribución curso Arquitectura de Computadores y Ensambladores 2

MODULO	TEORÍA	PRACTICA	TOTAL	EXAMEN	LAB ASSIGNMENTS	PROYECTOS
ADMINISTRACION DE TECNOLOGIA EN INNOVACION	28 horas	6 horas	34 horas	1	3	1
			28 horas			

5.3. Evaluación

Ponderación de tareas, prácticas y proyectos.

Tabla XVIII: Evaluación Arquitectura de Computadores y Ensambladores 2

ACTIVIDAD	NUMERO	PORCENTAJE	TOTAL
PRACTICAS	3	5.0 %	15.00 %
PROYECTO DE CLASE	1	70.0 %	70.00 %
EXAMEN	1	15.0 %	15.00 %
		TOTAL	100.00 %

5.4. Detalle de sesiones

Tabla XIX: Sesiones curso Arquitectura de Computadores y Ensambladores 2

No	TEORIA	No	PRACTICA	TUTORIAL / PROYECTO / LAB ASSIGNMENT
1	Admón. de tecnología e Innovación – 1			
2	Admón. de tecnología e Innovación – 2			
3	Admón. de tecnología e Innovación – 3			Práctica Ensayo 1
4	Admón. de tecnología e Innovación – 4			
5	Admón. de tecnología e Innovación – 5			
6	Admón. de tecnología e Innovación – 6			Práctica Ensayo 2
7	Admón. de tecnología e Innovación – 7			
8	Admón. de tecnología e Innovación – 8			
9	Admón. de tecnología e Innovación – 9			Práctica Ensayo 3
10	Admón. de tecnología e Innovación – 10			
11	Admón. de tecnología e Innovación – 11			
12	Admón. de tecnología e Innovación – 12			
13	Admón. de tecnología e Innovación – 13			
14	Admón. de tecnología e Innovación – 14			

5.5. Distribución de sesiones

Tabla XX: Distribución de sesiones de Ensambladores 2

SESIÓN NO. 1:	Introducción a la administración de tecnología
OBJETIVO:	Presentar el concepto de tecnología y el impacto tecnológico.
CONTENIDO:	<ol style="list-style-type: none"> 1. Introducción a la administración de tecnología <ol style="list-style-type: none"> 1.1 Administración de la tecnología 1.2 Tecnología 1.3 Impacto Tecnológico 1.4 Tangible e Intangible 1.5 El medio es el mensaje
SESIÓN NO. 2:	Interacción con las personas
OBJETIVO:	Explica la naturaleza y variantes en la interacción con las personas y su impacto en el desarrollo de la innovación
CONTENIDO:	<ol style="list-style-type: none"> 1. Interacción con las personas <ol style="list-style-type: none"> 1.1. Percepción 1.2. El medio 1.3. Interacción con las personas 1.4. Fuentes de información
SESIÓN NO. 3:	Aldea Global
OBJETIVO:	Mostrar el impacto que tiene la tecnología en la sociedad y la manera en que ha afectado el desarrollo de la comunicación global.

CONTENIDO:	<p>1. Aldea Global</p> <p>1.1 Implosión social.</p> <p>1.2 Aldea Global.</p> <p>1.3 Medios fríos y calientes.</p> <p>1.4 El impacto de los medios.</p>
SESIÓN NO. 4:	<i>Leonardo's Laptop</i>
OBJETIVO:	Explicar el concepto de psicología de las cosas y los elementos que lo conforman y mostrar la importancia de la creatividad y la innovación
CONTENIDO:	<p>1. <i>Leonardo's Laptop</i></p> <p>1.1 Psicología de las cosas</p> <p>1.2 Psicología conceptos</p> <p>1.3 El diseño</p> <p>1.4 Creatividad e Innovación</p>
SESIÓN NO. 5:	<i>Leonardo's Laptop 2</i>
OBJETIVO:	Mostrar la importancia de la usabilidad en el desarrollo de cualquier tecnología.
CONTENIDO:	<p>1. <i>Leonardo's Laptop 2</i></p> <p>1.1 Usabilidad</p> <p>1.2 New Computing</p> <p>1.3 Usabilidad Universal</p> <p>1.4 Variedad Tecnológica</p>
SESIÓN NO. 6:	Visión

OBJETIVO:	Explicar la importancia de la visión tanto para una organización o grupo como para una persona.
CONTENIDO:	<ol style="list-style-type: none"> 1. Visión <ol style="list-style-type: none"> 1.1. Visión 1.2. Las personalidades 1.3. Estudio 1.4. La sociedad y el lenguaje
SESIÓN NO. 7:	La voluntad del sentido
OBJETIVO:	Explicar la naturaleza del hombre de buscar la felicidad y el sentido de su vida, ver el punto de vista de 4 filósofos.
CONTENIDO:	<ol style="list-style-type: none"> 1. La voluntad del sentido <ol style="list-style-type: none"> 1.1. Víctor Frankl 1.2. Satisfacción, Freud 1.3. Satisfacción, Adler 1.4. Pirámide de Maslow
SESIÓN NO. 8:	El diseño de las cosas
OBJETIVO:	Explica conceptos sobre la creación del diseño de las cosas y mostrar como un buen diseño puede influir en varios aspectos diferentes.
CONTENIDO:	<ol style="list-style-type: none"> 1. El diseño de las cosas <ol style="list-style-type: none"> 1.1. Como hacerlo 1.2. <i>Matrix organization</i> 1.3. Comunicación, Técnica 1.4. Transparencia

SESIÓN NO. 9:	Tecnología Combinatoria
OBJETIVO:	Explicar el concepto de tecnología combinatoria y mostrar diferentes casos y áreas en los cuales se aplica.
CONTENIDO:	<ol style="list-style-type: none"> 1. Tecnología Combinatoria <ol style="list-style-type: none"> 1.1 Tecnología combinatoria 1.2 Un buen producto 1.3 Casos reales 1.4 Pruebas SW Distribuido
SESIÓN NO. 10:	Sistemas
OBJETIVO:	Mostrar los diferentes aspectos de un sistema y ver diferentes tipos de sistemas y su interacción con el medio.
CONTENIDO:	<ol style="list-style-type: none"> 1. Sistemas <ol style="list-style-type: none"> 1.1 Inteligencia 1.2 Sistema Mecánico 1.3 Humano 1.4 Sistema Automatizado
SESIÓN NO. 11:	Automatización
OBJETIVO:	Explicar el proceso de la automatización industrial y ver los diferentes métodos para lograr la automatización.
CONTENIDO:	<ol style="list-style-type: none"> 1. Automatización <ol style="list-style-type: none"> 1.1 Automatización Industrial 1.2 Equipo electrónico

	<p>1.3 Robots</p> <p>1.4 Impacto Social</p>
SESIÓN NO. 12:	PIC
OBJETIVO:	Mostrar que es un PIC y como se utiliza, mostrar las características más importantes de RISC.
CONTENIDO:	<p>1. PIC</p> <p>1.1. PIC</p> <p>1.2. Arquitectura Harvard</p> <p>1.3. RISC</p> <p>1.4. Control por registros</p> <p>1.5. Simulador</p>
SESIÓN NO. 13:	PLC
OBJETIVO:	Mostrar que es un PLC y la manera en que funciona así como los lenguajes de programación que utiliza.
CONTENIDO:	<p>1. PLC</p> <p>1.1 PLC</p> <p>1.2 T100MD1616+ PLC</p> <p>1.3 LADDER</p> <p>1.4 TBASIC</p> <p>1.5 Simulador</p>
SESIÓN NO. 14:	Examen Final

OBJETIVO:	Realizar el examen final del curso
CONTENIDO:	1. Examen Final

5.6. Detalle de *lab assignments*

PRÁCTICAS:

Criterio de Evaluación:

- Completar el trabajo 20 %
- Corrección del trabajo 80 %

Documentación a entregar: SI

- Resumen de la lectura en forma de ensayo

Módulo: Administración de Tecnología e Innovación.

Tabla XXI: Prácticas curso Arquitectura de Computadores y Ensambladores 2

LAB ASSIGNMENT 1:	Ensayo 1
OBJETIVO:	Reforzar el contenido de clase realizando una lectura extra relacionada al contenido, luego de haber leído la lectura extra se debe realizar un ensayo de un mínimo de 2 hojas sobre el tema leído.
Iniciada en sesión	3

Entregada en sesión	4
Viva / Demostración	NO
LAB ASSIGNMENT 2:	Ensayo 2
OBJETIVO:	Reforzar el contenido de clase realizando una lectura extra relacionada al contenido, luego de haber leído la lectura extra se debe realizar un ensayo de un mínimo de 2 hojas sobre el tema leído.
Iniciada en sesión	6
Entregada en sesión	7
Viva / Demostración	NO
LAB ASSIGNMENT 3:	Ensayo 3
OBJETIVO:	Reforzar el contenido de clase realizando una lectura extra relacionada al contenido, luego de haber leído la lectura extra se debe realizar un ensayo de un mínimo de 2 hojas sobre el tema leído.
Iniciada en sesión	9
Entregada en sesión	10
Viva / Demostración	NO

5.7. Detalle del proyecto

PROYECTO:

Criterio de Evaluación:

- Funcionamiento 70 %
- Conocimiento Técnico 10 %

- Documentación 20 %

Documentación a entregar: SI

- Enunciado

Módulo: Administración de Tecnología e Innovación

Tabla XXII: Proyecto curso Arquitectura de Computadores y Ensambladores 2

PROYECTO 1:	
OBJETIVO:	Aplicar los conocimientos adquiridos sobre la administración de la tecnología para crear un proyecto en un grupo grande de trabajo, tiene como objetivo unir aspectos de hardware y de software en el mismo proyecto.
Iniciada en sesión	1
Entregada en sesión	14
Viva / Demostración	SI

5.8. Detalle del examen

Examen 1. Todo el contenido de Administración de Tecnología e Innovación

- Duración: 1 hrs.
- Entregado en: última sesión.
- Criterio de Evaluación: subjetivo 100%

5.9. Examen de ejemplo

Figura 3: Examen ejemplo de Ensambladores 2

A continuación se le presentan 5 temas, usted debe escoger 3 de estos cinco temas y desarrollarlos entre 10 y 15 líneas. No se darán puntos extras por desarrollar más de tres temas.

Temas:

1. La aldea global: implosión social, estado de soledad, el cubo de McLuhan.
2. Leonardo's Laptop: psicología de las cosas, el diseño, creatividad e innovación.
3. La voluntad del sentido: Frankl, Freud, Adler, la pirámide de Maslow.
4. Tecnología combinatoria: un buen producto, casos reales, pruebas SW distribuido.
5. Automatización: automatización industrial, equipo electrónico, robots, impacto social.

5.10. Bibliografía sugerida

1. Ben Schneiderman, Leonardo's Laptop: Human Needs and the new Computing Technologies.
2. Marshall McLuhan, The Medium is the message.
3. Peter L. Berger, Construcción Social de la Realidad.

5.11. Lab handbook

Lab assignment 1. Reforzar el contenido de clase realizando una lectura extra relacionada al contenido, luego se debe realizar un ensayo de un mínimo de 2 hojas.

Especificaciones:

- Complementar el contenido base con un artículo relacionado al tema de interés.
- Comprobar la comprensión del estudiante sobre el tema en cuestión.
- Desarrollar la habilidad en el estudiante de expresarse en forma escrita sobre un tema relacionado a tecnología.

Lab assignment 2. Reforzar el contenido de clase realizando una lectura extra relacionada al contenido, luego se debe realizar un ensayo de un mínimo de 2 hojas.

Especificaciones:

- Complementar el contenido base con un artículo relacionado al tema de interés.
- Comprobar la comprensión del estudiante sobre el tema en cuestión.
- Desarrollar la habilidad en el estudiante de expresarse en forma escrita sobre un tema relacionado a tecnología.

Lab assignment 3. Reforzar el contenido de clase realizando una lectura extra relacionada al contenido, luego se debe realizar un ensayo de un mínimo de 2 hojas.

Especificaciones:

- Complementar el contenido base con un artículo relacionado al tema de interés.
- Comprobar la comprensión del estudiante sobre el tema en cuestión.
- Desarrollar la habilidad en el estudiante de expresarse en forma escrita sobre un tema relacionado a tecnología.

Proyecto 1. Crear un proyecto en un grupo grande de trabajo, con el objetivo de unir aspectos de hardware y de software en el mismo proyecto.

Especificaciones

- Desarrollar en el estudiante la habilidad de trabajar en equipo, especialmente en grupos grandes de trabajo.
- Comprobar las habilidades de planeación que tiene el estudiante.
- Cimentar los conocimientos de electrónica obtenidos en el curso anterior al aplicarlos en el proyecto.
- Hacer ver al estudiante que el hardware y el software siempre trabajan en conjunto, y que ambos aspectos son de igual importancia.

6. DOCUMENTACIÓN DE APOYO DE “SISTEMAS DE BASES DE DATOS 2”

A continuación se presenta el desarrollo de la documentación de apoyo del curso de Sistemas de Bases de Datos 2, el cual se realizó en formato de un libro conteniendo las unidades más importantes impartidas a lo largo del curso.

6.1 Datos generales

NOMBRE DE CURSO: Sistemas de Bases de Datos 2 (775)

PRE- REQUISITOS: Sistemas Operativos 1 (281)
Sistemas de Bases de Datos 1 (774)

POST – REQUISITOS: Bases de Datos Avanzadas (738)

OBJETIVO: Este libro está destinado para los estudiantes del curso de Sistemas de Bases de Datos 2, conteniendo teoría, ejemplos, diagramas, resumen, glosario y artículos de interés relacionados a cada una de las unidades del curso.

MÓDULO: Bases de Datos

Figura 4 – Páginas 1-4 Libro “Bases de Datos 2”

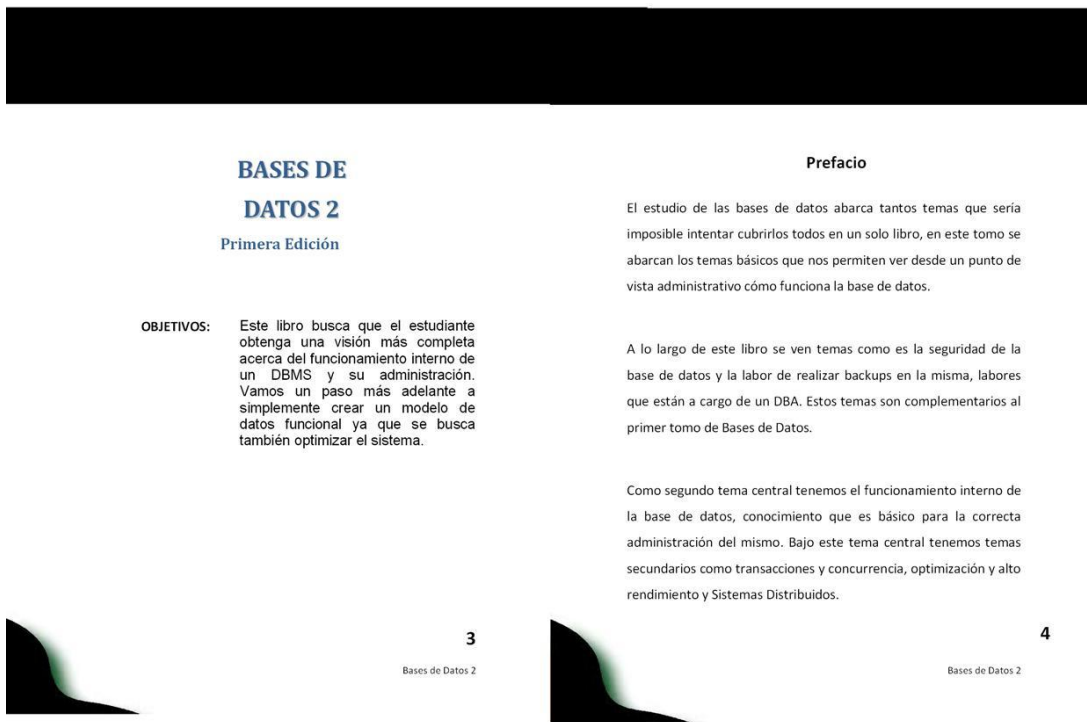


Figura 5 – Páginas 5-8 Libro “Bases de Datos 2”

Se cierra el libro con el tema de bases de datos multidimensionales, el cual se ve desde un punto de vista práctico ya que entrar en detalle con los aspectos técnicos del mismo son el objetivo de otro libro.

Este libro va dirigido a aquellas personas que necesitan un enfoque teórico para el mejor entendimiento del funcionamiento de una base de datos con el fin de lograr una mejor administración de la misma.

Prerrequisitos

Se debe tener un conocimiento básico sobre estructuras de datos, bases de datos, creación de modelos de datos y nociones de optimización. Por lo cual se recomienda leer el tomo numero uno de Bases de Datos.

CONTENIDO

1. Capítulo 1 – Transacciones y control de concurrencia	12
A. Alcance	13
1.1 Transacciones	14
1.1.1 Concepto de Transacción	14
1.1.2 ACID	16
1.1.3 Estado de la Base de datos	17
1.1.4 Puntos de sincronización	18
1.1.5 Bitácora	19
1.2 Concurrencia en Base de datos	20
1.2.1 Problemas de concurrencia	20
1.2.1.1 El problema del update perdido	20
1.2.1.2 El problema de dependencia no cometida	21
1.2.1.3 El problema de análisis inconsistente	22
1.2.2 Canchados	23
1.2.3 Deadlocks	25
B. Artículos relacionados	27
2. Capítulo 2 – Backup y Recuperación	28
A. Alcance	29
2.1 Respaldo y recuperación	30
2.1.1 Clasificación de fallos	30
2.1.1.1 Fallos de Hardware	30
2.1.1.2 Fallos de Software	30
2.1.1.3 Fallos de usuario	31



5

Bases de Datos 2



6

Bases de Datos 2

2.1.2 Tipos de backup	32
2.1.2.1 Archivos de sistema	32
2.1.2.2 Backup de datos	33
2.1.2.3 Naturaleza del backup	34
2.1.3 Recuperación basada en bitácora	35
2.1.3.1 Bitácora	35
2.1.3.2 Recuperación	37
2.2 Alta disponibilidad	37
2.2.1 Diferentes capas de almacenamiento y manejo de datos	40
2.2.2 Servidores en standby	43
2.2.2.1 Standby lógico	44
2.2.2.2 Standby físico	45
2.2.3 Servidores bajo cluster	45
B. Artículos relacionados	46
3. Capítulo 3 – Seguridad	48
A. Alcance	49
3.1 Seguridad	50
3.1.1 Mecanismos de seguridad	50
3.1.2 Otorgar y revocar permisos	51
3.2 Conceptos de seguridad	51
3.2.1 Identificación	51
3.2.2 Autenticación	52
3.2.3 Confidencialidad	52
3.2.4 Encriptación de datos	52
3.3 Consideraciones de seguridad	53
3.3.1 Seguridad Física	54



7

Bases de Datos 2

3.3.2 Seguridad de red	55
3.3.3 Seguridad a nivel de sistema	55
3.3.4 Seguridad a nivel de base de datos	56
3.3.4.1 Privilegios	56
3.3.4.2 Privilegios de sistema	57
3.3.4.3 Privilegios de objetos	57
3.3.4.4 Roles	57
B. Artículos relacionados	59
4. Capítulo 4 – Optimización y Alto Rendimiento	60
A. Alcance	61
4.1 Optimización de consultas	62
4.1.1 El proceso de ejecución de consultas	62
4.1.1.1 Parser	62
4.1.1.2 Plan de ejecución	63
4.1.1.3 Fetch	64
4.1.2 Optimizador por regla	64
4.1.2.1 Análisis de una instrucción	64
4.1.2.2 Tabla Pivot	66
4.1.2.3 Hints	67
4.1.3 Optimizador por costo	68
B. Artículos relacionados	69
5. Capítulo 5 – Sistemas Distribuidos	71
A. Alcance	72
5.1 Sistemas Distribuidos	74
5.1.1 Las 12 reglas de Codd	74
5.1.1.1 Autonomía local	74



8

Bases de Datos 2

Figura 6 – Páginas 9-12 Libro “Bases de Datos 2”



5.1.1.2 No dependencia de un sitio central	75
5.1.1.3 Operación continua	76
5.1.1.4 Independencia con respecto a la localización	76
5.1.1.5 Independencia con respecto a la fragmentación	76
5.1.1.6 Independencia de replica	77
5.1.1.7 Procesamiento distribuido de consultas	77
5.1.1.8 Manejo distribuido de transacciones	78
5.1.1.9 Independencia con respecto al equipo	78
5.1.1.10 Independencia con respecto al sistema operativo	78
5.1.1.11 Independencia con respecto a la red	79
5.1.1.12 Independencia con respecto al DBMS	79
5.1.2 Problemas de los sistemas distribuidos	79
5.1.2.1 Procesamiento de consultas	79
5.1.2.2 Administración del catalogo	80
5.1.2.3 Propagación de actualizaciones	80
5.1.2.4 Control de recuperación	81
5.1.2.5 Control de concurrencia	81
5.1.3 Commit de dos fases	81
5.2 Diseño de sistemas distribuidos	82
5.2.1 La distribución de los datos	83
5.2.1.1 Centralizada	83
5.2.1.2 Distribuida	83
5.2.1.3 Replicación	83
5.2.2 Modelos de replicación	84
5.2.2.1 Localidad primaria	84
5.2.2.2 Localidad secundaria	84

9

Bases de Datos 2



5.2.3 Replicación simétrica y asimétrica	85
5.2.3.1 Replicación asimétrica	85
5.2.3.2 Conflictos de replicación	85
5.2.3.3 Replicación simétrica	86
B. Artículos relacionados	86
6. Capítulo 6 – Análisis Multidimensional y Data Warehouse	88
A. Alcance	89
6.1 Bases de Datos Multidimensionales	90
6.1.1 OLAP, ROLAP, MOLAP	90
6.1.2 Business Intelligence	91
6.2 Data Warehouse	92
6.2.1 Data Warehouse y Data Mart	92
6.2.1.1. Data Warehouse	92
6.2.2.2. Data Mart	92
6.2.2 Diseño de un Data Warehouse	93
6.2.2.1 Modelo Estrella	93
6.2.2.2 Modelo Snow Flake	94
6.2.3 ETL	94
6.3 Componentes de un Data Warehouse	96
6.3.1 Fact Table	97
6.3.2 Dimensiones	97
6.3.3 Niveles	98
6.4 Operando con el Data Warehouse	99
6.4.1 DSS	99
6.4.2 Data Mining	100
B. Artículos relacionados	101

10

Bases de Datos 2



I. Glosario	102
II. Referencias Bibliográficas	104
III. Apéndice A: Licencia	106

ILUSTRACIONES

I. Figuras	
Figura No. 1 – Transacción	15
Figura No. 2 – Update Perdido	20
Figura No. 3 – Dependencia no cometida	21
Figura No. 4 – Análisis inconsistente	22
Figura No. 5 – Deadlock update perdido	25
Figura No. 6 – Porcentajes de disponibilidad	38
Figura No. 7 – Causas de fallos	39
Figura No. 8 – 6 capas de almacenamiento y manejo de datos	43
Figura No. 9 – Orden de procesamiento	65
Figura No. 10 – Instrucción discriminante	65
Figura No. 11 – Tabla Pivot	66
Figura No. 12 – Modelo Estrella	93
Figura No. 13 – Modelo Snow Flake	94
Figura No. 14 – Fuentes de Datos	96
Figura No. 15 – Dimensión en niveles	97
Figura No. 16 – Dimensión	98

11

Bases de Datos 2



Capítulo 1

TRANSACCIONES Y CONTROL DE CONCURRENCIA

Este capítulo presentamos el concepto de transacción, se describe la manera en que el DBMS maneja las transacciones y como estas afectan la base de datos. El segundo tema es la concurrencia y como esta afecta la base de datos, vemos algunos casos en los cuales existe concurrencia y como se puede manejar.

12

Bases de Datos 2



Figura 7 – Páginas 13-16 Libro “Bases de Datos 2”

ALCANCE

- 1. Transacciones y control de concurrencia
 - 1.1. Transacciones
 - 1.1.1. Concepto de Transacción
 - 1.1.2. ACID
 - 1.1.3. Estado de la Base de datos
 - 1.1.4. Puntos de sincronización
 - 1.2. Concurrencia en Base de datos
 - 1.2.1. Problemas de concurrencia
 - 1.2.1.1 El problema del update perdido
 - 1.2.1.2 El problema de dependencia no cometida
 - 1.2.1.3 El problema de análisis inconsistente
 - 1.2.2. Candados
 - 1.2.3. Deadlocks

1.1 Transacciones

1.1.1 Concepto de Transacción

Una transacción es un conjunto de operaciones que están delimitadas por un inicio y un fin, este conjunto de instrucciones se ejecutan de manera atómica por lo que no pueden ser divididas. Llevan a la base de datos de un estado consistente a otro estado consistente, pero no conserva la consistencia necesariamente en puntos intermedios de la transacción.

El inicio se da cuando se ejecuta la primera instrucción del conjunto. El fin de la transacción se da después de ejecutarse la última instrucción y hacer un commit.

Otra manera en que puede terminar una transacción es cuando se da un error, cuando sucede esto se regresa al estado inicial de la base de datos, antes de la transacción. De manera que si el error ha ocurrido luego de haberse ejecutado dos de las instrucciones de la transacción, estas instrucciones serán ejecutadas de manera inversa para deshacer los cambios.

Se puede decir que se dan 2 situaciones: o se ejecutan todas las instrucciones o no se ejecuta ninguna.

Y este es el objetivo principal de la estructura de una transacción, se usan transacciones cuando se tienen instrucciones que son vitales para el programa. De ejecutarse solamente una parte de las instrucciones sería catastrófico para la aplicación.

13

Bases de Datos 2

14

Bases de Datos 2

Un ejemplo claro de esto es la situación con los bancos, si usted está haciendo una transferencia de fondos de una cuenta a otra y ocurre un fallo ocurren problemas. Usted podría terminar con menos dinero sin haber transferido el dinero o incluso terminar con la misma cantidad de dinero con la transferencia ya hecha. De ejecutarse solamente una parte de la transacción se genera un desbalance de saldos.

Este es un modelo básico de la estructura del funcionamiento de las transacciones.

```

begin work
-----
) Instrucciones
) de la
) transacción
-----
commit work o rollback work
    
```

Figura No. 1 – Transacción

Begin work: Con esta instrucción se especifica que está comenzando una transacción.

Commit work: La instrucción commit nos sirve para guardar las instrucciones que teníamos en memoria luego de una transacción. Una vez que se llega a un commit ya no es posible regresar a los datos previos a la transacción.

Rollback work: Mediante esta instrucción estamos indicando que hubo un error y debemos volver la base de datos al estado en el que estaba antes de comenzar la transacción. Existen algunas instrucciones como la creación de tablas que tienen un commit implícito, pero la mayoría de operaciones básicas soportan un rollback.

La imagen anterior nos muestra el modelo de transacción para el estándar NO ANSI, en el cual se comienza cada transacción con Begin Work y se termina ya sea con Commit Work o Rollback Work. El modelo más utilizado es el estándar ANSI, en el cual el Begin Work está implícito y por lo tanto no es necesario usarlo. Oracle usa el estándar ANSI mientras que SQL Server utiliza el estándar no ANSI.

1.1.2 ACID

Existe una definición más exacta en base a las características de las instrucciones para poder ser consideradas como una transacción. Están basadas en el concepto de las siglas ACID, este acrónimo está basado en 4 propiedades básicas las cuales son: Atomicity, Consistency, Isolation y Durability (Atomicidad, consistencia, aislamiento y durabilidad respectivamente) por sus siglas en inglés ACID.

Por lo tanto se puede decir que para que una transacción se considere como tal debe cumplir con estas cuatro características, vamos a explicar cada una con más detalle:

15

Bases de Datos 2

16

Bases de Datos 2

Figura 8 – Páginas 17-20 Libro “Bases de Datos 2”

Atomicidad: esta propiedad considera a la transacción en sí como una instrucción gigante a pesar de estar compuesta por varias instrucciones. Por lo tanto se podrá o no ejecutar la transacción, nunca una de sus partes únicamente.

Consistencia: esta propiedad asegura que la base de datos estará en un estado consistente al terminar la transacción, en estados intermedios no debe estar necesariamente en un estado consistente. Por lo tanto se ejecutan aquellas operaciones que van acorde a las reglas y directrices de integridad de la base de datos.

Aislamiento: Cada instrucción de la transacción no debe afectar a las otras instrucciones en la base de datos, se tiene un aislamiento a nivel de cada instrucción para evitar errores de acceso sobre los mismos datos.

Durabilidad: Se asegura que una vez se haya considerado exitosa la transacción los cambios realizados sobre la base de datos sean persistentes.

1.1.3 Estado de la base de datos

Algo importante que debemos saber sobre el estado de la base de datos concierne a transacciones son los diferentes modos sobre los cuales se puede trabajar. La mayoría de DBMS soportan estos modos por lo que es importante saberlos.

Modo Autocommit: es el modo default sobre el cual esta SQL Server. Bajo un modo auto commit no es necesario usar instrucciones como commit o rollback. Cada instrucción es

considerada como una transacción y por lo tanto luego de cada instrucción se ejecuta automáticamente un commit sobre la base de datos, guardando así automáticamente todo cambio.

Modo explícito: Comienza cuando utilizamos la instrucción begin work para SQL y termina al usar un commit o un rollback. Por lo tanto cuando estamos utilizando esta estructura estamos utilizando el modo explícito para transacciones, donde nosotros especificamos cada parte de la transacción de manera manual.

Modo implícito: Bajo este modo no es necesario utilizar begin work para comenzar una transacción, en realidad la base de datos bajo este modo considera todas las instrucciones ingresadas como una transacción. Se podría decir que es lo opuesto al modo auto commit. En este modo las instrucciones ingresadas no son guardadas permanentemente hasta que nosotros ingresemos un commit, en este momento comienza una nueva transacción a espera de otro commit o un rollback.

1.1.4 Puntos de sincronización

Sabemos que luego de una transacción ejecutada exitosamente se termina con un commit, esto establece lo que se llama punto de commit o también llamado punto de sincronización. Un punto de sincronización se define como un punto en el cual se ha ejecutado exitosamente un conjunto de instrucciones y por lo tanto la base de datos esta en un estado consistente.



Ocurren dos cosas cuando nosotros llegamos a un punto de sincronización. Todas las instrucciones que se han ejecutado antes de un punto de sincronización se consideran cambios tentativos ya que pueden ser deshechos en cualquier momento por medio de un rollback, una vez que se realiza la sincronización todos los cambios se hacen permanentes y ya no pueden ser deshechos. Luego de la sincronización o luego de un rollback se pierde todo posicionamiento temporal sobre tuplas y se deshabilitan todas las llaves sobre las tuplas también permitiendo nuevos cambios sobre estas.

1.1.5 Bitácora

Para comprender como es posible realizar un rollback y regresar a un estado anterior de la base de datos, debemos explicar brevemente lo que es la bitácora. Una bitácora está formada por uno o más archivos que guardan todos los cambios hechos sobre la base de datos mientras se está ejecutando. La manera en que se guarda es siempre al final del archivo haciendo de este un proceso rápido.

Guardando aquí todas las acciones ejecutadas sobre la base de datos es posible volver a un estado anterior en la misma, y de esta manera preservar la consistencia.

1.2 Concurrencia en Bases de Datos

1.2.1 Problemas de concurrencia

Vamos a ver los problemas que ocurren al ejecutarse exitosamente dos transacciones en paralelo. Aquí la concurrencia se da sobre los datos mismos, acciones ocasionadas por dos diferentes transacciones. Vamos a ver que surge un problema de consistencia con los datos, pero paradójicamente las transacciones ejecutadas no están violando ninguna regla. Existen tres casos en los que tenemos un problema de concurrencia, estos son:

1.2.1.1 El problema del update perdido

Transacción A	Tiempo	Transacción B
	t1	
Obtener datos Tabla X	t2	Obtener datos Tabla X
	t3	
Update Tabla X	t4	Update Tabla X

Figura No. 2 – Update Perdido

Con esta imagen podemos explicar el problema del update perdido, se tienen las transacciones A y B. Estas dos transacciones están trabajando con la misma tabla llamada X, lo que hacen es obtener ciertos datos de la tabla, realizar algunos cambios y luego actualizar la tabla.



Figura 9 – Páginas 21-24 Libro “Bases de Datos 2”

Cada paso se realiza en una unidad de tiempo en común, en el primer paso la transacción A ha obtenido datos de la tabla X, en el segundo paso la transacción B hace lo mismo. En el tercer paso la transacción A actualiza los datos cambiados en la tabla X pero aquí es donde surge el problema, como no se ha hecho un commit sobre la transacción A (porque todavía no ha terminado la transacción) los datos no han sido guardados.

Antes de poder realizar un commit sobre la transacción A, la transacción B actualiza sus cambios sobre la misma tabla X, en este paso se han perdido los cambios realizados sobre la tabla X por la transacción A.

1.2.1.2 El problema de dependencia no cometida (uncommitted)

Transacción A	Tiempo	Transacción B
	t1	
Obtener datos Tabla X	t2	Update Tabla X
	t3	
	t4	Rollback

Figura No. 3 – Dependencia no cometida

Este problema se da cuando una transacción obtiene datos no actualizados, datos que han sido cambiados por otra transacción pero en donde dichos cambios no han sido cometidos, por lo que existe la probabilidad que los cambios sean regresados por medio de un rollback.

Vemos este ejemplo, en el tiempo 1 la transacción B ha realizado un update sobre la tabla X, en el tiempo 2 la transacción A ha obtenido datos de la tabla X para trabajar con ellos, en el tiempo 3 la transacción B realiza un rollback de la información.

¿Qué es lo que sucede aquí?, si la transacción B hace un rollback sobre los datos el update realizado en el tiempo 1, este update realmente nunca sucedió, por lo tanto la transacción A esta trabajando con datos que no existen creando serios problemas de consistencia.

1.2.1.3 El problema de análisis inconsistente

Transacción A	Tiempo	Transacción B
Leer tupla 1, suma = 30	t1	
Leer tupla 2, suma = 50	t2	
	t3	Leer tupla 3
	t4	Update tupla 3, 40 -> 25
	t5	Leer tupla 1
	t6	Update tupla 1, 30 -> 45
	t7	COMMIT
Leer tupla 3, suma = 105	t8	

Figura No. 4 – Análisis inconsistente

Tupla 1= 30, tupla 2=50, tupla 3=40. Suma = 120

Para ejemplificar este caso tenemos un ejemplo más complicado. Tenemos dos transacciones que trabajan sobre las

cuentas de un banco, para este ejemplo solo vamos a tener 3 cuentas en la tabla. La transacción A esta haciendo una suma de saldos y la transacción B está realizando una transferencia de fondos, esta debitando 15 de la cuenta 3 y le esta sumando 15 a la cuenta 1.

El proceso comienza con los tiempos 1 y 2 con la transacción A haciendo sumas de las cuentas 1 y 2 de la tabla, por ahora la suma es 80 hasta ahora todo va bien. En el tiempo 3 la transacción B comienza a realizar la transferencia de fondos, sustrae 15 de la cuenta 3 y agrega 15 a la cuenta 1, luego de esto se realiza el commit.

En el tiempo 8 la transacción A sigue con la suma de cuentas y ahora toma el valor de la cuenta 3 y lo suma, teniendo como resultado 105. Esta suma es incorrecta.

Se dice entonces que la transacción A ha leído información inconsistente de la base de datos y por lo tanto ha realizado un análisis inconsistente.

1.2.2 Candados:

Hemos visto los problemas que pueden surgir por problemas de concurrencia, estos problemas se pueden solucionar utilizando candados. Como lo sugiere la palabra, esta técnica consiste en permitirle el uso exclusivo a una transacción sobre un objeto en particular por medio de la negación de acceso a otras transacciones.

Existen dos tipos de candados. Los candados exclusivos (X) y los candados compartidos (S), ahora vamos a explicar la naturaleza de cada tipo de candado.

Si hemos puesto un candado tipo X sobre una tupla y otra transacción quiere poner su propio candado (no importa si es de tipo X o S), esto no será permitido y tendrá que esperar hasta que el candado exclusivo sea liberado.

Si hemos puesto un candado tipo S sobre una tupla y otra transacción quiere poner un candado tipo X, esto no será permitido. Pero si el tipo de candado que la otra transacción quiere poner también es de tipo S, entonces le será permitido poner su propio candado.

Por esta razón a este tipo de candados también se les conoce como candados de escritura (X) y candados de lectura (S).

Existe un protocolo en cuanto a la utilización de candados, en realidad siempre que se quiere realizar una acción sobre una tupla en la base de datos se debe de crear un candado sobre este, estas son las 4 reglas:

Una transacción que desea obtener datos de una tupla debe de imponer un candado S.

Una transacción que desea realizar un update sobre una tupla debe de imponer un candado X. Si la transacción ya tiene puesto un candado tipo S sobre esta tupla, debe de eliminar el candado S y reemplazarlo con un candado X.

Figura 10 – Páginas 25-28 Libro “Bases de Datos 2”

Si la transacción A desea poner un candado sobre una tupla pero esta no puede por conflicto de permisos por un candado impuesto por la transacción B, la transacción A debe esperar hasta que la transacción B libere este candado.

Los candados tipo X siempre son liberados hasta que se termina una transacción. Los candados tipo S usualmente son liberados al terminar la transacción pero puede haber excepciones.

1.2.3 Deadlocks:

Ahora que sabemos de la existencia de los candados vamos a revisar el caso del problema del update perdido aplicando este nuevo concepto.

Transacción A	Tiempo	Transacción B
Obtener datos Tabla A (candado S)	t1	
	t2	Obtener datos Tabla A (candado S)
Update Tabla A (candado X)	t3	
esperando...	t4	Update Tabla A (candado X)
esperando...	t5	esperando...
esperando...	t6	esperando...
esperando...	t7	esperando...
esperando...	t8	esperando...

Figura No. 5 – Deadlock update perdido

Tenemos el mismo caso para el update perdido, pero aquí ya estamos aplicando candados. En los pasos 1 y 2 las transacciones A y B respectivamente piden candados S para leer los datos. En el

tiempo 3 la transacción A intenta poner un candado X pero debe esperar hasta que la transacción B quite su candado S. En el tiempo 4 la transacción B intenta poner un candado X pero debe esperar hasta que la transacción A termine de usar su candado. Aquí en este caso ambas transacciones se quedan esperando indefinidamente.

Este problema que se da en este caso es llamado un Deadlock. Un deadlock ocurre cuando dos transacciones se encuentran en un estado de espera, esperando la liberación de un candado que en este caso no puede ocurrir. Es posible también que se den deadlocks involucrando 3 o incluso más transacciones, todas esperando la liberación de un candado que otra transacción no libera esperando la liberación de otro candado. Pero en general el caso más común es el deadlock involucrando dos transacciones.

Existen en general dos maneras de lidiar con este problema. La primera involucra simplemente escoger una de las transacciones que está causando el problema, hacer un rollback y esperar que el deadlock se rompa. Otro enfoque es por medio de tiempos de espera, se asume que después de cierto tiempo límite esperando una transacción se encuentra en un deadlock y por lo tanto debe hacerse un rollback.

Ahora desde el punto de vista del sistema este problema se puede a su vez tratar de dos maneras. La primera es volver a ejecutar la transacción que tuvo que ser terminada, asumiendo que el deadlock causado ya no se volverá a dar. La otra es simplemente



25

Bases de Datos 2



26

Bases de Datos 2

mandar una excepción al sistema y dejar que el programador lidie con esto.

Artículos relacionados

- **Controlling Database Transactions in ColdFusion MX**
Simon Horwith
Ref: <http://www.adobe.com/devnet/coldfusion/articles/cftransaction.html>
- **Reducing SQL Server locks**
Brad McGehee
Ref: http://www.sql-server-performance.com/tips/reducing_locks_p1.aspx
- **Managing transactions with Hibernate**
A.P. Raisinghkar
Ref: <http://www.devarticles.com/c/a/Java/Managing-Transactions-with-Hibernate/>
- **Pitfalls of transactions with PHP**
Kimberlee Jensen
Ref: http://onlamp.com/pub/a/php/2003/12/18/transaction_pitfalls.html



27

Bases de Datos 2

Capítulo 2

BACKUP Y RECUPERACIÓN

Este capítulo mostramos las diferentes técnicas para tener copias de seguridad de la base de datos y la manera de recuperar estas copias. Explicamos los elementos involucrados en este proceso y presentamos el concepto de la alta disponibilidad.

ALCANCE



28

Bases de Datos 2

Figura 11 – Páginas 29-32 Libro “Bases de Datos 2”

- 2. Backup y recuperación
 - 2.1. Respaldo y recuperación
 - 2.1.1. Clasificación de fallos
 - 2.1.1.1 Fallos de Hardware
 - 2.1.1.2 Fallos de Software
 - 2.1.1.3 Fallos de usuario
 - 2.1.2. Tipos de backup
 - 2.1.2.1 Archivos de sistema
 - 2.1.2.2 Backup de datos
 - 2.1.2.3 Naturaleza del backup
 - 2.1.3. Recuperación basada en bitácora
 - 2.1.3.1 Bitácora
 - 2.1.3.2 Recuperación
 - 2.2. Alta disponibilidad
 - 2.2.1. Diferentes capas de almacenamiento y manejo de datos
 - 2.2.2. Servidores en standby
 - 2.2.2.1 Standby lógico
 - 2.2.2.2 Standby físico
 - 2.2.3. Servidores bajo cluster

2.1.1 Clasificación de fallos

Existen muchas formas en las cuáles puede fallar nuestra base de datos, en general tenemos tres grandes clasificaciones:

2.1.1.1 Fallos de Hardware

Este tipo de fallos se dan por daño físico en el equipo de cómputo. Como ejemplo de esto tenemos que se dañe el disco duro, que falle la tarjeta de red o la tarjeta de video, fallos en la memoria RAM o en general fallos en la infraestructura de la red. Todos estos fallos son causados por el malfuncionamiento de un componente físico de la máquina.

2.1.1.2 Fallos de Software

Este tipo de fallos se da en el software que utilizamos en nuestro servidor. Bajo esta clasificación caen fallos como daños causados por bugs del software o malfuncionamiento de este por fallos en la programación del mismo. Fallos causados por la infección de la máquina por algún virus y también fallos de seguridad causados por el software.

2.1.1.3 Fallos de Usuario

2.1 Respaldo y Recuperación

29

Bases de Datos 2

30

Bases de Datos 2

Este tipo de fallos puede ser clasificado bajo errores humanos, en este tipo de errores se puede incluir pérdida de datos por ejecutar por error algún query de eliminación. Pérdida de objetos y también pérdida de archivos se clasifican bajo este error. Por ejemplo la eliminación accidental de archivos del sistema o incluso archivos donde se había realizado un backup.

Podemos ver que existen muchas maneras en las cuáles puede resultar un fallo en la base de datos, en general lo que se hace aquí es un análisis de costo de backup contra costo de pérdida.

Cuando hablamos de costos de backup hablamos de la cantidad de dinero que se debe invertir en realizar backups periódicos sobre la base de datos. Los costos de pérdida son el dinero que se pierde por perder cierta cantidad de información. Los costos de pérdida varían enormemente entre sistemas.

Puede que en algunos casos la información que se pierde no sea tan importante y que el costo de realizar backups periódicamente se considere innecesario, optando de esta manera por realizar backups una vez al mes de manera manual. Se puede tener también el caso en el cuál la información sea extremadamente valiosa y no se corten gastos a la hora de tener un sistema de backup listo.

Existen maneras para bajar considerablemente la probabilidad de tener un fallo para cada una de las tres clasificaciones de fallos. Para fallos de hardware se tiene la opción de crear un sistema en

stand by o bajo cluster en discos redundantes. Para fallos de software se tiene la opción de tener sistemas automáticos de backup. Para fallos de usuario se pueden reducir al mínimo creando un esquema estricto de seguridad para los usuarios del sistema.

En todos estos casos siempre se realiza un análisis de costos de pérdida contra costos de backup.

Hemos estado utilizando la palabra backup en varias ocasiones sin haber dado una definición formal. Un backup es un proceso por medio del cual se generan archivos en base a la información contenida en la base de datos que estamos utilizando. Estos archivos pueden ser utilizados luego para volver a cargar la información que se tenía en cierto momento específico en el tiempo. Y este es el objetivo principal, lograr recuperar después la base de datos luego de que haya ocurrido algún tipo de fallo sobre esta, lograr reconstruirla por completo de manera exitosa.

2.1.2 Tipos de backup

2.1.2.1 Archivos de sistema

Cuando hablamos de Oracle existen tres grandes clasificaciones para los archivos de sistema. Tenemos los archivos de datos, los archivos de sistema y los archivos de la bitácora. Pero al utilizar SQL Server se observa que la mayoría de la información de la base de datos se encuentra en tablas del sistema. Es recomendable al

31

Bases de Datos 2

32

Bases de Datos 2

Figura 12 – Páginas 33-36 Libro “Bases de Datos 2”

realizar un backup tomar en cuenta estos tres grupos y también saber la manera en que funciona el DBMS que hemos escogido.

Backup de Datos: en un backup de datos se generan archivos que tienen el contenido de las tablas de nuestra base de datos. Esto se realiza por medio de comandos como Export/Import. La desventaja de este proceso es que es más lento que otros backups.

Backup de archivos: en un backup de archivo se realizan copias directamente de los archivos que utilizan el DBMS para poder funcionar. Esto no es nada más que un copy paste de archivos de sistema.

Backup de Bitácora: se realiza una copia de los archivos utilizados por la bitácora. Este tipo de backup se puede configurar para que trabaje de manera automática.

2.1.2.2 Backup de datos

Backup en frío: en un backup en frío de datos se debe detener la base de datos como primer paso. Luego de esto se ejecutan las instrucciones para realizar el backup. Como paso final se vuelve a iniciar la base de datos. Uno de los grandes inconvenientes de este tipo de backup es tener que bajar la base de datos durante el backup, cosa que en algunos casos resulta muy costoso. La ventaja es que este backup es más rápido que su contraparte.

Backup en caliente: en un backup en caliente de datos no se tiene que detener la base de datos para realizar el backup, esto se logra por medio de la bitácora. El proceso comienza indicando a la

base de datos el inicio del backup, en este momento todos los cambios realizados a la base de datos no se guardan en el disco duro, se guardan en la bitácora hasta que el backup termine. Una vez terminado el backup se ejecutan todas las instrucciones dejadas temporalmente en la bitácora. La ventaja es obvia al no tener que detener la base de datos. La desventaja está en que el proceso se alarga ya que requiere de más recursos de la máquina al estar realizando un backup y al mismo tiempo ejecutando la base de datos.

2.1.2.3 Naturaleza del backup

Backup completo: en este tipo de backup se toman todos los archivos, es decir todos los datos que contiene la base de datos. Este tipo de backup se hace normalmente cuando se hace el primer backup. Por ser el primer backup que se realiza es el que ocupa más espacio y es el que más tiempo se tarda, pero se tiene la ventaja que no se necesita de otro backup para realizar la recuperación.

Backup incremental: en este tipo de backup únicamente estamos guardando los datos que han cambiado desde el último backup, esto funciona por medio de comparación de fechas de modificación a los datos. Por ejemplo si durante el último backup se tenían 100 registros y en este momento se tienen 110 registros, vamos a realizar un backup únicamente sobre los 10 registros nuevos. Cuando se realiza otro backup este se compara contra el último incremental realizado. La ventaja de este sistema es que no se tarda mucho tiempo en realizar el backup porque solo actualiza

33

Bases de Datos 2

34

Bases de Datos 2

los nuevos cambios, el problema se da a la hora de realizar la recuperación. Si se quiere recuperar por ejemplo la base de datos completa a partir de cero, se necesitan todos los backups incrementales que hemos hecho.

Backup diferencial: tiene la misma mecánica que el backup incremental, las únicas diferencias son que compara el contenido de los archivos y no únicamente la fecha de modificación para saber que archivos se han cambiado. Su mayor diferencia está en que los backups se comparan contra un backup completo no contra otro backup parcial. Se puede considerar al backup diferencial como una manera de nivelar el tiempo de backup y recuperación tan desproporcionado de los backups incrementales. Esto se da porque cada backup diferencial es comparado contra un backup completo, de manera que la recuperación aquí es más rápida.

2.1.3 Recuperación basada en bitácora

2.1.3.1 Bitácora

En la sección anterior se hizo una breve descripción de lo que es la bitácora, es básicamente un diario de las acciones ejecutadas sobre la base de datos, de todas las transacciones y a que objetos afectan.

Pero para tener la habilidad de regresar la base de datos a un estado anterior necesitamos más información que esto. Necesitamos también tener referencias a la información que ha

sido cambiada y es aquí donde se pierde la mayor cantidad de espacio guardando los logs. Teniendo toda esta información es posible recuperar los cambios realizados en la base de datos a partir de la bitácora.

Es importante saber también que la bitácora trabaja de manera circular. Físicamente la bitácora existe en archivos binarios. Por ejemplo en Oracle se debe tener como mínimo dos archivos para poder manejar la bitácora, vamos a explicar este proceso. Cuando se comienza a escribir en la bitácora se utiliza el primer archivo disponible, una vez lleno este archivo se va al siguiente. Cuando todos los archivos están llenos se regresa al archivo inicial y se comienza el proceso de nuevo.

Aquí existe una pequeña variación dependiendo del modo en el que está el archive log. Si tenemos activado el modo archive log, cuando el DBMS tenga que utilizar un archivo que ya ha sido utilizado antes, primero va a realizar una copia de este archivo en otro lugar que nosotros especificamos. De esta manera estaremos guardando todos los archivos de la bitácora. Pero si el modo archive log no está activado, una vez que se encuentre un archivo usado anteriormente el DBMS simplemente lo va a borrar y va a trabajar encima de este.

35

Bases de Datos 2

36

Bases de Datos 2

Figura 13 – Páginas 37-40 Libro “Bases de Datos 2”

2.1.3.2 Recuperación

Cuando ocurre una falla en el sistema por medio de la bitácora es posible re hacer instrucciones que no fueron guardadas en el disco duro por alguna razón, de la misma manera es posible también deshacer cambios realizados sobre esta. Por medio de la bitácora es posible realizar recuperaciones en un momento específico en el tiempo, ya que teniendo todas las acciones ejecutadas podemos regresar a un estado previo de la base de datos.

Hemos visto el papel crucial que tiene la bitácora a la hora de realizar un backup en caliente, si no tenemos activado el modo archive log no podemos realizar un backup sin bajar la base de datos. De la misma manera si el modo archive log no está activado no tenemos la opción de hacer una recuperación de la base de datos en un momento en el tiempo específico.

Por lo tanto la bitácora juega un papel crucial tanto en las operaciones cotidianas de la base de datos como en operaciones de backup y recuperación.

2.2 Alta disponibilidad

Un sistema de alta disponibilidad es un sistema ya sea por hardware y/o por software que tiene como objetivo asegurar la continuidad operacional. Se utiliza en sistemas que necesitan estar disponibles 24 horas al día 365 días al año.

La efectividad de un sistema de alta disponibilidad se mide en el porcentaje de operación y el porcentaje de tiempo que este provee. Aquí tenemos una tabla en la cual se detalla a partir de cierto porcentaje de operación cuáles son sus porcentajes de caída, tiempo de caída por año y el tiempo de caída por semana.

% tiempo de operación	% tiempo de caída	tiempo de caída X año	tiempo de caída X semana
98%	2%	7.3 días	3 h. 22 min.
99%	1%	3.65 días	1 h. 41 min.
99.8%	0.2%	17 h. 30 min.	20 min. 10 seg.
99.9%	0.1%	8 h. 45 min.	10 min. 5 seg.
99.99%	0.01%	52.5 min.	1 min.
99.999%	0.001%	5.25 min.	6 seg.
99.9999%	0.0001%	31.5 seg.	0.6 seg.

Figura No. 6 – Porcentajes de disponibilidad

Podemos ver que para un sistema de alta disponibilidad al 98% su tiempo de caída por semana es de 3 horas y 22 minutos y para un sistema al 99.9999% el tiempo de caída por semana es de 0.6 segundos.

Existen varias formas de lograr alcanzar la alta disponibilidad pero todas se basan en un principio muy sencillo, la replicación. Por medio de la replicación de datos y de hardware podemos aumentar el porcentaje de tiempo de operación de nuestro sistema.



Recordemos también los tipos de fallos que existen, dividiendo en detalle las fuentes principales de problemas tenemos la siguiente estadística sobre el origen de las fallas:

- 40 % causadas por Software
- 30 % es planeado
- 15 % causado por el usuario
- 10 % causado por hardware
- 5 % causado por el ambiente

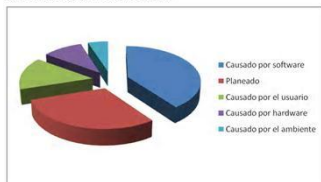


Figura No. 7 – Causas de fallos

Cuando hablamos de fallas planeadas se habla de reinicio del sistema por actualizaciones de algún componente de software o revisión de hardware.

Estas estadísticas nos recuerdan que es importante tener un sistema de alta disponibilidad pero que las fallas de un sistema dependen de una gran variedad de factores.

Disponibilidad: al hablar de alta disponibilidad este concepto es importante. La disponibilidad es una fórmula por medio de la cual se puede medir el tiempo en el que el sistema está funcionando correctamente.

D= grado de disponibilidad (porcentaje)

TMEF= tiempo medio entre fallas

TMRF= tiempo máximo para resolver o reparar el problema

En base a estas variables es posible calcular exactamente la disponibilidad de un sistema con la siguiente fórmula:

$$D = \text{TMEF} / (\text{TMEF} + \text{TMRF})$$

2.2.1 Diferentes capas de almacenamiento y manejo de datos

Vamos a ver los diferentes niveles definidos para el manejo y almacenamiento de datos, aquí se utiliza la replicación para lograr un sistema de alta disponibilidad. Por medio de estos métodos estamos protegiendo la integridad de los datos, esta es una de las partes más importantes a considerar en un sistema de alta disponibilidad.

Discos Duros físicos: son los discos duros donde se guarda la información.



Figura 14 – Páginas 41-44 Libro “Bases de Datos 2”

SAN: o storage área networks son una arquitectura por medio de la cual se pueden conectar varios dispositivos de almacenamiento remotos. Estos se configuran de tal manera que para fines funcionales se trata como si fuera un solo dispositivo de almacenamiento. El sistema operativo no ve ninguna diferencia entre una SAN y un disco duro sencillo. Esta arquitectura trabaja a nivel de bloque. Comúnmente estos dispositivos se conectan por medio de un canal de fibra.

NAS: o network attached storage es básicamente un servidor de archivos. Por medio de una NAS podemos compartir varios dispositivos de almacenamiento con varios servidores por medio de una red LAN o WAN. Esta arquitectura trabaja a nivel de archivos. Una NAS es particularmente útil para compartir información entre diferentes plataformas, ya que aquí la NAS misma se encarga de manejar el sistema de archivos.

Para ambas configuraciones estamos logrando el mismo objetivo, compartir un conjunto de dispositivos de almacenamiento a varias computadoras.

RAID por hardware: es un acrónimo para Redundant Array of Independent Disks, es un sistema de replicación que toma un conjunto de discos duros y replica o distribuye los datos dentro de estos. Existen varios niveles para un RAID, por ejemplo el nivel 0 de RAID se encarga únicamente de distribuir la información en varios discos para que aparente ser un disco duro gigante. Los niveles posteriores para un RAID tienen como objetivo la replicación de los datos en diferentes formas. Esto tiene como fin aumentar la

tolerancia a los fallos. Si uno de los discos duros falla automáticamente se obtienen la información en otro de los discos duros, ya que la información está replicada.

RAID por software : existe también la opción de utilizar una configuración RAID por software, se tiene la opción de crear un RAID de nivel 5 por ejemplo en un mismo disco duro utilizando discos dinámicos dentro de este. El problema de esta configuración es que se pierde el nivel extra de seguridad al tener discos duros independientes.

Sistema de archivos o dispositivos crudos (raw devices): debemos saber que no todas las áreas de almacenamiento deben estar formateadas con un sistema de archivos. Existen aplicaciones que no necesitan áreas de almacenamiento formateadas y ven estas áreas como una línea continua de bits. Las ventajas de los dispositivos crudos es que su acceso a los datos es mucho más rápido.

Aplicación o Base de Datos: la base de datos o la aplicación en sí finalmente va a escribir la información en el dispositivo o en la configuración que hemos escogido.

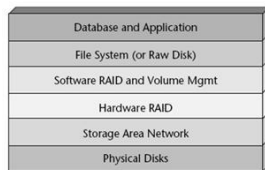


Figura No. 8 – 6 capas de almacenamiento y manejo de datos
Fuente: Blueprints for High Availability por Marcus y Stern.

Estos son los diferentes niveles a considerar para el almacenamiento de los datos, es muy importante notar que cada nivel es independiente del otro. Si decidimos utilizar una SAN para guardar la información no debe influir en nuestra decisión de la posibilidad de implementación de un RAID por ejemplo.

2.2.2 Servidores en Standby

El concepto de la replicación bajo standby se basa en tener una copia en hardware del servidor principal. Se tiene una computadora que puede funcionar por sí sola como servidor, funcionando junto al servidor principal que vamos a llamar servidor secundario. Si durante el funcionamiento del servidor principal este comienza a

tener problemas, se desactiva como servidor principal y se activa el servidor secundario. De esta manera el tiempo que el sistema está caído es el tiempo en que se tarda en hacer el intercambio de servidores.

El servidor principal funciona en todo momento como la base de datos, realizando todas las tareas de actualización necesarias sobre la base de datos misma. El servidor principal luego de realizar cualquier operación manda esta misma instrucción al servidor secundario para que este también esté actualizado. Ya que el servidor secundario es una copia al carbón del servidor principal es comúnmente utilizado para realizar consultas sobre los datos y reportes. De esta manera se le da un uso al servidor secundario y se reduce la carga sobre el servidor principal.

Existen dos maneras en las cuales se mantiene al servidor secundario actualizado.

2.2.2.1 Standby Lógico

En una base de datos standby lógica se utilizan la bitácora para actualizar al servidor secundario. El servidor toma la bitácora, transforma cada registro en instrucciones SQL y luego las ejecuta sobre el servidor secundario. Como servidor en standby la base de datos debe estar en modo de solo lectura para mantener la consistencia de la base de datos. Las tablas en la base de datos del servidor secundario pueden tener índices diferentes y otras



Figura 15 – Páginas 45-48 Libro “Bases de Datos 2”

características físicas diferentes, pero siempre se debe mantener una consistencia lógica con la base de datos principal.

2.2.2.2 Standby Físico

En este tipo de standby la base de datos del servidor secundario debe ser exactamente igual físicamente a la base de datos principal. La estructura misma de la base de datos secundaria debe ser idéntica a nivel de bloques, porque el proceso de actualización aplica los cambios realizados en el servidor principal afectando directamente los bloques de datos y utiliza el rowid físico. El schema de la base de datos e incluso los índices deben ser los mismos.

2.2.3 Servidores bajo cluster

En un sistema bajo cluster se tiene un conjunto de servidores llamados nodos, usualmente 2 pero se puede llegar a tener n servidores. Estos servidores utilizan un sistema de balanceo de carga y también se caracterizan por tener discos compartidos por medio de una SAN o una NAS.

La importancia de un sistema bajo cluster es el sistema de alta disponibilidad que provee. Si durante el funcionamiento del servidor principal ocurre una falla ya sea de hardware o de software, el sistema lo detecta automáticamente y reinicia la aplicación en uno de los nodos. Si ocurre un error en alguno de los

nodos se repite este proceso activando el siguiente nodo. Este proceso automático es llamado Failover.

Cuando uno de los nodos o el servidor principal falla se debe realizar una recuperación en estas máquinas. Una vez que la máquina funciona nuevamente se devuelve el control a esta reiniciando los servicios aquí nuevamente. Este proceso es llamado Failback. De esta manera se asegura una estructura de alta disponibilidad.

Por la manera en que esta creado el cluster provee la opción de balanceo de carga, algo que un servidor en standby no provee por completo. Se tiene también la seguridad extra de poder agregar n número de nodos al cluster.

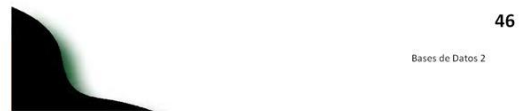
Artículos Relacionados

- **What are digital vaults?**
Calum Macleod
Ref: <http://www.storagesearch.com/cyberark-art2.html>



45

Bases de Datos 2



46

Bases de Datos 2

- **Surviving non-traditional data disasters**
Ian Masters
Ref: <http://www.storagesearch.com/sunbelt-art1.html>
- **Do CDs and DVDs Have a Long Term Future as Digital Storage?**
Zsolt Kerekes
Ref: <http://www.storagesearch.com/view56.html>
- **Offsite Backup Exposes Potential Threats of Tape Backup**
Lee Morrell
Ref: <http://www.articlesbase.com/data-recovery-articles/offsite-backup-exposes-potential-threats-of-tape-backup-33495.html>

Capítulo 3

SEGURIDAD

En este capítulo se desarrolla el tema de seguridad, no solamente se ve el tema de base de datos vemos la seguridad a nivel de usuario y sistema. Se revisan conceptos básicos de seguridad y entramos al final en detalle el tema de seguridad de base de datos.



47

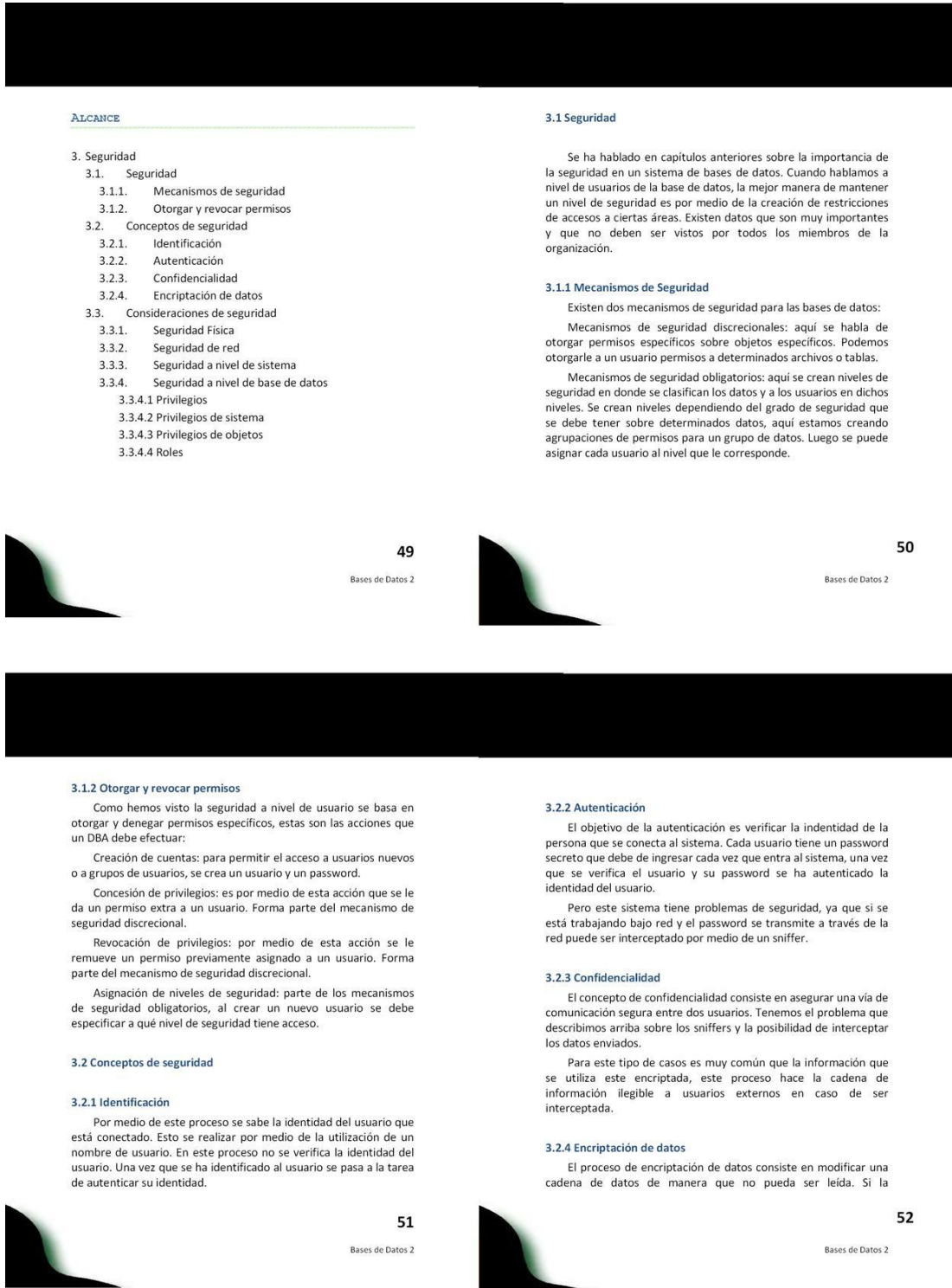
Bases de Datos 2



48

Bases de Datos 2

Figura 16 – Páginas 49-52 Libro “Bases de Datos 2”



ALCANCE

3. Seguridad

- 3.1. Seguridad
 - 3.1.1. Mecanismos de seguridad
 - 3.1.2. Otorgar y revocar permisos
- 3.2. Conceptos de seguridad
 - 3.2.1. Identificación
 - 3.2.2. Autenticación
 - 3.2.3. Confidencialidad
 - 3.2.4. Encriptación de datos
- 3.3. Consideraciones de seguridad
 - 3.3.1. Seguridad Física
 - 3.3.2. Seguridad de red
 - 3.3.3. Seguridad a nivel de sistema
 - 3.3.4. Seguridad a nivel de base de datos
 - 3.3.4.1 Privilegios
 - 3.3.4.2 Privilegios de sistema
 - 3.3.4.3 Privilegios de objetos
 - 3.3.4.4 Roles

49

Bases de Datos 2

3.1 Seguridad

Se ha hablado en capítulos anteriores sobre la importancia de la seguridad en un sistema de bases de datos. Cuando hablamos a nivel de usuarios de la base de datos, la mejor manera de mantener un nivel de seguridad es por medio de la creación de restricciones de accesos a ciertas áreas. Existen datos que son muy importantes y que no deben ser vistos por todos los miembros de la organización.

3.1.1 Mecanismos de Seguridad

Existen dos mecanismos de seguridad para las bases de datos:

Mecanismos de seguridad discrecionales: aquí se habla de otorgar permisos específicos sobre objetos específicos. Podemos otorgarle a un usuario permisos a determinados archivos o tablas.

Mecanismos de seguridad obligatorios: aquí se crean niveles de seguridad en donde se clasifican los datos y a los usuarios en dichos niveles. Se crean niveles dependiendo del grado de seguridad que se debe tener sobre determinados datos, aquí estamos creando agrupaciones de permisos para un grupo de datos. Luego se puede asignar cada usuario al nivel que le corresponde.

50

Bases de Datos 2

3.1.2 Otorgar y revocar permisos

Como hemos visto la seguridad a nivel de usuario se basa en otorgar y denegar permisos específicos, estas son las acciones que un DBA debe efectuar:

Creación de cuentas: para permitir el acceso a usuarios nuevos o a grupos de usuarios, se crea un usuario y un password.

Concesión de privilegios: es por medio de esta acción que se le da un permiso extra a un usuario. Forma parte del mecanismo de seguridad discrecional.

Revocación de privilegios: por medio de esta acción se le remueve un permiso previamente asignado a un usuario. Forma parte del mecanismo de seguridad discrecional.

Asignación de niveles de seguridad: parte de los mecanismos de seguridad obligatorios, al crear un nuevo usuario se debe especificar a qué nivel de seguridad tiene acceso.

3.2 Conceptos de seguridad

3.2.1 Identificación

Por medio de este proceso se sabe la identidad del usuario que está conectado. Esto se realiza por medio de la utilización de un nombre de usuario. En este proceso no se verifica la identidad del usuario. Una vez que se ha identificado al usuario se pasa a la tarea de autenticar su identidad.

51

Bases de Datos 2

3.2.2 Autenticación

El objetivo de la autenticación es verificar la identidad de la persona que se conecta al sistema. Cada usuario tiene un password secreto que debe de ingresar cada vez que entra al sistema, una vez que se verifica el usuario y su password se ha autenticado la identidad del usuario.

Pero este sistema tiene problemas de seguridad, ya que si se está trabajando bajo red y el password se transmite a través de la red puede ser interceptado por medio de un sniffer.

3.2.3 Confidencialidad

El concepto de confidencialidad consiste en asegurar una vía de comunicación segura entre dos usuarios. Tenemos el problema que describimos arriba sobre los sniffers y la posibilidad de interceptar los datos enviados.

Para este tipo de casos es muy común que la información que se utiliza este encriptada, este proceso hace la cadena de información ilegible a usuarios externos en caso de ser interceptada.

3.2.4 Encriptación de datos

El proceso de encriptación de datos consiste en modificar una cadena de datos de manera que no pueda ser leída. Si la

52

Bases de Datos 2

Figura 17 – Páginas 53-56 Libro “Bases de Datos 2”

información debe ser usada se debe desencriptar. Existen varios métodos para encriptar datos, la mayoría está basada en el uso de llaves simétricas y asimétricas.

Cuando se está transportando información importante de un lugar a otro es crucial que está este encriptada, ya que si llega a ser interceptada no puede llegar a ser leída a menos que se tenga la llave para desencriptar. También es una buena práctica de seguridad encriptar la información importante aunque solo se utilice localmente.

Cuando se utiliza una llave simétrica se utiliza la misma llave para encriptar y desencriptar la información. Cuando se utiliza una llave asimétrica se tienen dos llaves diferentes, una llave pública y una llave privada. Una vez que la llave pública encripta algo solamente la llave privada puede desencriptarlo y viceversa.

3.3 Consideraciones de seguridad

Existen varios factores que debemos considerar cuando hablamos de la seguridad de la base de datos. Algo tan sencillo como dejar el password default para la creación de una cuenta nueva puede tener efectos desastrosos.

Es por esta razón que toda organización debe tener un conjunto de políticas de seguridad para minimizar la probabilidad de una falla en la seguridad.

Si tenemos un acceso a internet en nuestra base de datos debemos protegernos de virus y también de la posibilidad de ataques de acceso por parte de hackers.

De la misma forma tenemos que cuidarnos de ataques internos, es muy común ver ataques hacia una organización internamente por parte de un empleado descontento. De hecho estudios estadísticos nos dicen que un gran porcentaje de los ataques que sufre una organización son cometidos por miembros internos a la misma.

En general existen muchos aspectos que debemos considerar, comenzamos con la seguridad física.

3.3.1 Seguridad Física

Aquí estamos hablando de evitar que una persona no autorizada tenga acceso físico al servidor. Se deben tomar las precauciones necesarias para evitar que esto pase, algo tan sencillo como una cámara de seguridad en el lugar donde se guarda el servidor puede evitar que se llegue a causar un gran daño.

Para evitar violaciones de seguridad de este tipo es común utilizar también equipo biométrico o sistemas de seguridad por medio de tarjetas. También es común ver políticas de seguridad para las personas que están en contacto con el servidor, como por ejemplo registrar a los empleados antes de salir del edificio. Todas estas precauciones se toman en cuenta para la seguridad física del servidor.

53

Bases de Datos 2

54

Bases de Datos 2

3.3.2 Seguridad de red

La mayoría de sistemas deben estar conectados a internet para darle acceso a los clientes, en algunos casos solamente es una red interna para la organización. Cuando se tiene una red instalada y está en contacto con el servidor de la base de datos hay una serie de consideraciones que tenemos que tomar en cuenta.

Debemos configurar el router para que solamente los paquetes apropiados sean re direccionados desde internet a la red interna, aquí es donde vemos uno de los aspectos más importantes el firewall. Un firewall puede ser implementado ya sea por medio de software o por medio de hardware especializado para la tarea.

La labor del firewall se resume a la filtración de paquetes y a la administración de puertos. Existen también opciones como la utilización de un servidor proxy que por medio de una NAT protege a la red interna de internet. La utilización de VPN o redes virtuales privadas también es una opción viable para mantener la seguridad.

Debemos considerar los problemas de seguridad que aporta tener una red inalámbrica. Es muy importante limitar el acceso a personas externas a la organización ya que solamente se necesita una tarjeta de red inalámbrica y estar lo suficientemente cerca para poder conectarse.

3.3.3 Seguridad a nivel de sistema

El sistema operativo sobre el cual va a operar la base de datos también es una viabilidad de seguridad a considerar. Cuando

instalamos el sistema operativo debemos estar seguros de los servicios que son necesarios, no debemos instalar servicios de más y mucho menos hacer una instalación default del sistema operativo ya que puede generar problemas de seguridad.

De la misma manera debemos tener cuidado de los servicios que tiene el sistema operativo y únicamente habilitar los que se están utilizando. Debemos también revisar los parches de seguridad para el sistema operativo y mantenerlo actualizado.

La consideración más importante es el manejo correcto de los usuarios que tienen acceso al sistema y siempre reemplazar configuraciones default para la creación de cuentas.

3.3.4 Seguridad a nivel de base de datos

De igual manera que el sistema operativo se debe tener un método de autenticación por medio de usuario y password, solo que aquí se está hablando del entorno de la base de datos. También se debe poner especial atención a los permisos que se otorgan a los usuarios, vamos a ampliar este tema.

3.3.4.1 Privilegios:

Un privilegio se interpreta como el permiso de realizar determinada acción en la base de datos. Existe una gran variedad de permisos diferentes y estas se clasifican en dos grandes grupos.

55

Bases de Datos 2

56

Bases de Datos 2

Figura 18 – Páginas 57-60 Libro “Bases de Datos 2”

3.3.4.2 Privilegios de sistema:

Los privilegios de sistema son acciones que afectan a todo el sistema de base de datos, son permisos usualmente otorgados únicamente al DBA de la base de datos. Por ejemplo habilidades como crear sesiones, crear otros usuarios y crear tablas, estos permisos caen bajo esta clasificación.

3.3.4.3 Privilegios de objeto

Los privilegios sobre objetos son acciones que afectan a objetos de un schema específico. Cuando realizamos un esquema de seguridad y queremos asignar permisos específicos a cada uno de los usuarios del schema se utiliza este tipo de permisos. Los más comunes son permisos de eliminación, inserción y selección de tablas específicas.

3.3.4.4 Roles

Los roles los podemos visualizar como un grupo de permisos que encierran el cargo de una persona. Por medio de los roles podemos crear agrupaciones de permisos haciendo más fácil la labor de asignar permisos a nuevos usuarios. Por ejemplo un encargado del mantenimiento de la base de datos y un usuario tienen diferentes permisos, pero dentro de los roles encargado de mantenimiento de base de datos y usuario se manejan prácticamente los mismos permisos.

La manera en que funciona el sistema de roles es igual a como manejaríamos una nueva asignación de un permiso específico, si queremos asignar un rol a un usuario específico simplemente otorgamos por medio de un GRANT.

Cuando se habla de crear un esquema de seguridad estamos hablando de la creación de los diferentes roles que necesitamos para manejar de manera global la seguridad de la base de datos. Cada vez que se crea un nuevo usuario se le asigna un rol y si es necesario se asigna o elimina permisos específicos a cada usuario. Este esquema de seguridad es la mejor manera de llevar el control de la seguridad a nivel de base de datos, por lo tanto la labor de crear roles es la base de todo esquema de seguridad. Vamos a ver a continuación ejemplos de creación y asignación de roles.

```
CREATE ROLE rol_mantenimiento NOT IDENTIFIED;  
GRANT select TO role_mantenimiento;  
GRANT role_mantenimiento TO horaceh;
```

Podemos ver que el rol se maneja como si fuera un permiso más a la hora de asignarlo y cuando queremos darle privilegios se trata como si fuera un usuario.



57

Bases de Datos 2



58

Bases de Datos 2

Artículos Relacionados

- **Database security (Common sense principles)**
Blake Wiedman
Ref: http://www.governmentsecurity.org/Database_Security_Common-sense_Principles
- **The truth about database security**
Jon Oltsik
Ref: http://news.cnet.com/8301-10784_3-5834896-7.html
- **Network security is necessary to protect your information**
Jake Ruston
Ref: <http://ezinearticles.com/?Network-Security-is-Necessary-to-Protect-Your-Information&id=2503490>
- **Common misconceptions about database security**
Slavik Markovich
Ref: http://news.zdnet.com/2100-1009_22-200224.html



59

Bases de Datos 2



60

Bases de Datos 2

Capítulo 4

OPTIMIZACIÓN Y ALTO RENDIMIENTO

En este capítulo hablamos de la optimización en la base de datos. Vemos en detalle la manera de optimizar consultas y el proceso que conlleva. Luego vemos herramientas utilizadas para tuning de bases de datos, esto basado en las estadísticas de la misma.

Figura 19 – Páginas 61-64 Libro “Bases de Datos 2”



ALCANCE

- 4. Optimización y alto rendimiento
 - 4.1. Optimización de consultas
 - 4.1.1. El proceso de ejecución de consultas
 - 4.1.1.1 Parser
 - 4.1.1.2 Plan de ejecución
 - 4.1.1.3 Fetch
 - 4.1.2. Optimizador por regla
 - 4.1.2.1 Análisis de una instrucción
 - 4.1.2.2 Tabla Pivot
 - 4.1.2.3 Hints
 - 4.1.3. Optimizador por costo

4.1 Optimización de consultas

4.1.1 El proceso de ejecución de consultas

En esta sección vemos el proceso detrás de optimizar la ejecución de consultas, esta tarea está a cargo del Optimizador. El Optimizador entra en acción desde que nosotros ejecutamos una instrucción SQL, existen tres fases principales con las cuales se trabaja estas son:

4.1.1.1 Parser

En la etapa de parser verifica la escritura sintáctica y semántica de las instrucciones SQL ejecutadas. La primera acción que se realiza es verificar como una cadena completa la instrucción SQL y luego verificarla con la cache de la base de datos. El objetivo de esta operación es evitar realizar un parseo sobre una cadena que ya ha sido analizada antes.

Luego de esta verificación el sistema pasa a traducir toda la instrucción SQL de una manera más fácil de interpretar para la maquina, aquí se realiza lo que se llama árbol de query o un árbol abstracto de sintaxis. Luego de haber hecho la conversión tenemos la verificación de la sintaxis de la instrucción. Si se detecta algún error de sintaxis se detiene el proceso y se devuelve un error, esto es igual para todas las etapas siguientes.



61

Bases de Datos 2



62

Bases de Datos 2



Después se tiene la verificación semántica, aquí se realiza un análisis más a fondo, por ejemplo si estamos llamando a una tabla o a un campo que no existe detectamos estos errores. Una vez que la sintaxis y la semántica han sido aprobadas, se procede a convertir lo que antes era un árbol de query a una serie de instrucciones de bajo nivel. Luego de este paso podemos pasar a la etapa siguiente, a encontrar un plan de ejecución.

4.1.1.2 Plan de ejecución

En esta etapa debemos saber qué tipo de optimizador estamos usando. Existen dos tipos de optimizadores, tenemos el optimizador por costos y el optimizador por reglas, los cuales veremos en detalle en una sección posterior.

Sin importar que tipo de optimizador se tenga el objetivo es el mismo, queremos obtener los datos de la manera más rápida posible. Lo que varía es la técnica o el plan utilizado para llegar a este objetivo.

En esta etapa se toman todas las operaciones de bajo nivel que necesitamos para ejecutar la consulta completa. Luego se genera una serie de posibles planes, se analiza cada uno de estos planes y se escoge el más conveniente. Para asignarle un valor numérico a la eficiencia de cada algoritmo necesitamos un método de evaluación, dependiendo del tipo de optimizador que se tenga este método de evaluación cambia.

Un elemento muy importante en esta etapa es la utilización de índices, se deben de tener siempre en cuenta ya que son una poderosa herramienta de búsqueda.

4.1.1.3 Fetch

En esta etapa ya se tiene resuelto el problema de cómo ir a traer los datos. El algoritmo ha sido definido por la etapa de Plan de ejecución. Como nos dice la palabra misma en ingles Fetch, en esta etapa simplemente se van a traer los datos.

4.1.2 Optimizador por regla

Vamos a explicar la manera en que funciona el optimizador por reglas, como lo indica su nombre se basa en una serie de reglas para estimar cual es el mejor plan a ejecutar.

4.1.2.1 Análisis de una instrucción

Al utilizar un optimizador por reglas es crucial saber la manera en que se analizan las instrucciones. Debemos saber que el parseo de una instrucción de SQL se realiza de derecha a izquierda y de abajo hacia arriba.



63

Bases de Datos 2



64

Bases de Datos 2

Figura 20 – Páginas 65-68 Libro “Bases de Datos 2”

```
SQL> SELECT LPAD(' ',2*(LEVEL-1))||operation operation, options,
2 object_name, position
3 FROM plan_table
4 START WITH id = 8 AND statement_id = 'M1 plan'
5 CONNECT BY PRIOR id = parent_id AND
6 statement_id = 'M1 plan';
```

Figura No. 9 – Orden de procesamiento

Es por esta razón que siempre debemos de colocar las instrucciones que dividen los datos en la mayor cantidad hasta abajo en orden de importancia. Por ejemplo si se está lidiando con una tabla que tiene 1000 registros y la última instrucción que colocamos nos deja únicamente con 100 registros con que trabajar, reducimos el tiempo de procesamiento considerablemente al lidiar ahora con una cantidad mucho menor de registros. Por esta razón siempre se colocan más abajo las instrucciones que dividen más los datos.

```
select at.nombre, e.nombre, es.nombre from
asignacion a, atleta at, entrenador e, estudiante es
where a.atleta = at.codigo
and a.entrenador = e.codigo
and a.estudiante = es.codigo
and a.calificacion >= 61
order by at.nombre, e.nombre, es.nombre;
```

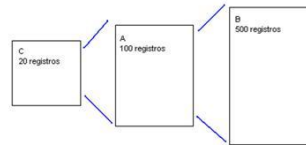
Figura No. 10 – Instrucción discriminante

Podemos ver a partir del ejemplo de arriba que tenemos dentro de la instrucción WHERE 4 diferentes instrucciones discriminatorias. La instrucción que reduce en mayor cantidad el número de registros está subrayada, de esta manera estamos optimizando el tiempo de procesamiento.

Este concepto es prácticamente el mismo cuando estamos tratando con tablas, esto lo vemos al conocer el concepto de la tabla pivot.

4.1.2.2 Tabla pivot

El orden en que se toman las tablas al realizar un SELECT puede afectar el tiempo de procesamiento de una consulta compleja. Cuando estamos utilizando un optimizador por reglas la tabla con la cual se comienza el análisis se llama la tabla pivot. La tabla pivot siempre debe ser la tabla más pequeña en nuestro select, la comprobación de esta afirmación se comprende mejor por medio de esta imagen:



65

Bases de Datos 2

66

Bases de Datos 2

Figura No. 11 – Tabla Pivot

Si comenzamos con la tabla C en este ejemplo tenemos únicamente 20 registros a partir de los cuales realizamos el pivot para seguir con la siguiente tabla. Se le llama pivot porque a partir de cada uno de los registros de la tabla A vamos a explorar todos los registros de la tabla C. Al utilizar la tabla A como pivot estamos reduciendo el número de búsquedas. Si por el contrario comenzamos con la tabla B de 500 registros estamos duplicando el número de búsquedas porque para cada uno de los registros de la tabla B vamos a buscar en la siguiente tabla.

Este ejemplo sigue el mismo principio que se vio en la sección anterior, dividir al mayor grado posible el número de registros con los cuales se trabaja con el fin de tener cada vez menos registros en el siguiente procesamiento.

4.1.2.3 Hints

Existe una manera por medio de la cual podemos indicar directamente que tipo de plan de ejecución queremos seguir para una instrucción determinada. Por medio de los Hints le indicamos al optimizador que orden debe seguir para buscar la manera óptima de obtener la respuesta.

La manera en que se le indica un Hint al optimizador es encerrando entre comentarios la instrucción que se desea, este

comentario se coloca exactamente después de la palabra SELECT, vamos a ver un ejemplo:

```
SELECT /*+ index(t1 t1_idx) index(t2 t2_idx) */
COUNT(*) FROM t1, t2
WHERE t1.c1 = t2.c1;
```

Aquí le estamos indicando al optimizador que lo primero que debe hacer es utilizar el índice t1_idx para la columna t1 y luego utilizar el índice t2_idx para la columna t2. De la misma manera existe una gran variedad de Hints diferentes que podemos utilizar con el mismo fin.

4.1.3 Optimizador por costos

El optimizador por costos se basa en el uso de estadísticas para determinar cuál es el mejor plan a utilizar. Actualmente la generación y manejo de estadísticas para los DBMS ha mejorado mucho, por esta razón actualmente el optimizador por costos es el optimizador que recomiendan utilizar la mayoría de autores.

Para simplificar su explicación el cálculo de estas estadísticas se basa en el número de lecturas lógicas realizadas a los registros de las bases de datos. Una lectura lógica se da cuando vamos a obtener datos que residen actualmente en la memoria RAM. Por el contrario una lectura física se da cuando no existe el dato que

67

Bases de Datos 2

68

Bases de Datos 2

Figura 21 – Páginas 69-72 Libro “Bases de Datos 2”

necesitamos en memoria y tenemos que traerlo del disco duro, haciendo así la lectura física. El acceso a los datos por medio de una lectura física es mucho más rápido y por esta razón se le favorece.

- *Optimizing and Tuning Your MySQL Database*
Julie C. Meloni
Ref: <http://www.informit.com/articles/article.aspx?p=29406>

Artículos Relacionados

- *Database Optimization: Increase query performance with indexes and statistics*
Mark Kaelin
Ref: http://articles.techrepublic.com/5100-10878_11-5146588.html
- *Identify Missing Indexes Using SQL Server DMV*
Ashish Kumar Metha
Ref: http://www.sql-server-performance.com/articles/perf/Identify_Missing_Indexes_Using_SQL_Server_DMV_p1.aspx
- *Database Optimization: Measuring Oracle disk I/O speed*
Donald Burleson
Ref: http://articles.techrepublic.com/5100-10878_11-5159055.html



69

Bases de Datos 2



70

Bases de Datos 2



ALCANCE

- 5. Sistemas Distribuidos
- 5.1. Sistemas Distribuidos
- 5.1.1. Las 12 reglas de CODD
- 5.1.1.1 Autonomía local
- 5.1.1.2 No dependencia de un sitio central
- 5.1.1.3 Operación continua
- 5.1.1.4 Independencia con respecto a la localización
- 5.1.1.5 Independencia con respecto a la fragmentación
- 5.1.1.6 Independencia de replica
- 5.1.1.7 Procesamiento distribuido de consultas
- 5.1.1.8 Manejo distribuido de transacciones
- 5.1.1.9 Independencia con respecto al equipo
- 5.1.1.10 Independencia con respecto al sistema operativo
- 5.1.1.11 Independencia con respecto a la red
- 5.1.1.12 Independencia con respecto al DBMS
- 5.1.2. Problemas de los sistemas distribuidos
- 5.1.2.1 Procesamiento de consultas
- 5.1.2.2 Administración del catálogo
- 5.1.2.3 Propagación de actualizaciones
- 5.1.2.4 Control de recuperación
- 5.1.2.5 Control de concurrencia

Capítulo 5

SISTEMAS DISTRIBUIDOS DE BASES DE DATOS

En este capítulo vemos la manera en que se manejan las bases de datos distribuidas. Vemos los diferentes problemas con los cuales se debe lidiar para manejarlas y vemos algunas reglas básicas. Vemos conceptos clave como la replicación, la actualización de los datos que son cruciales en este tipo de sistemas.



71

Bases de Datos 2



72

Bases de Datos 2

Figura 22 – Páginas 73-76 Libro “Bases de Datos 2”

- 5.1.3. Commit de dos fases
- 5.2. Diseño de sistemas distribuidos
 - 5.2.1. La distribución de los datos
 - 5.2.1.1 Centralizada
 - 5.2.1.2 Distribuida
 - 5.2.1.3 Replicación
 - 5.2.2. Modelos de replicación
 - 5.2.2.1 Localidad primaria
 - 5.2.2.2 Localidad secundaria
 - 5.2.3. Replicación simétrica y asimétrica
 - 5.2.3.1 Replicación asimétrica
 - 5.2.3.2 Conflictos de replicación
 - 5.2.3.3 Replicación simétrica

5.1 Sistemas Distribuidos

Un sistema distribuido de bases de datos es un sistema en el cual participan varias maquinas que pueden encontrarse en lugares remotos, todas estas maquinas tienen acceso al mismo sistema de bases de datos. Existe una serie de reglas y consideraciones especiales que debemos tomar en cuenta al tener una situación como esta, el manejo de un sistema distribuido hace surgir nuevos problemas sobre el manejo y la actualización de los datos.

5.1.1 Las 12 reglas de CODD:

Existe un principio en el cual se basan las bases de datos distribuidas, su regla principal es que desde el punto de vista del usuario el sistema distribuido debe verse exactamente igual a un sistema normal. En otras palabras el usuario no debe notar la diferencia o la existencia de la base de datos distribuida. Las doce reglas de CODD son reglas derivadas de la regla principal de las bases de datos distribuidas.

5.1.1.1 Autonomía Local

Es muy importante que cada uno de los miembros de un sistema distribuido tenga autonomía, lo importante aquí es que no haya una dependencia entre dos lugares diferentes. Si nosotros queremos realizar una operación en nuestra base de datos debemos tener la capacidad de realizar estas operaciones en ese

73

Bases de Datos 2

74

Bases de Datos 2

lugar y nunca tener la necesidad de ir a otro lugar para realizar estas operaciones.

Como estamos lidiando con un sistema distribuido es muy probable que los datos que necesitamos no estén necesariamente en el lugar donde se está trabajando, pero las operaciones que se quieren realizar si se deben poder realizar desde allí.

5.1.1.2 No dependencia de un sitio central

Este objetivo es similar al punto número uno y de hecho si se logra obtener la autonomía local se logra la no dependencia de un sitio central también. A pesar de esto este punto es importante porque de no lograr llegar a una autonomía local completa por lo menos se debe llegar a cumplir este punto.

Por ejemplo si en un sistema distribuido tenemos una maquina dedicada a realizar consultas y otra maquina por medio de la cual se realizan las configuraciones del sistema, en estas maquinas tenemos dependencias de un sitio central. Debemos evitar estos casos ya que podrían generar un cuello de botella en los sitios centrales y lo peor de todo es si uno de estos puntos falla causa la falla del sistema completo.

5.1.1.3 Operación continua

En todo sistema distribuido se requiere de un sistema de alta disponibilidad, el sistema de bases de datos debe tener la capacidad de seguir funcionando en todo momento aunque se tengan que realizar operaciones de mantenimiento o de actualización.

5.1.1.4 Independencia con respecto a la localización

En un sistema distribuido no debe importar el lugar donde están los datos, especialmente desde el punto de vista del usuario. Cuando vamos a acceder a los datos en el sistema se debe de ver como si los datos estuvieran en la misma maquina donde están siendo solicitados, por esta razón al usuario no debe importarle donde están los datos que busca.

5.1.1.5 Independencia con respecto a la fragmentación

La fragmentación se da cuando nosotros dividimos una tabla y sus partes se colocan en lugares diferentes. Por ejemplo si tenemos una tabla de ventas y tenemos tres sucursales, los registros que más se usan en cada sucursal son los registros locales, por lo tanto sería conveniente fragmentar esta tabla. La fragmentación ayuda a mejorar el desempeño de la base de datos y aliviar carga sobre la red.

75

Bases de Datos 2

76

Bases de Datos 2

Figura 23 – Páginas 77-80 Libro “Bases de Datos 2”

Ahora a pesar de haber fragmentado los datos, en un sistema distribuido esto también debe ser transparente al usuario. Si el usuario quiere ver los datos de ventas de todas las sucursales no debe de buscar en la tabla de cada sucursal, desde su punto de vista únicamente existe una tabla con los valores de todas las ventas.

5.1.1.6 Independencia de replicas

Ahora vemos el caso de las replicas de datos. La replicación de datos consiste en tener un mismo grupo de datos en diferentes lugares, esto nos da más seguridad en caso de perder los datos en uno de los lugares el sistema siempre seguirá funcionando mientras una de las fuentes este activa.

Al igual que con la fragmentación, la replicación en un sistema distribuido debe ser completamente transparente al usuario. El usuario al buscar la información no necesita saber que la información esta replicada y que debe actualizarse en varios lugares, este es un proceso automático.

5.1.1.7 Procesamiento distribuido de consultas

Este principio nos enseña que todo sistema distribuido debe estar en una base de datos relacional, ya que aqui las consultas pueden ser optimizadas. La optimización en los sistemas distribuidos puede hacer la diferencia entre obtener los datos en

unos segundos y tenerlos en varias horas, esto se da por la lejanía entre los datos mismos y las limitaciones de conexión.

5.1.1.8 Manejo distribuido de transacciones

Hemos visto los problemas que surgen para mantener la integridad de los datos utilizando transacciones, este problema empeora en caso de los sistemas distribuidos. Para resolver todos los problemas de coordinación para el manejo apropiado de transacciones se utiliza lo que se conoce como el Commit de dos fases, este concepto se verá más adelante.

5.1.1.9 Independencia con respecto al equipo

Sin importar el equipo de cómputo que se tenga en los diferentes lugares del sistema distribuido es importante que todas las máquinas sean capaces de ejecutar el mismo sistema de base de datos, y que sean capaces de soportar todas las operaciones que se realizan en una base de datos.

5.1.1.10 Independencia con respecto al sistema operativo

Este principio es similar al anterior pero ahora lidiamos con el sistema operativo. De la misma manera sin importar que sistema operativo se utilice en las máquinas se debe utilizar el mismo sistema de base de datos.

77

Bases de Datos 2

78

Bases de Datos 2

5.1.1.11 Independencia con respecto a la red

Para un sistema distribuido localizado en diferentes lugares es necesario tener la capacidad de poder manejar diferentes redes. El número de redes o la distancia entre ellas no deben ser un problema para comunicarse entre los diferentes lugares que se manejan.

5.1.1.12 Independencia con respecto al DBMS

Según C.J. Date este paso nos dice que de tener varios DBMS distintos que manejen la misma interfaz se lograría la coordinación necesaria para trabajar de manera distribuida. Date nos dice que un sistema de bases de datos distribuido ideal debería cumplir con este principio. En la práctica este paso se ignora en la mayoría de los casos.

5.1.2 Problemas de los sistemas distribuidos

5.1.2.1 Procesamiento de consultas

Como explicamos antes en los doce puntos de CODD, las limitaciones de ancho de banda hacen crucial la optimización en el procesamiento de consultas en un ambiente distribuido. El optimizador es el encargado de mantener el número de operaciones realizadas al mínimo y en consecuencia el uso de la red.

5.1.2.2 Administración de catálogo

El catálogo de la base de datos además de tener la información básica sobre la base de datos completa tiene también la información necesaria para asegurar la autonomía local. Esto se traduce a tener información sobre la localización, fragmentación y la réplica utilizada en todo sistema distribuido.

Para evitar la dependencia de un sitio central y la carga de tener replicas completas en cada lugar de trabajo, se realiza un proceso que minimiza el acceso a catálogos remotos teniendo una serie de catálogos con la información de los objetos creados en cada sitio.

5.1.2.3 Propagación de actualizaciones

En sistemas donde existe la replicación se tiene el problema de tener que actualizar constantemente los datos. No se puede realizar una actualización constante después de cada cambio porque sería demasiada carga para el sistema.

Aquí se utiliza el método de la llave primaria, se tiene un lugar centralizado que siempre esta actualizado y que en un momento posterior se encarga de actualizar las demás replicas.

5.1.2.4 Control de Recuperación

79

Bases de Datos 2

80

Bases de Datos 2

Figura 24 – Páginas 81-84 Libro “Bases de Datos 2”

En los sistemas distribuidos el control de recuperación esta basado en el commit de dos fases, aqui se tiene el mismo sistema que mencionamos en la sección de transacciones pero aplicado a un ambiente distribuido, este tema se detalla en la siguiente sección.

5.1.2.5 Control de concurrencia

El control de concurrencia al igual que con sistemas normales está basado en el principio de transacción. La manera de solucionar esto aquí también es por medio de candados, hacer la solicitud de acceso a un registro y utilizar candados. Cuando se habla de sistemas distribuidos el manejo de candados se convierte en mensajes, aquí se utiliza una técnica muy similar al método de llave primaria pero aplicado al manejo de candados.

5.1.3 Commit de dos fases

Hemos hablado sobre los problemas de coordinación a la hora de manejar transacciones y los problemas de acceso concurrente que pueden haber, cuando hablamos de sistemas distribuidos llegamos a otro nivel de dificultad. Por ejemplo si estamos manejando tres lugares diferentes en el sistema y el lugar A realiza un commit, este commit se debe ver reflejado también en los lugares B y C. La tarea de coordinar estas operaciones está a cargo de El Coordinador, el está encargado de realizar las operaciones de commit o rollback a todas las estaciones dentro del sistema.

81

Bases de Datos 2

El proceso de commit para un sistema distribuido se divide en dos fases para poder manejar todos los sitios del sistema, vamos a explicar paso a paso como se realiza la confirmación luego de una transacción. La primera fase consiste en otorgarle el poder al Controlador, en este momento los administradores de recursos de cada sitio simplemente se quedan esperando a las ordenes del Controlador. El Controlador tiene su bitácora independiente para manejar todo este proceso.

Luego de haberse ejecutado la transacción o haber recibido algún mensaje de error en el camino el Controlador registra el cambio en su bitácora, en este momento comienza la segunda fase. Ahora procede a avisar a cada uno de los administradores locales del cambio. Cada administrador local se encarga de realizar los cambios en su bitácora local, aquí se da por terminado el proceso completo.

5.2 Diseño de sistemas distribuidos

82

Bases de Datos 2

5.2.1 La distribución de los datos

La manera en que se maneja la información en un sistema distribuido puede ser manejada con enfoques diferentes. Tenemos tres técnicas principales que dictan la manera en que organizamos y manejamos los datos.

5.2.1.1 Centralizada

En una distribución centralizada los datos residen en un lugar específico podríamos considerar esto como un esquema cliente servidor. El servidor contiene los datos y los diferentes clientes que se encuentran en lugares remotos deben ir a este lugar para buscar la información.

5.2.1.2 Distribuida

En un sistema distribuido ya no tenemos este esquema. La información se encuentra repartida entre varios de los lugares que conforman el sistema. Es posible que tras realizar una consulta tengamos que traer los datos desde varios lugares distintos para obtener la información.

5.2.1.3 Replicación

En un sistema donde existe la replicación llevamos el manejo de la información un paso más lejos. Al mencionar replicar nos referimos a tener en lugares diferentes la misma información, esto

83

Bases de Datos 2

se realiza por fines de seguridad y desempeño. Es posible también que en este tipo de sistemas la información está también distribuida.

5.2.2 Modelos de replicación

El tema de replicación de datos involucra un proceso complejo de sincronización. La mayor parte de los problemas surgen a la hora de actualizar la información, existen diferentes formas por medio de las cuales logramos crear la replicación y estas son las clasificaciones.

5.2.2.1 Localidad primaria e instantáneas

En este modelo tenemos primero una tabla primaria (para cada tabla en la base de datos), puede haber una o más replicas. Al lugar donde se encuentra la tabla primaria se le llama Sitio Primario. La característica más importante de este modelo es que las replicas de la tabla primaria son solo de consulta, únicamente podemos ver datos no podemos realizar operaciones sobre ellas. Las modificaciones únicamente se pueden hacer en la tabla primaria.

5.2.2.2 Replicación secundaria o distribuida

En este modelo las replicas de la tabla primaria si permiten modificaciones, ya que las replicas son tablas completas. La

84

Bases de Datos 2

Figura 25 – Páginas 85-88 Libro “Bases de Datos 2”

replicación de las tablas primarias se realiza con snapshots actualizables, cada cierta cantidad de tiempo se realiza una actualización asíncrona.

5.2.3 Replicación simétrica y asimétrica

5.2.3.1 Replicación asimétrica

En la replicación asimétrica las actualizaciones para los datos no se realizan al momento de haber cambios en los mismos. La actualización se realiza posteriormente, pero este método a pesar de ayudar considerablemente en el desempeño genera lo que llamamos conflictos de replicación.

5.2.3.2 Conflictos de replicación

El proceso de actualizar tablas replicadas de manera asíncrona puede generar conflictos. Por ejemplo si tenemos tres estaciones de trabajo y se realizan cambios en una de las estaciones, podría darse el caso que en otra de las estaciones se esté trabajando con los mismos datos y aquí también se realicen cambios antes de lograr hacer la actualización. En este caso se da el conflicto de replicación.

Existen técnicas para saber qué datos utilizar en caso de conflictos. Primero tenemos la prioridad, simplemente asignamos un mayor grado de prioridad a un servidor y como este es

actualizado más seguido se opta por obtener los datos de aquí primero. Otra técnica es utilizar ya sea el valor más grande o el valor más pequeño a la hora de tener un conflicto para minimizar problemas. Finalmente tenemos los timestamps, a cada transacción se le deja una fecha y una hora, en este caso se toma la transacción mas reciente.

5.2.3.3 Replicación simétrica

La replicación simétrica se realiza en el momento en que surge un cambio, este tipo de replicación tiene una tasa alta en el desempeño de la base de datos. Existen dos maneras de realizar replicaciones. En la replicación por datos se mandan los registros actualizados para las tablas que los necesitan, en la replicación por instrucciones se mandan instrucciones de SQL para ser ejecutadas en el otro servidor. Cuando se están actualizando objetos como por ejemplo índices siempre se manda la instrucción en SQL.

Artículos Relacionados

- *An Active Web-based Distributed Database System for E-Commerce*
Hiroshi Ishikawa, Manabu Ohta
Ref. <http://www.dcs.bbk.ac.uk/webDyn/webDynPapers/ishikawa.pdf>

85

Bases de Datos 2

86

Bases de Datos 2

- *Data Allocation in distributed database systems performed by mobile intelligent agents*

Horea Grebla, Grigor Moldovan, Sergiu Adrian Darabant, Alina Câmpan
Khawar M. Zuberi and Kang G. Shin
Ref. <http://www.emis.de/journals/AUA/acta8/Grebla-Moldovan-Darabant-Campan.pdf>

- *Transaction Management in Distributed Database Systems: the Case of Oracle's Two-Phase Commit*

Ghazi Alkhatib, Ronny S. Labban
Ref. <http://www.jise.appstate.edu/Issues/13/095.pdf>

- *Performance Analysis of distributed database systems*

Michael Stonebraker, John Woodfill, Jeff Ranstrom, Joseph Kalash,
Kenneth Arnold and Erika Andersen
Ref. <https://reprints.kfupm.edu.sa/57518/1/57518.pdf>

Capítulo 6

ANÁLISIS MULTIDIMENSIONAL Y DATA WAREHOUSE

En este capítulo vemos el concepto de las bases de datos multidimensionales y en general el concepto de Business Intelligence. Vemos los conceptos principales que se manejan en un data warehouse. Al final explicamos la manera en que se utiliza un data warehouse, esto es por medio del DSS y de técnicas de Data Mining.

87

Bases de Datos 2

88

Bases de Datos 2

Figura 26 – Páginas 89-92 Libro “Bases de Datos 2”



ALCANCE

- 6. Análisis Multidimensional y Data Warehouse
 - 6.1. Bases de Datos Multidimensionales
 - 6.1.1. OLAP, ROLAP, MOLAP
 - 6.1.2. Business Intelligence
 - 6.2. Data Warehouse
 - 6.2.1. Data Warehouse y Data Mart
 - 6.2.1.1. Data Warehouse
 - 6.2.2. Data Mart
 - 6.2.2. Diseño de un Data Warehouse
 - 6.2.2.1. Modelo Estrella
 - 6.2.2.2. Modelo Snow Flake
 - 6.2.3. ETL
 - 6.3. Componentes de un Data Warehouse
 - 6.3.1. Fact Table
 - 6.3.2. Dimensiones
 - 6.3.3. Niveles
 - 6.4. Operando con el Data Warehouse
 - 6.4.1. DSS
 - 6.4.2. Data Mining

6.1 Bases de datos Multidimensionales

Se utilizan principalmente para crear aplicaciones OLAP, por la manera en que funcionan estas bases de datos.

Esta optimizada para realizar consultas sobre esta, por lo que operaciones de select son mucho más rápidas en este tipo de bases de datos. Esto es posible ya que los datos usualmente están des normalizados evitando perder tiempo en realizar joins.

Operaciones de Inserción, eliminación y update son más lentas, ya que los datos están replicados en varios lugares. Es por esta razón que se usa este tipo de base de datos para la realización de un Data Warehouse.

6.1.1 OLAP, ROLAP, MOLAP

OLAP son las siglas para On-line analysis processing, las bases de datos OLAP son las que hemos descrito arriba. De aquí tenemos dos divisiones principales.

Tenemos las bases de datos MOLAP, la M por Multidimensional y tenemos las bases de datos ROLAP la R por relational. Ambos tipos de bases de datos tienen el mismo objetivo pero lo hacen bajo un sistema diferente. Las bases de datos MOLAP trabajan bajo un sistema multi dimensional, las bases de datos ROLAP lo hacen bajo el sistema tradicional relacional que conocemos.



6.1.2 Business Intelligence

El concepto de business intelligence se refiere a la ventaja competitiva que logra una empresa al tener información más actualizada y más rápida que le permite tomar decisiones cruciales sobre su negocio.

La clase de información que se puede tener con un buen sistema de Bi puede influir en todos los aspectos de la estrategia de una empresa.

Los niveles altos de la gerencia utilizan estas herramientas para obtener información variada, por ejemplo tomemos el caso de una empresa de venta de ropa.

Con un sistema de Bi podemos averiguar datos como:

Que ropa se vende más.

Que sucursales venden más.

Quien es el mejor vendedor.

Que proveedores no nos están sirviendo.

Por qué un mes se vende más que otro.

Y otros tipos de datos que normalmente obtenemos por medio de reportes, la diferencia aquí es que estos datos se obtienen de manera dinámica y en cuestión de segundos.

Pero tal vez la herramienta más poderosa está en la predicción basada en modelos de regresión. Podemos predecir

cómo serán las ventas para el siguiente año, que prendas se venderán mas, etc. Y este es el tipo de información que le da una ventaja competitiva a una empresa, influyendo en decisiones importantes.

6.2 Data Warehouse

6.2.1 Data Warehouse y Data Mart

6.2.1.1 Data Warehouse

Un Data Warehouse es utilizado para un propósito general, obtener datos y tendencias de manera muy rápida basándonos en datos históricos de una empresa.

Cuando hablamos de datos históricos nos referimos a datos de al menos unos 10 años atrás, ya que es de esta manera que trabaja un Data Warehouse.

6.2.1.2 Data mart

Un Data Mart es una porción del data warehouse completo, que puede funcionar independientemente. Es un subconjunto del conjunto completo que es el DW y este subconjunto está enfocado a un área específica. El único problema es que no se tienen todos los datos, pero se puede trabajar con el de la misma forma que con un data warehouse.



Figura 27 – Páginas 93-96 Libro “Bases de Datos 2”

6.2.2 Diseño de un Data Warehouse

En general existen dos esquemas con los cuales trabaja un modelo de Data Warehouse, tenemos el modelo estrella y el modelo snow flake.

6.2.2.1 Modelo estrella

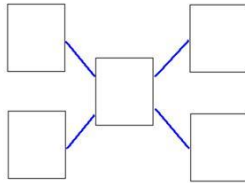


Figura No. 12 – Modelo Estrella

En este tipo de modelo la información se almacena en dimensiones en las cuales todos los datos están desnormalizados. Este modelo es el que más se utiliza, y en algunas herramientas de Data Warehouse la única opción que se tiene.

93

Bases de Datos 2

6.2.2.2 Modelo Snow flake

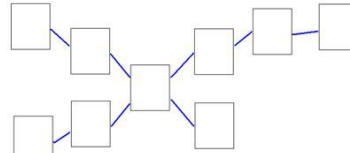


Figura No. 13 – Modelo Snow Flake

En este modelo las dimensiones si están normalizadas. A pesar de esto el objetivo de esta base de datos sigue siendo el mismo. Este modelo no es muy popular entre desarrolladores.

6.2.3 ETL

El proceso de Extraction Transformation Load (ETL) consiste básicamente en extraer los datos de varias fuentes diferentes y convertirlos de manera que se puedan subir a la base de datos multidimensional.

El proceso de ETL puede ser tan complejo como lo requiera la situación, ya que este proceso debe ser capaz de obtener datos tanto de bases de datos como de archivos.

94

Bases de Datos 2

En general se puede dar el caso de tener información en dos bases de datos relacionales en diferentes DBMS, otro conjunto de datos en hojas de excel y otro grupo de datos en archivos de texto separados por coma.

Existen herramientas para solucionar la carga de datos en estas situaciones y de hecho automatizarla ya que es una de las finalidades.

La manera en que se trabaja es tomar la información de la fuente directamente y por medio de conversiones adaptarla a nuestro modelo multidimensional.

Luego de tener el proceso listo, debemos prepararlo para que pueda trabajar de manera automática.

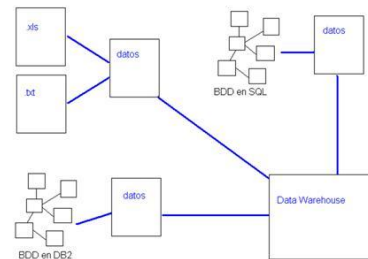


Figura No. 14 – Fuentes de Datos

6.3 Componentes de un Data Warehouse

Existe una serie de componentes que conforman todo data warehouse y conceptos que se manejan que debemos tener claros al armar un modelo multidimensional.

95

Bases de Datos 2

96

Bases de Datos 2

Figura 28 – Páginas 97-100 Libro “Bases de Datos 2”

6.3.1 Fact Table

Es la tabla en la cual se centraliza la información, aquí es importante notar que prácticamente solo se manejan datos numéricos, y en mayor parte llaves primarias que son unidas a otras dimensiones.

Cuando hablamos de dimensiones, estamos hablando de elementos de datos que usualmente están conformados por varios niveles. Como habíamos dicho anteriormente, los datos están desnormalizados para ahorrar tiempo, es por esto que se crean las dimensiones.

6.3.2 Dimensiones

Un ejemplo claro de la creación de una dimensión puede ser este modelo sencillo:

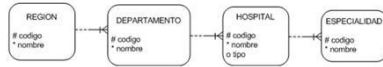


Figura No. 15 – Dimensión en niveles

Tenemos cuatro tablas, ninguna de estas entidades tiene razón de existir por sí sola realmente, por esta razón vamos a crear la dimensión Especialidad.

Cabe notar que los códigos de las tablas que fusionamos no se incluyen, solamente los campos importantes.

En general la manera en que creamos nuestras dimensiones está dirigida a optimizar el modelo multidimensional.



Figura No. 16 – Dimensión

6.3.3 Niveles

Los niveles como podemos intuir de la imagen anterior, son la jerarquía en la cual se puede configurar la dimensión.

Para la dimensión Especialidad los niveles son:

- Región
- Departamento
- Hospital
- Especialidad
- Métricas

La combinación de las métricas adecuadas con las dimensiones necesarias nos dan los datos que necesitamos.

97

Bases de Datos 2

98

Bases de Datos 2

Ejemplos usuales de métricas pueden ser: # de ventas, precio, utilidad, ganancia, etc. En general valores que son relevantes al hacer consultas.

La combinación de las métricas adecuadas con las dimensiones necesarias nos dan los datos que necesitamos.

Ejemplos usuales de métricas pueden ser: # de ventas, precio, utilidad, ganancia, etc. En general valores que son relevantes al hacer consultas.

6.4 Operando con el Data Warehouse

6.4.1 Decision Support System (DSS)

Hasta este punto hemos visto como se construye el data warehouse pero no hemos entrado en detalle con el tema de su utilización. El usuario final de un data warehouse necesita una interfaz amigable con la cual puede mover dimensiones y cambiar métricas a su gusto.

El DSS permite que las personas que toman las decisiones en la empresa utilicen la información base que se encuentra en el data warehouse, sin el DSS la información en sí no tiene ningún significado real. Existen diferentes tipos de DSS dependiendo del DBMS que se está utilizando, por ejemplo para SQL Server el DSS que se utiliza es Excel.

6.4.2 Data Mining

Este es un tema que involucra las áreas de inteligencia artificial y la estadística, el objetivo es encontrar tendencias escondidas en la información. Durante el proceso de DSS nosotros movemos dimensiones y escogemos métricas para asociar la información que nosotros queremos ver. Durante el proceso de data mining no se busca una dimensión específica, estamos buscando entre varias fuentes de información y utilizando algunos parámetros base buscamos algún tipo de relación entre los datos.

Otra de las funciones más importantes que se desarrolla en esta etapa es la creación de modelos de regresión, utilizando datos históricos de la base de datos podemos predecir el comportamiento en el futuro de los datos. De esta manera se pueden crear los planes estratégicos que describimos en la sección de business intelligence.

99

Bases de Datos 2

100

Bases de Datos 2

Figura 29 – Páginas 101-104 Libro “Bases de Datos 2”



Artículos relacionados

- **Business Intelligence not just for bosses anymore**
Meridith Levinson
Ref: http://www.cio.com/article/16544/Business_Intelligence_Not_Just_for_Bosses_Anymore
- **Slowly Changing Dimensions Are Not Always as Easy as 1, 2, 3**
Margy Ross y Ralph Kimball
Ref: <http://www.intelligententerprise.com/showArticle.jhtml?articleID=59301280>
- **Tomorrow's real time ETL Imperative**
Dan Linstedt
Ref: <http://www.teradata.com/tpage/127085/index.html>
- **Data Mining on the web**
Dan R. Greening
Ref: <http://www.webtechniques.com/archives/2000/01/greening/>

Glosario

ACID: De sus siglas en inglés, las cuatro propiedades básicas de una transacción, estas son: atomicity, consistency, isolation y durability

Deadlock: estado de un sistema en el cual un objeto obstruye el camino de otro deteniendo el progreso de un sistema completo.

Commit: Acción por medio de la cual se le indica a la base de datos que los cambios que se han realizado son permanentes.

Rollback: Acción por medio de la cual se pueden deshacer las acciones realizadas sobre una base de datos.

Fetch: este verbo escrito en inglés se puede traducir como: ir a traer algo.

Parser: Aplicación que tiene como objetivo leer una porción de texto y por medio de una lista de palabras predefinidas se valida.

Hint: Instrucción por medio de la cual se le indica al optimizador de la base de datos que método queremos que utilice para traer los datos.

Backup: Proceso por medio del cual se realiza una copia de seguridad de cierto contenido para una posible recuperación posterior.

ROLAP: Relational OLAP, tiene el mismo significado que OLAP pero aplicado a bases de datos relacionales.

MOLAP: Multidimensional OLAP, tiene el mismo significado que OLAP pero aplicado a bases de datos multidimensionales.



101

Bases de Datos 2



102

Bases de Datos 2



Warehouse: Almacén en inglés, utilizado en el término Data Warehouse.

Fact Table: La traducción literal a español es tabla de hechos, es un componente utilizado en el diseño de un Data Warehouse.

Snowflake: En inglés copo de nieve. El esquema snowflake es una de las dos posibles opciones a utilizar en un Data Warehouse.

ETL: Proceso por medio del cual se carga información a un Data Warehouse a partir de una o más fuentes de datos.

DSS: Decision Support System o sistema de soporte para toma de decisiones en inglés. Medio por el cual se interactúa con la información de un Data Warehouse.

Referencias Bibliográficas

- Lillian Hobbs, Susan Hillson, Shilpa Lawande, Pete Smith. *Oracle Database 10g Data Warehousing*.
- Maria Sueli Almeida, Missao Ishikawa, Joerg Reinschmidt, Torsten Roerber. *Getting started with Data Warehouse and Business Intelligence*.
- Evan Marcus y Hal Stern. *Blueprints for High Availability*. Wiley Publishing.
- Paul Nielsen. *SQL Server 2005 Bible*. Wiley Publishing.
- C.J. Date, *Introducción a los sistemas de bases de datos, Quinta Edición, 1992*. Addison Wesley, Iberoamérica.
- Elmasri y Navathe. *Sistemas de Bases de datos, Segunda Edición*. Addison Wesley, Iberoamérica.
- Ron Soukup, *Microsoft SQL Server a Fondo*. Editorial McGraw Hill 1997.



103

Bases de Datos 2



104

Bases de Datos 2

Figura 30 – Páginas 105-108 Libro “Bases de Datos 2”

Ya que la mayoría de agentes de este tipo se especializan en un medio determinado, cuando su medio cambia no logran lidiar bien con este cambio. La especialización del medio incrementa su desempeño, pero para casos en los cuales el medio es variable se debe preparar al agente para este cambio.

Existen tres sub clasificaciones para los agentes reactivos estos son:

6.3.1.1 Agentes orientados a una meta

Este tipo de agentes tiene un grado más alto de autonomía comparado a un agente reactivo normal. En lugar de basarse en un conjunto de reglas preestablecidas para reaccionar ante su medio un agente orientado a una meta realiza un análisis para lograr su objetivo. Este análisis está basado ya sea en búsquedas o en la planeación. Por la manera en que se comporta este tipo de agente no considera cuantos recursos ha consumido, lo único que le importa es llegar a la meta impuesta.

6.3.1.2 Agentes basados en la utilidad

Es muy similar a los agentes orientados a una meta pero a diferencia de estos los agentes orientados a la utilidad si toman en cuenta los recursos que tienen a su disposición. Aquí se tienen dos objetivos principales. La primera es llegar la meta impuesta por su usuario y la segunda es optimizar la utilización de recursos para

lograrlo. El recurso más común que se llega a optimizar en la mayoría de los casos es el tiempo.

Podemos ver ahora la diferencia entre los agentes reactivos, orientados a una meta y orientados a la utilidad. La diferencia entre estos es básicamente la motivación que tienen para llegar a su meta. Aquí se introduce lo que se considera el nivel de felicidad del agente, la felicidad se traduce al grado de éxito que tiene el mismo. La felicidad del agente es un factor que se puede llegar a utilizar en muchas maneras, por ejemplo para agentes orientados a la utilidad un recurso que se puede llegar a optimizar es la felicidad (o la probabilidad de éxito).

6.3.2 Agentes de Interfaz

Aquí la palabra interfaz realmente se refiere al usuario. Los agentes de interfaz interactúan directamente con el usuario, usualmente tienen una interfaz amigable o una buena animación para interactuar. Los asistentes interactivos caen bajo esta clasificación, por ejemplo asistentes de escritorio que ayudan en las búsquedas.

Es común que este tipo de agentes no interactúen con otros agentes únicamente con el usuario pero no es una regla, es posible de ser necesario que interactúen con otros agentes. Este tipo de agentes es utilizado comúnmente para realizar labores repetitivas



105

Inteligencia Artificial



106

Inteligencia Artificial

por parte del usuario, luego de ver varias veces las acciones del usuario se le puede indicar que este las realice por su cuenta.

6.3.3 Agentes Móviles

La movilidad tiene un significado diferente dependiendo del tipo de agente que usamos. Para agentes robóticos el medio en el cual son móviles son en la vida real, pueden caminar en un lugar específico. Para agentes de software la movilidad es en la red, ya sea en una red interna o en internet.

Utilizando agentes móviles tenemos la habilidad de distribuir el trabajo en varias computadoras e incluso usar los recursos de una computadora más poderosa a la computadora donde reside normalmente. Este tipo de movimiento también reduce el consumo de ancho de banda al realizar operaciones directamente en otras computadoras en vez de hacer solicitudes en la red. Los agentes que no tienen la capacidad de movilidad se les llaman estáticos.

6.3.4 Agentes de Información

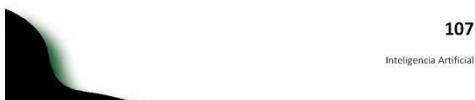
Los agentes de información son agentes utilizados para buscar información en internet. Por esto se les llama también agentes de internet aunque también se les llama agentes de recopilación de información. Este tipo de agentes son muy útiles ya que en algunos casos alivian la labor de búsqueda de información por su gran rapidez y la posibilidad de trabajar de manera asincrónica.

Un ejemplo sencillo de este tipo de agentes son los search bots utilizados por los motores de búsqueda para actualizar sus bases de datos. Agentes de información más sofisticados son capaces de aprender e incluso trabajar en conjunto con otros agentes.

6.4 Sistemas multi-agente

Una característica importante de los agentes dentro de un sistema multi-agente es que por su cuenta cada agente no puede resolver el problema completo. Cada agente aporta una parte de la solución y sus esfuerzos unidos le dan la solución al problema. De hecho cada uno de los agentes no necesita saber cuál es el objetivo final del sistema completo, lo único que debe tener claro es cuál es su sub meta dentro del sistema. Los agentes pueden ser colaborativos o competitivos, esto depende del objetivo final del sistema.

Una de las características más interesantes para agentes de este tipo es la posibilidad de poder trabajar en paralelo, en un sistema bien sincronizado se puede lograr reducir considerablemente el tiempo de procesamiento para la resolución del problema.



107

Inteligencia Artificial



108

Inteligencia Artificial

Figura 31 – Páginas 109-112 Libro “Bases de Datos 2”



has received express permission from the Licensor to exercise rights under this License despite a previous violation.

j. **“Publicly Perform”** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

k. **“Reproduce”** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. **Fair Dealing Rights.** Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. **License Grant.** Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked “The original work was translated from English to Spanish,” or a modification could indicate “The original work has been modified.”;

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

i. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License.

ii. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and

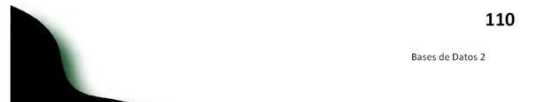
iii. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 5(f), all rights not expressly granted by Licensor are hereby reserved.

4. **Restrictions.** The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section (4a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.

b. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the “Applicable License”), you must comply with the terms of the Applicable License generally and the following provisions: (i) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (ii) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (iii) You



must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (iv) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section (4b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.

c. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections,

You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution (“Attribution Parties”) in Licensor’s copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., “French translation of the Work by Original Author,” or “Screenplay based on original Work by Original Author”). The credit required by this Section 4(c) may be implemented in any reasonable manner, provided, however, that in the case of an Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author’s honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action

prejudicial to the Original Author’s honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. **Representations, Warranties and Disclaimer.** UNLESS OTHERWISE MUTUALLY AGREED BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. **Limitation on Liability.** EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. **Termination.**

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. **Miscellaneous.**

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as are set forth in this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this



Figura 32 – Páginas 113-114 Libro “Bases de Datos 2”



License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

- d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
- f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor. Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of the License. Creative Commons may be contacted at <http://creativecommons.org/>.



7. DOCUMENTACIÓN DE APOYO DE “INTELIGENCIA ARTIFICIAL 1”

A continuación se presenta el desarrollo de la documentación de apoyo del curso de Inteligencia Artificial 1, el cual se realizó en formato de un libro conteniendo las unidades más importantes impartidas a lo largo del curso.

7.1 Datos generales

NOMBRE DE CURSO: Inteligencia Artificial 1 (972)

PRE- REQUISITOS: Organización de Lenguajes y Compiladores 2 (781)

Sistemas de Bases de Datos 2 (775)

Teoría de Sistemas 2 (724)

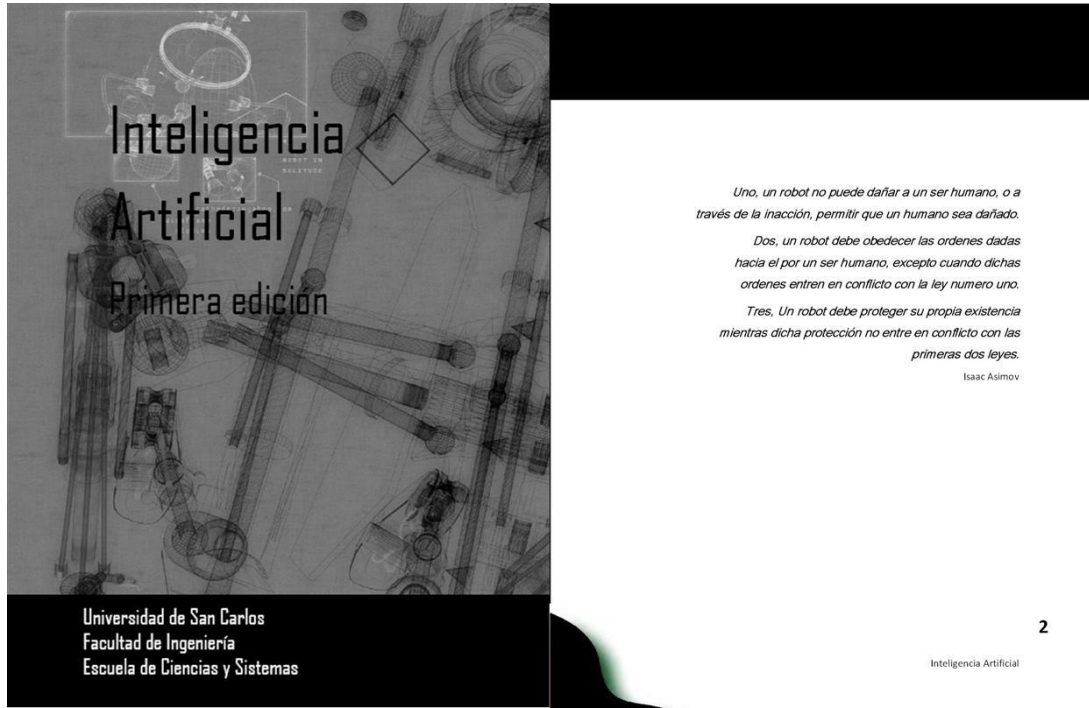
POST – REQUISITOS: Inteligencia Artificial 2 (968)

OBJETIVO: Este libro está destinado para los estudiantes del curso de Inteligencia Artificial 1, conteniendo teoría, ejemplos, diagramas, resumen, glosario y artículos de interés relacionados a cada una de las unidades del curso.

MÓDULOS:

Inteligencia Artificial

Figura 33 – Páginas 1-4 Libro “Inteligencia Artificial”

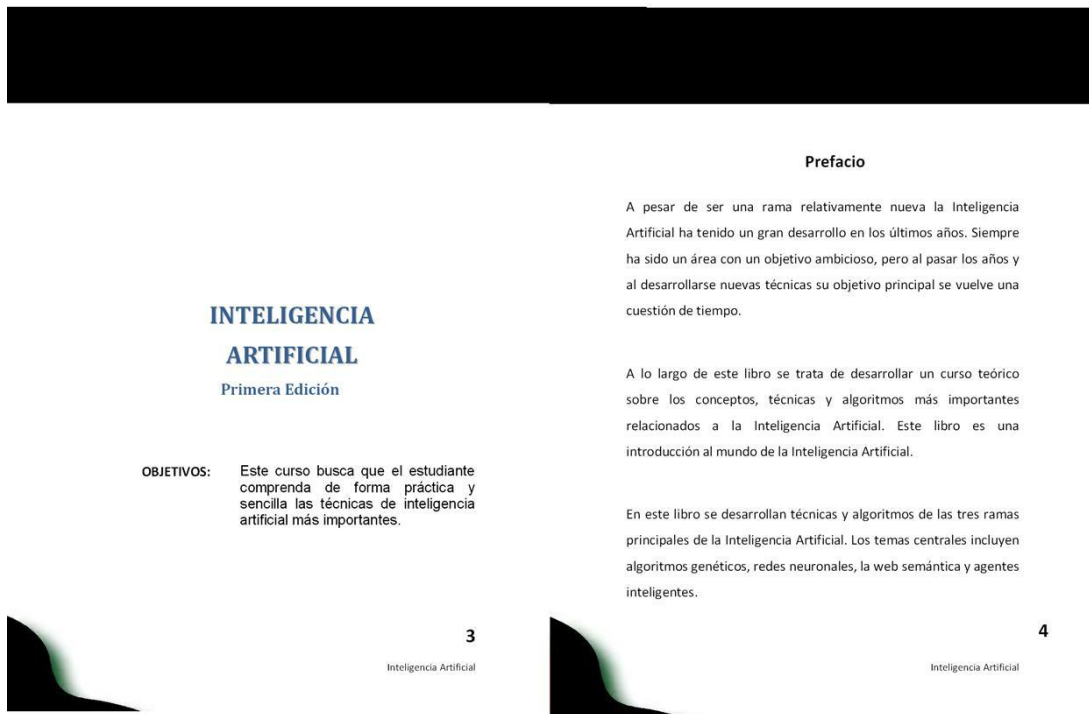


Uno, un robot no puede dañar a un ser humano, o a través de la inacción, permitir que un humano sea dañado.
Dos, un robot debe obedecer las ordenes dadas hacia el por un ser humano, excepto cuando dichas ordenes entren en conflicto con la ley numero uno.
Tres, Un robot debe proteger su propia existencia mientras dicha protección no entre en conflicto con las primeras dos leyes.

Isaac Asimov

2

Inteligencia Artificial



INTELIGENCIA ARTIFICIAL

Primera Edición

OBJETIVOS: Este curso busca que el estudiante comprenda de forma práctica y sencilla las técnicas de inteligencia artificial más importantes.

3

Inteligencia Artificial

Prefacio

A pesar de ser una rama relativamente nueva la Inteligencia Artificial ha tenido un gran desarrollo en los últimos años. Siempre ha sido un área con un objetivo ambicioso, pero al pasar los años y al desarrollarse nuevas técnicas su objetivo principal se vuelve una cuestión de tiempo.

A lo largo de este libro se trata de desarrollar un curso teórico sobre los conceptos, técnicas y algoritmos más importantes relacionados a la Inteligencia Artificial. Este libro es una introducción al mundo de la Inteligencia Artificial.

En este libro se desarrollan técnicas y algoritmos de las tres ramas principales de la Inteligencia Artificial. Los temas centrales incluyen algoritmos genéticos, redes neuronales, la web semántica y agentes inteligentes.

4

Inteligencia Artificial

Figura 34 – Páginas 5-8 Libro “Inteligencia Artificial”



Este libro va dirigido a aquellas personas que necesitan un enfoque práctico a la aplicación de algoritmos utilizados en inteligencia artificial. Se busca al final unificar las diferentes ramas y disciplinas descritas en la clasificación de la inteligencia artificial del primer capítulo.

Prerrequisitos

Se debe tener un conocimiento básico sobre estructuras de datos, bases de datos, estadística. Nociones básicas de matemática como derivación y poder interpretar funciones.

CONTENIDO

1. Capítulo 1 – Historia de la Inteligencia Artificial	11
A. Alcance	12
1.1. La base estadística	12
1.2. Inteligencia Artificial	13
1.3. Historia	14

B. Artículos relacionados	16
2. Capítulo 2 – Algoritmos Genéticos	18
A. Alcance	19
2.1. Definición	19
2.2. Iniciando la población	21
2.3. Crossover	23
2.3.1 Crossover de n puntos	23
2.3.2 Crossover uniforme	24
2.4. Mutación	24
2.5. Condición de finalización	25
B. Artículos relacionados	26
3. Capítulo 3 – La web semántica	27
A. Alcance	28
3.1. La web actualmente	28
3.2. La web semántica	29
3.3. Ontologías	33
B. Artículos relacionados	34
4. Capítulo 4 – Machine Learning	36
A. Alcance	37
4.1. Definición de Machine Learning	37
4.1.1. Razonamiento Inductivo	38
4.1.2. Razonamiento Deductivo	39
4.2. Árboles de decisión	39
4.2.1. Árboles binarios	41
4.2.2. Reducción de Incerteza	44
4.2.3. Árboles con valores no binarios	48



5

Inteligencia Artificial



6

Inteligencia Artificial



4.2.4. Árboles multivariados	49
B. Artículos relacionados	52
5. Capítulo 5 – Redes Neuronales	53
A. Alcance	54
5.1. Definición	55
5.1.1. Conexionismo	55
5.1.2. Neuronas Orgánicas	55
5.2. Redes Neuronales	58
5.2.1. Perceptrón	61
5.2.1.1 Aprendizaje y Generalización	64
5.2.1.2 Aplicando aprendizaje supervisado	66
5.3. Función de activación	68
5.3.1. Función escalón	69
5.3.2. Función límite lineal	69
5.3.3. Función Sigmoideal	70
5.3.4. Función Gaussiana	71
5.4. Tipos de Redes Neuronales	72
5.4.1. Auto-asociativas	72
5.4.2. Hetero-asociativas	73
5.4.3. Arquitectura Unidireccional	74
5.4.4. Arquitectura recursiva o de retroalimentación	75
5.5. MLP Multi Layer Perceptron	75
5.5.1. Algoritmo de propagación Inversa	79
5.5.2. Ejemplo de MLP	80
5.5.3. Teorema de Kolmogorov	86
B. Artículos relacionados	88

6. Capítulo 6 – Agentes	90
A. Alcance	91
6.1. Agentes	92
6.1.1. Agentes de Software	95
6.2. Características de los agentes	98
6.2.1. Autonomía	98
6.2.2. Inteligencia	99
6.2.3. Pro actividad	100
6.2.4. Reactividad	100
6.2.5. Habilidad de aprender	101
6.2.6. Cooperación	102
6.2.7. Movilidad	103
6.3. Clasificación de Agentes	104
6.3.1. Agentes Reactivos	104
6.3.1.1 Agentes orientados a una meta	105
6.3.1.2 Agentes basados en la utilidad	105
6.3.2. Agentes de interfaz	106
6.3.3. Agentes móviles	107
6.3.4. Agentes de información	107
6.4. Sistema Multi Agentes	108
6.4.1. Comunicación	109
6.5. Clasificación de las propiedades del medio ambiente	110
6.5.1. Accesibles e Inaccesibles	110
6.5.2. Determinista y no Determinista	111
6.5.3. Episódico y no Episódico	111
6.5.4. Estático y Dinámico	112



7

Inteligencia Artificial



8

Inteligencia Artificial

Figura 35 – Páginas 9-12 Libro “Inteligencia Artificial”

6.5.5. Discreto y Continuo	112	Figura No. 12 – Modelo de una neurona	56
6.6. Arquitecturas	112	Figura No. 13 – Modelo de una neurona artificial	59
6.6.1. Arquitectura basada en la lógica	113	Figura No. 14 – División Lineal	62
6.6.2. Arquitecturas reactivas	114	Figura No. 15 – Perceptrón	63
6.6.3. Arquitectura BDI	116	Figura No. 16 – División lineal con tres variables	64
B. Artículos relacionados:	120	Figura No. 17 – Generalización	68
I. Glosario	121	Figura No. 18 – Función escalón	69
II. Referencias Bibliográficas	123	Figura No. 19 – Función Límite Lineal	70
III. Apéndice A: Licencia	125	Figura No. 20 – Función sigmoideal	71
		Figura No. 21 – Función Gausiana	72
		Figura No. 22 – red auto asociativa	73
		Figura No. 23 – Red hetero asociativa	74
		Figura No. 24 – MLP	76
		Figura No. 25 – Neuronas Ij	78
		Figura No. 26 – Tabla de verdad 3	81
		Figura No. 27 – Ejemplo XOR antes de balance	82
		Figura No. 28 – Ejemplo XOR balanceado	85
		Figura No. 29 – Teorema de Kolmogorov	86
		Figura No. 30 – Comunicación	110
		Figura No. 31 – Arquitectura BDI	118

ILUSTRACIONES

I. Figuras

Figura No. 1 - Algoritmo Genético	21
Figura No. 2 – Web semántica	31
Figura No. 3 – Arquitectura de la Web Semántica	33
Figura No. 4 – Árbol Binario	40
Figura No. 5 – Tabla de verdad 1	42
Figura No. 6 – Árbol Binario para DNF	43
Figura No. 7 – Teorema de Kolmogorov	44
Figura No. 8 – Tabla de verdad 2	45
Figura No. 9 – Árbol Binario resuelto	48
Figura No. 10 – Grafo de decisión	50
Figura No. 11 – Árbol multivariado	51



9

Inteligencia Artificial



10

Inteligencia Artificial

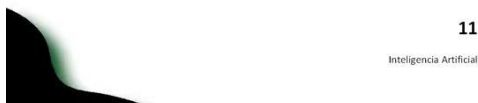
ALCANCE

1. Historia de la Inteligencia Artificial
 - 1.1. La base estadística
 - 1.2. Inteligencia Artificial
 - 1.3. Historia

Capítulo 1

HISTORIA DE LA INTELIGENCIA ARTIFICIAL

Este capítulo nos da una introducción a la historia de la inteligencia artificial desde su base estadística. Luego se desarrolla la rama en si como una ciencia establecida hasta llegar a la clasificación actual de Inteligencia Artificial.



11

Inteligencia Artificial

1.1 La base estadística

Comenzamos con una breve historia de la estadística ya que es una parte fundamental en los procesos desarrollados por la inteligencia artificial.

La estadística nace hace mucho tiempo y comenzó con trabajos cuyo objetivo era lograr contar objetos como habitantes o el número de frutas producidas anualmente. Surge en imperios como China y Egipto. La estadística nace con la finalidad de recolectar información económica y demográfica para beneficio del gobierno.



12

Inteligencia Artificial

Figura 36 – Páginas 13-16 Libro “Inteligencia Artificial”

La estadística fue evolucionando y luego surge la estadística matemática y la teoría del muestreo. Al final del siglo XVIII se realizan los primeros trabajos sobre análisis de regresión. Alrededor de 1930 surge el análisis de datos multi variante. Esta técnica descriptiva tenía como objetivo mostrar relaciones entre diferentes variables.

En 1936 se presenta el primer trabajo sobre análisis discriminatorio, aquí se tiene una variable de respuesta que nos dice la clase de cada objeto, se encuentra aquí la mejor combinación lineal para distinguir las diferentes clases. Este problema de clasificación de clases es llamado clustering.

1.2 Inteligencia Artificial

La inteligencia artificial es una ciencia relativamente joven, podemos considerar el nacimiento de la IA como una disciplina formal desde 1956 cuando McCarthy acuña el término “Inteligencia Artificial”. La ciencia de la inteligencia artificial tiene como objetivo central lograr que máquinas o computadoras sean capaces de realizar actividades que requieren las habilidades mentales de los seres humanos.

El área de la inteligencia artificial se enfoca principalmente en actividades que son muy complicadas para una computadora. Un ejemplo clásico es el análisis de imágenes, reconocer números escritos a mano por una persona es algo extremadamente fácil para los seres humanos pero muy complicado para una computadora.

13

Inteligencia Artificial

Otro ejemplo es el análisis del lenguaje natural, poder analizar lo que una persona comunica verbalmente es una actividad complicada para una computadora. La inteligencia artificial desarrolla técnicas y algoritmos para resolver este tipo de problemas.

1.3 Historia

Las bases de la inteligencia artificial son el paradigma de Von Neumann y las técnicas de computación secuencial, se apoyaba también fuertemente de las matemáticas y la lógica. Luego esta disciplina se apoyo en la teoría simbólica propuesta por Newell y Simon en 1956. Luego en 1961 Minsky divide la inteligencia artificial en 5 temas clave: búsqueda, reconocimiento de patrones, aprendizaje, planeación e inducción; enfocándose principalmente en técnicas de búsqueda.

Algunos años después surgen nuevas técnicas como: sistemas expertos, representación de conocimiento, machine learning y el procesamiento del lenguaje natural. En los años 70s surge la inteligencia artificial Micro Distribuida de donde surgen las redes neuronales y el procesamiento de lenguaje natural.

Las redes neuronales artificiales son un modelo que se creó basado en la manera en que funcionan las redes neuronales orgánicas, imitando su funcionamiento. Surgen en 1943 siendo descritas por McCulloch y Pitts. Por medio de la metodología de redes neuronales se tenía un modelo basado en el aprendizaje, a

14

Inteligencia Artificial

partir del aprendizaje se llega a tener conocimiento y a partir de este conocimiento se llega a la generalización. El modelo inicial para las redes neuronales fue el perceptrón presentado por Minsky y Papert en 1969, tenía como objetivo principal resolver el problema de analizar imágenes.

Años después se descubrió que había una gran limitación con la utilización del perceptrón para resolver problemas, únicamente podía resolver problemas linealmente separables. A raíz de esto luego de un breve periodo de inactividad surgen los perceptrones multi capa y su nuevo método de aprendizaje el algoritmo de propagación inversa. Por medio de esta nueva estructura se pudo resolver la limitación impuesto en el perceptrón simple. Luego de esto se dedica más estudio hacia las redes neuronales recursivas.

Las otras dos ramas principales de la inteligencia artificial surgen alrededor de la misma fecha que la IA micro distribuida, surge la Inteligencia Artificial evolutiva y la Inteligencia artificial macro distribuida.

La IA evolutiva da lugar al surgimiento de los algoritmos genéticos que se basan en el modelo evolutivo. Por medio de una búsqueda heurística paralela es posible encontrar una solución entre grandes espacios de posibles respuestas, esto basado en un sistema de castigo y recompensa. La inteligencia artificial macro distribuida da origen a los sistemas multi agentes y a otras técnicas. La introducción de los agentes específicamente los sistemas multi

agentes dan origen a robots reactivos una de las áreas más importantes de la inteligencia artificial.

Artículos relacionados

- **Timeline: A brief history of artificial intelligence**
Bruce G. Buchanan, University of Pittsburgh
Ref: <http://www.aaai.org/ITopics/pmwiki/pmwiki.php/ITopics/BriefHistory>
- **History of AI**
USA Today Staff
Ref: <http://www.usatoday.com/tech/news/2001-06-20-ai-history.htm>
- **Can robots be polite?**
Michael Schirber, Live Science
Ref: http://www.livescience.com/technology/041025_interactive_Robots.html

15

Inteligencia Artificial

16

Inteligencia Artificial

Figura 37 – Páginas 17-20 Libro “Inteligencia Artificial”

- *Timeline: real robots*
BBC Staff Writer.
Ref:
http://news.bbc.co.uk/1/hi/in_depth/sci_tech/2001/artificial_intelligence/1531432.stm

Capítulo 2

ALGORITMOS GENÉTICOS

Este capítulo nos da una introducción al algoritmo que simboliza la rama de la Inteligencia Artificial Evolutiva. Describimos el algoritmo en cada etapa introduciendo nuevos conceptos propios del mismo algoritmo. Se dan ejemplos sencillos para algunos incisos para ejemplificar de mejor manera los conceptos.

17

Inteligencia Artificial

18

Inteligencia Artificial

ALCANCE

2. Algoritmos Genéticos
 - 2.1. Definición
 - 2.2. Iniciando la población
 - 2.3. Crossover
 - 2.4. Mutación
 - 2.5. Condición de finalización

2.1 Definición

Los algoritmos genéticos se basan en la teoría evolutiva y en la genética para resolver problemas. Por medio de los algoritmos genéticos es posible encontrar una convergencia rápida hacia un problema teniendo un espacio grande de posibles respuestas. Aquí se trata cada posible respuesta como un individuo, cada individuo está representado por una cadena de genes. El algoritmo comienza con un conjunto de individuos llamado población y sobre esta

población se simula el proceso de evolución durante varias generaciones.

Al igual que el modelo evolutivo los cromosomas de los individuos cambian por medio del apareamiento de dos individuos de la población creando así un nuevo individuo, los cromosomas son heredados de ambos padres. Este proceso se repite en cada generación creando una nueva población con la cual trabajar. Los algoritmos genéticos son un proceso estocástico y son capaces de abarcar grandes espacios de posibles respuestas a la vez, por lo que se llega a una respuesta más rápido.

Existe una serie de pasos que debe seguir un algoritmo genético, esta es una grafica del algoritmo, vamos a describir cada punto.

19

Inteligencia Artificial

20

Inteligencia Artificial

Figura 38 – Páginas 21-24 Libro “Inteligencia Artificial”

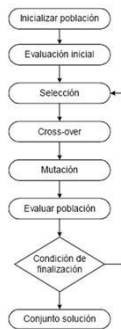


Figura No. 1 – Algoritmo Genético
Fuente: Algoritmos Genéticos por Samuel Orozco

2.2 Iniciando la población

Como habíamos explicado antes el algoritmo genético está basado en una población que cambia constantemente. El tamaño de la población inicial debe ser una consideración importante en

este punto, ya que debe ser lo suficientemente grande para poder garantizar la diversidad de las soluciones.

La manera en la cual representamos las posibles respuestas debe ser reducida a una cadena binaria, este es el primer paso que debemos dar antes de comenzar el algoritmo. Debemos crear esta cadena pensando de antemano en qué manera vamos a poder evaluar su rendimiento respecto a la respuesta.

Debemos determinar también cual es la condición de terminación, tenemos dos opciones. Podemos optar por terminar la ejecución al encontrar una solución que llegue a cierto rango de exactitud, o simplemente terminar la ejecución después de un número determinado de generaciones.

Una vez que tenemos definidos estos puntos podemos seguir con la fase de la primera evaluación. Esto se realiza por medio de la función fitness, esta función tiene como objetivo dar un valor numérico a cada individuo de la población. El valor numérico asociado a cada individuo indica el grado de exactitud que tiene hacia la respuesta que buscamos.

Luego de esta etapa debemos escoger un método de selección, lo que estamos haciendo aquí es determinar que individuos van a sobrevivir y que individuos serán eliminados. El tamaño de la población puede ser estático o variable nosotros definimos estos parámetros. La única manera de distinguir entre los diferentes individuos es por medio del valor numérico asignado por la función fitness, por lo tanto es común utilizar un algoritmo de selección que conserve los individuos con valores más altos.

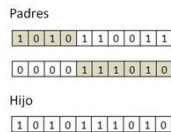


2.3 Crossover

Luego de la etapa de selección ya tenemos una nueva población con la cual trabajar, en este momento se debe realizar la etapa de apareamiento o crossover. Se le llama crossover ya que estamos cruzando los cromosomas de dos individuos diferentes para generar uno nuevo. Existen muchos métodos para realizar la función de crossover vamos a ver algunos ejemplos:

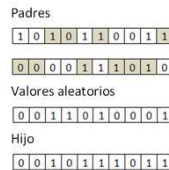
2.3.1 Crossover de n puntos

Dividimos un número determinado de cromosomas para ambos padres y luego se mezclan los bloques de cromosomas para generar al hijo.



2.3.2 Crossover uniforme

Se genera una cadena de genes aleatoria y luego se cruzan los genes de ambos padres a partir de estos valores aleatorios. Podríamos asignar cruzar con el primer padre si el valor es 1 y con el segundo padre si el valor es 0.



Existen otros patrones para realizar la operación de crossover pero estos dos son los más comunes.

2.4 Mutación

En los modelos evolutivos reales existe la mutación, esta se da aproximadamente cada 1000 generaciones y se puede considerar como un cambio aleatorio en los genes, es decir el cromosoma que cambiamos no es heredado de ninguno de los padres. A nuestro



Figura 39 – Páginas 25-28 Libro “Inteligencia Artificial”

algoritmo genético debemos aplicar la mutación aproximadamente con la misma frecuencia cada 1000 generaciones. Para realizar este proceso simplemente escogemos un cromosoma al azar y lo cambiamos. Este proceso ayuda a preservar la diversidad en la población pero es importante no abusar de la mutación ya que reduce el algoritmo genético a una búsqueda aleatoria.

Luego de haber realizado los procesos de crossover y de mutación debemos realizar nuevamente el proceso de evaluación. Debemos utilizar la función fitness para asignar valores numéricos a la nueva población que surge del crossover y de la mutación. Una vez que todos los individuos tienen un valor numérico asignado podemos seguir a la última fase del algoritmo.

2.5 Condición de finalización

Luego de haber evaluado nuevamente la población debemos revisar las condiciones de finalización. Dependiendo de las condiciones que hemos impuesto debemos revisar las variables y tomar una decisión. Si se ha cumplido con las condiciones podemos dar por terminado el proceso completo. Si no se han cumplido las condiciones todavía vamos a regresar al proceso de selección inicial del algoritmo y repetir el proceso completo, cada ciclo viajado a través del algoritmo es denominado una generación. El proceso por lo tanto se repite hasta que se cumpla con las condiciones de finalización y podamos mostrar una respuesta.

Artículos Relacionados

- *Use of Genetic Algorithms in Optimization of Irrigation Pumping Stations*
S. I. Rodin, M. Moradi-Jalal
Ref: http://stullia.t.k.ru/irrpump/ir_pump.htm
- *Genetic Algorithm*
Todd Rowland
Ref: <http://mathworld.wolfram.com/GeneticAlgorithm.html>
- *Applications of Genetic Algorithm for Solving Multi-Objective Optimization Problems in Chemical Engineering*
Abhijit Tarafder, Ajay K. Ray and Santosh K. Gupta
Ref: <http://www.itk.ac.in/directors/directsept04/santosh-newww.pdf>
- *Genetic Algorithms - Nature's Way*
Chris Lucas
Ref: <http://www.calresco.org/genetic.htm>



25

Inteligencia Artificial



26

Inteligencia Artificial



ALCANCE

3. La Web Semántica
 - 3.1. La web actualmente
 - 3.2. La web semántica
 - 3.3. Ontologías

Capítulo 3

LA WEB SEMÁNTICA

En este capítulo se desarrolla el concepto de la Web Semántica y se exponen los problemas de la web actual. Se presentan las diferentes capas que conforman esta nueva estructuración de la web y se cierra con el concepto de ontologías.



27

Inteligencia Artificial

3.1 La web actualmente:

Actualmente en internet existen problemas para encontrar la información que buscamos de manera eficiente, en este principio se basa este capítulo. Lo que se utiliza en estos días para buscar la información que necesitamos son los motores de búsqueda que proveen páginas como Yahoo y Google. Estos motores de búsqueda están basados en palabras clave dentro de una página web, y en algunos casos como Google el orden de las coincidencias de una búsqueda se ordena por el número de hits que tiene la página web.



28

Inteligencia Artificial

Figura 40 – Páginas 29-32 Libro “Inteligencia Artificial”

Por ejemplo nosotros estamos buscando información sobre la Universidad de San Carlos así que en una página como Google ponemos la palabra clave “San Carlos”, el tipo de información que podemos llegar a obtener es: noticias de prensa, universidades, hoteles, museos, asociaciones, trabajos, publicaciones, etc. Esto causa que una búsqueda con pocas palabras clave genere miles de páginas posibles para lo que estamos buscando. En general una simple labor de búsqueda se puede convertir en una tarea complicada, esto es causado por la manera en que están creados estos motores de búsqueda.

Por que se recurre a este medio para realizar búsquedas? Porque en general no existe un formato que se pueda seguir para clasificar páginas. Este problema hace difícil la tarea de diferenciar páginas relacionadas a noticias, hospitales, museos, etc.

Este problema se ha tratado de resolver mediante el uso de agentes inteligentes, pero incluso ellos llegan a tener limitaciones similares. Un agente puede llegar a facilitar la labor de búsqueda pero únicamente hasta cierto grado. Siempre se tiene el problema del formato y la clasificación de la información.

3.2 La web semántica

La web semántica es un concepto desarrollado por Tim Berners-Lee, vamos a ver dos definiciones del autor mismo.

“El primer paso es colocar los datos en la web de un modo en que las maquinas puedan entenderlos naturalmente o convertirlos

a esa forma. Esto crea lo que yo llamo una web semántica: una red de datos que pueden ser procesados directa o indirectamente por maquinas”

“La web semántica es una extensión de la actual web en la cual la información se da mediante un significado bien definido, lo que facilita que los ordenadores y la gente trabajen en cooperación”

Lo que la web semántica busca es darle un formato comprensible a las maquinas para automatizar todos los procesos de búsqueda de información. De llegar a este punto se podrían tener agentes capaces de realizar compras en línea buscando el mejor precio, agentes que logren encontrar información sobre una persona, agentes que encuentren el mejor apartamento libre para nuestra situación, etc.

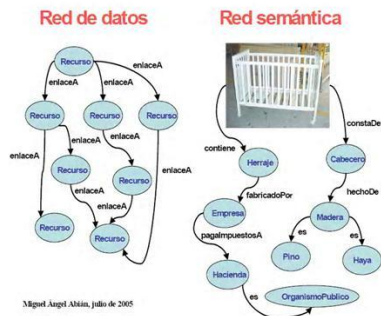
Pero la web semántica va mas allá de simplemente darle un formato a las páginas web, también busca darle una estructura a la información. El verdadero poder que se llega a alcanzar proviene de las relaciones que se pueden establecer con la información, este concepto se explica mejor con la siguiente imagen:

29

Inteligencia Artificial

30

Inteligencia Artificial



Miguel Ángel Abian, julio de 2005

Figura No. 2 – Web semántica

Fuente: El futuro de la web por Miguel Ángel Abian

Actualmente el formato más utilizado para intercambiar información es XML. Este formato está basado en los formatos UNICODE y URI. UNICODE es un formato que utiliza números enteros positivos para representar letras. Mediante URI (Unique Resource Identifier) se identifica de manera única cada recurso en la web semántica.

El formato RDF (Resource Description Framework) es un lenguaje que puede ser considerado como una extensión de XML. Permite incluir semántica en su descripción de recursos algo que no se puede hacer directamente con XML.

Por medio de las ontologías es posible leer el formato RDF y crear las relaciones entre la información, este inciso se describe en detalle más adelante.

Usando una ontología se puede llegar a realizar conclusiones lógicas sobre la información. Por medio de conclusiones lógicas podemos llegar a probar la certeza de la información. Y una vez que se tienen pruebas sobre la información se llega a un nivel de confianza, convirtiendo así la información en conocimiento.

31

Inteligencia Artificial

32

Inteligencia Artificial

Figura 41 – Páginas 33-36 Libro “Inteligencia Artificial”

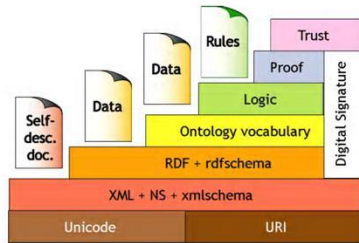


Figura No. 3 – Arquitectura de la Web Semántica
Fuente: El futuro de la web por Miguel Angel Abian

3.3 Ontologías

Este concepto es utilizado dentro del ámbito de la inteligencia artificial y la representación del conocimiento. En una ontología se trabaja con un dominio, sobre este dominio se crea un esquema para establecer una jerarquía entre los conceptos del dominio.

Al crear una clasificación exhaustiva de los conceptos de un dominio se establece la jerarquía entre estos, en una ontología también se manejan conceptos como sinónimos y relaciones para los conceptos del dominio.

Existen varias herramientas para modelar ontologías. Tenemos herramientas open source como Protegé y herramientas pagadas como OntoEdit. Todas las herramientas de edición de ontologías permiten generar una salida de la ontología construida usualmente en formato RDF, para poder ser cargada por una aplicación posteriormente si es necesario.

Artículos Relacionados

- *Making sense of the "semantic web"*
Steve Mollman
Ref: http://www.cnn.com/2008/TECH/12/17/db.semanticweb/index.html?section=cnn_latest
- *Semantic Web Patterns: a guide to semantic technologies*
Alex Iskold
Ref: http://www.readwriteweb.com/archives/semantic_web_patterns.php

- *Semantic Web plays key role in Net's future*
Jennifer Zaino
Ref: <http://www.semanticweb.com/article.php/3792006>
- *The Semantic Web: A Primer*
Edd Dumbill
Ref: <http://www.xml.com/pub/a/2000/11/01/semanticweb>

Capítulo 4

MACHINE LEARNING

En este capítulo se introduce el tema de machine learning, se explica también la manera en que se interpreta el conocimiento. Luego se introduce a la primera herramienta de machine learning, los árboles de decisión.

Figura 42 – Páginas 37-40 Libro “Inteligencia Artificial”

ALCANCE

- 4. Machine Learning
 - 4.1. Definición de Machine Learning
 - 4.1.1. Razonamiento Inductivo
 - 4.1.2. Razonamiento Deductivo
 - 4.2. Árboles de decisión
 - 4.2.1. Árboles binarios
 - 4.2.2. Reducción de Incerteza
 - 4.2.3. Árboles con valores no binarios
 - 4.2.4. Árboles multivariados

4.1 Definición de Machine Learning

Esta es una rama de la inteligencia artificial que estudia el desarrollo de algoritmos que permiten que una máquina logre realizar el proceso de aprendizaje.

37

Inteligencia Artificial

Para dar una definición más exacta de cómo es que una máquina aprende mostramos la siguiente afirmación: una máquina ha aprendido una vez que cambia su propia estructura de funcionamiento o los datos que utiliza, de manera que mejora su rendimiento a partir de ejemplos provistos con fines de aprendizaje.

Es por medio de algoritmos que podemos procesar información de entrada y utilizar esta información para mejorar la manera en que funciona un programa.

Existen problemas que no se pueden resolver programándolos directamente como por ejemplo el reconocimiento de voz o el reconocimiento de la letra escrita a mano. Esto se da porque las variables con las cuales trata son muchas o cambian constantemente, es en este tipo de problemas que resulta más factible usar técnicas de machine learning.

Existen dos procesos de aprendizaje basados en dos tipos diferentes de razonamiento:

4.1.1 Razonamiento Inductivo

En el razonamiento inductivo utilizamos una serie de premisas para hacer generalizaciones sobre las conclusiones a las cuales llegamos. En otras palabras basándonos en la información que tenemos asumimos algunos hechos para llegar a nuestras conclusiones.

38

Inteligencia Artificial

4.1.2 Razonamiento deductivo

Este tipo de razonamiento está basado en la lógica, aquí todas las conclusiones a las cuales se llegan están basadas en hechos o información obtenida, comparándolo con el razonamiento inductivo no se llega a asumir nada ya que todo está apoyado en una afirmación.

Existe una serie de herramientas utilizadas en la rama de machine learning de las cuales vamos a entrar en detalle con dos, la primera de estas herramientas son los árboles de decisión.

4.2 Árboles de decisión

Podemos definir un árbol de decisión como un conjunto de nodos conectados entre sí, dichos nodos se dividen en más nodos haciendo crecer el número de ramas por nivel, y en donde las ramas internas son divisiones y las ramas terminales o externas son categorías de clasificación.

En un árbol de decisión podemos tener n cantidad de variables con las cuales debemos de tratar y pueden estar divididas en cualquier número de categorías, aquí mostramos un ejemplo sencillo de un árbol de decisión:

39

Inteligencia Artificial

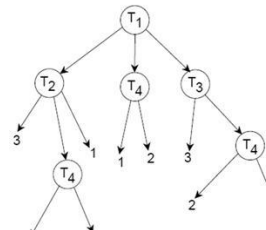


Figura No. 4 – Árbol Binario
Fuente: Introduction to Machine Learning por Nils J. Nilsson

Aquí tenemos cuatro variables a ser evaluadas y tres posibles clasificaciones. Partimos a partir del nodo que se encuentra más arriba del nodo raíz, luego dependiendo del valor obtenido luego de procesar el valor de T1 podemos tener tres posibles resultados. Este proceso se repite hasta que se llegue a un nodo terminal donde finalmente se le asigna a una clasificación.

Los valores que se pueden utilizar en los árboles de decisión no tienen un límite realmente, se puede usar valores binarios,

40

Inteligencia Artificial

Figura 43 – Páginas 41-44 Libro “Inteligencia Artificial”

valores reales, letras, etc. Si el árbol binario que estamos usando utiliza únicamente valores booleanos podemos decir que estamos usando un árbol booleano.

En los nodos internos, donde ocurre la toma de decisiones, es posible evaluar una o varias variables a la vez. Si el árbol de decisión analiza varias variables al mismo tiempo se le llama multivariante, si solamente analiza una variable a la vez se le llama univariante. Las salidas de los nodos pueden dividirse en dos o más posibles salidas, si nuestro árbol de decisión únicamente tiene dos salidas para todos los nodos lo llamamos un árbol binario.

4.2.1 Árboles binarios

Cuando tratamos con árboles binarios la manera más fácil de armar un árbol de decisión es por medio de su función DNF. La función DNF (Forma normal disyuntiva) representa una función booleana como una suma de términos disyuntivos. Una vez que tenemos las variables que forman la respuesta podemos incluso obviar ciertas variables que no forman parte de su función DNF. Vamos a ver el siguiente ejemplo:

Vamos a ver el siguiente ejemplo:

X1	X2	X3	X4	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Figura No. 5 – Tabla de verdad 1

Desde el DNF de la siguiente tabla es:

$$DNF = X1 \cdot X2 \cdot X3 + X1 \cdot X3$$

Con esta función es fácil comenzar a armar el árbol de decisión. Aquí podemos tomar dos caminos rápidos, el primero es tomar X1 como nodo raíz el otro es tomar X3. A X2 no lo consideramos como posible raíz ya que solo afecta a dos posibles respuestas. Ahora X4 ni siquiera se toma en cuenta para la respuesta final por lo que no debe aparecer en el árbol binario.

41

Inteligencia Artificial

42

Inteligencia Artificial

Vamos a optar por comenzar con la variable X1 y tenemos el siguiente árbol.

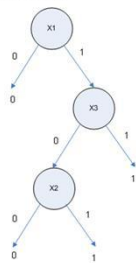


Figura No. 6 – Árbol Binario para DNF

Podemos viajar a través de las ramas y ver que para los dos resultados para 1 obtenemos los dos componentes de la función DNF. Si hubiéramos optado por comenzar con la variable X3 habríamos tenido un nodo extra. Este es el único inconveniente de utilizar la función DNF únicamente se toma como un punto de referencia. Pero existe un método estadístico que reduce el porcentaje de error en mayor grado.

4.2.2 Reducción de incerteza

Vamos a comenzar estudiando los árboles de decisión binarios ya que tiene la función de evaluación más fácil. Si el valor es 0 nos vamos hacia la izquierda y si es 1 hacia la derecha. Evaluar la función en cada nodo es fácil pero no es tan fácil saber en qué orden debemos de establecer los nodos para optimizar el árbol. La manera en que equilibramos el árbol es colocando los nodos que dividen en mayor grado los resultados en niveles superiores.

Para encontrar estos nodos utilizamos el método de reducción de incerteza, aquí usamos el principio de entropía. La entropía la definimos como el grado de aleatoriedad que tiene el sistema. Por lo tanto lo que queremos lograr es reducir al mínimo el grado de entropía luego de la primera evaluación en el primer nodo.

Esta es la fórmula con la cual se trabaja:

$$H(\Xi) = - \sum_i p(i|\Xi) \log_2 p(i|\Xi)$$

Figura No. 7 – Teorema de Kolmogorov

Fuente: Introduction to Machine Learning por Nils J. Nilsson

Donde identificamos con Ξ al grupo de patrones que tenemos para nuestro criterio de clasificación. Identificamos con i como las diferentes clasificaciones o resultados posibles que podemos lograr.

43

Inteligencia Artificial

44

Inteligencia Artificial

Figura 44 – Páginas 45-48 Libro “Inteligencia Artificial”

De esta manera calculamos la entropía para un elemento, para encontrar la reducción de entropía al escoger una de las variables como nodo inicial debemos de calcular la entropía para el sistema completo y luego la entropía para cada una de las variables.

Ejemplo de uso de entropía:

Vamos a utilizar el siguiente ejemplo para mostrar el cálculo de la entropía:

X1	X2	X3	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Figura No. 8 – Tabla de verdad 2

Vamos a comenzar calculando la entropía para el sistema completo:

Tenemos que Ξ es igual a 8 ya que existen 8 elementos para este ejemplo. Tenemos dos posibles respuestas que son 0 y 1, por lo tanto la sumatoria de H va a ser conformada por 2 elementos.

$$H = -(6/8)\log_2(6/8) - (2/8)\log_2(2/8) = 0.81$$

Usamos para i los valores de 6 y 2 porque existen 6 salidas para $y=0$ y 2 salidas para $y=1$. Recordemos que Ξ siempre es igual a 8 en este ejemplo.

Ahora vamos a calcular la entropía para x_2 , para este caso debemos realizar dos cálculos uno para cada rama. Vamos a comenzar con la rama izquierda con valor 0. Cuando x_2 es 0 únicamente tenemos para $y=0$, por lo tanto tenemos para $i = 4$ y 0.

$$H_{x_2 0} = -(4/4)\log_2(4/4) - (0/4)\log_2(0/4) = 0$$

Ahora seguimos con la rama derecha con valor 1. Cuando $x_2 = 1$ tenemos la mitad de resultados con $y=1$ y la otra mitad con $y=0$, por esta razón tenemos para $i = 2$ y 2.

$$H_{x_2 1} = -(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1$$

Ya con las dos ramas calculadas encontramos el promedio de la entropía de x_1 usando las dos ramas.

$$H_{x_2} = 0.5 * H_{x_2 0} + 0.5 * H_{x_2 1} = 0.5 * 0 + 0.5 * 1 = 0.5$$

La entropía para x_2 es igual a 0.5, ahora para calcular la reducción de entropía usando x_2 como nodo inicial debemos restar a la entropía del sistema completo la entropía de x_2 .

$$H - H_{x_2} = 0.81 - 0.5 = 0.31$$

Vamos a calcular la entropía para X_1 :

$$H_{x_1 0} = -(3/4)\log_2(3/4) - (1/4)\log_2(1/4) = 0.81$$

$$H_{x_1 1} = -(3/4)\log_2(3/4) - (1/4)\log_2(1/4) = 0.81$$

$$H_{x_1} = 0.5 * 0.81 + 0.5 * 0.81 = 0.81$$

45

Inteligencia Artificial

46

Inteligencia Artificial

Y la reducción de entropía para X_1 es:

$$H - H_{x_1} = 0.81 - 0.81 = 0.$$

Si utilizamos X_1 como nodo inicial la reducción de entropía es igual a 0, por lo tanto descartamos esta variable.

Vamos a calcular la entropía para X_3 :

$$H_{x_3 0} = -(4/4)\log_2(4/4) - (0/4)\log_2(0/4) = 0$$

$$H_{x_3 1} = -(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1$$

$$H_{x_3} = 0.5 * 0 + 0.5 * 1 = 0.5$$

Y la reducción de entropía para X_3 es:

$$H - H_{x_3} = 0.81 - 0.5 = 0.31$$

Tenemos que la reducción de entropía para X_2 y X_3 es la misma, por lo tanto comenzando con cualquiera de estas dos variables logramos optimizar el árbol. Podemos ver que la variable x_1 no era útil para formar este árbol. Al final el árbol queda así:

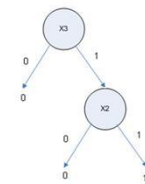


Figura No. 9 – Árbol Binario resuelto

Este algoritmo decrece en efectividad mientras más salidas tengan los árboles, por la naturaleza de la entropía, se debe tomar en cuenta para nodos con muchas salidas.

4.2.3 Árboles con valores no binarios

La función de evaluación utilizada en los nodos de árboles binarios es sencilla, pero cuando estamos utilizando valores reales o letras (las letras representan en si valores categóricos) debemos saber qué tipo de evaluación vamos a realizar en el nodo. Para este tipo de datos debemos crear rangos de valores conforme al número de salidas que se necesitan. Para los casos en que no está bien definido este rango de valores podemos aplicar también el principio de incerteza para encontrar los valores óptimos para los nodos del

47

Inteligencia Artificial

48

Inteligencia Artificial

Figura 45 – Páginas 49-52 Libro “Inteligencia Artificial”

árbol. El problema aquí es que debemos realizar varias pruebas para encontrar los rangos con menor grado de incerteza.

4.2.4 Árboles multivariados

Los árboles que hemos visto hasta ahora han sido árboles univaluados, existen árboles de decisión en los cuales cada nodo puede evaluar más de una variable a la vez. Vamos a realizar un ejemplo para la función DNF siguiente:

$$DNF = X1X3 + X2X4$$

En la imagen que vamos a ver a continuación podemos apreciar el problema que surge y la solución grafica que hemos propuesto.

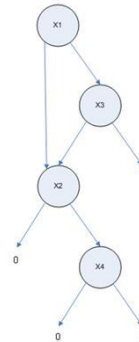


Figura No. 10 – Grafo de decisión

Por las respuestas que necesitamos para este árbol tenemos que hacer un "grafo de decisión" para evitar replicar la rama a partir de la variable X2. Este es un ejemplo del problema llamado sub árboles replicados, existen otras maneras para resolver esto y



no tener que usar un grafo de decisión (en este momento ya no se le puede llamar árbol técnicamente ya que dos de las salidas de dos nodos diferentes tienen el mismo destino), aquí podemos utilizar árboles multivariados. Por lo tanto vamos a evaluar dos variables por nodo formando el siguiente árbol de decisión.

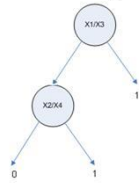


Figura No. 11 – Árbol multivariado

Si en el primer nodo los valores para x1 y x3 son igual a 1 tenemos un valor verdadero, de tener un 0 seguimos al siguiente nodo. En el segundo nodo hacemos la misma evaluación para las variables X2 y X4 teniendo un 1 o un 0.



Artículos Relacionados

- **Building decision trees in Python**
Christopher Roach
Ref: http://www.onlamp.com/pub/a/python/2006/02/09/ai_decision_trees.html
- **Decision Trees**
Estelle Brand, Rob Gerritsen
Ref: <http://www.dbmsmag.com/9807m05.html>
- **Breeding decision trees using evolutionary techniques**
Athanasios Papagelis, Dimitris Kalles
Ref: <http://www.gatree.com/data/BreedinDecisionTreeUsinEvo.pdf>
- **Orthogonal Decision Trees**
Hilol Kargupta and Haimonti Dutta
Ref: <http://www.cs.umbc.edu/~hilol/IPS/PUBS/Papers/ICDM04.pdf>

Figura 46 – Páginas 53-56 Libro “Inteligencia Artificial”

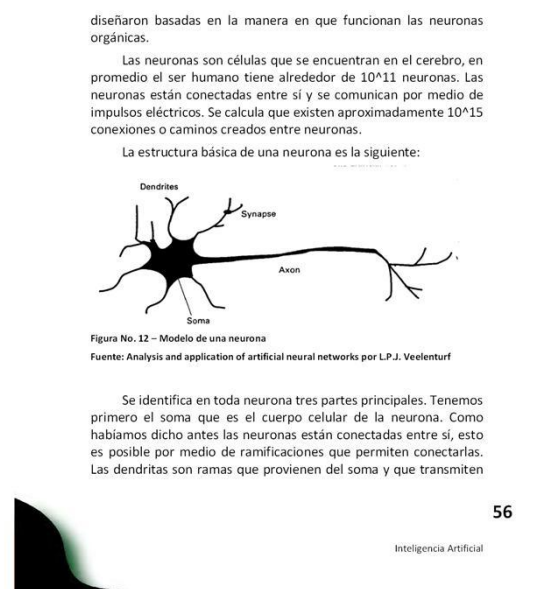
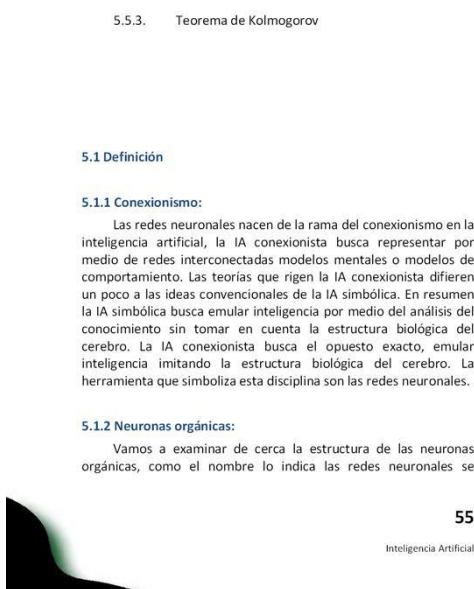
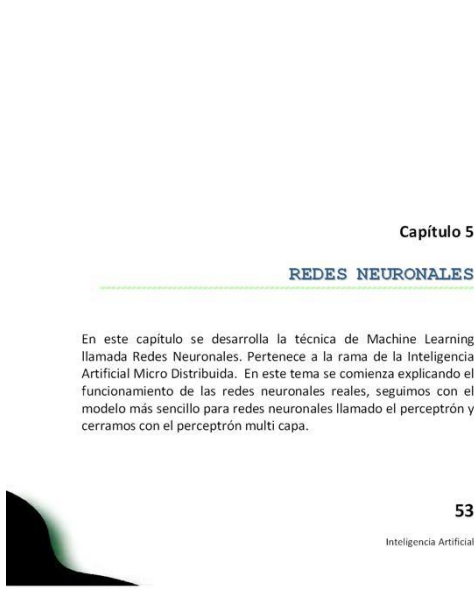


Figura 47 – Páginas 57-60 Libro “Inteligencia Artificial”

señales eléctricas hacia adentro de la neurona. El axón también es una rama que proviene del soma pero que transmite señales hacia afuera. Cada axón a su vez se divide en varias ramas para hacer contacto con otras neuronas. A los lugares donde se crean nuevas ramas o donde las ramas se tocan se le llama sinapsis, es en estos puntos donde la interacción entre neuronas ocurre.

Seguimos ahora con los estados en los cuales se encuentran las neuronas, las neuronas están encendidas o apagadas, en un estado de excitación o inhibición. Los estados de las neuronas son afectados por medio de las conexiones entre ellas, específicamente por medio de las sinapsis.

Las sinapsis son un punto crucial para nuestro estudio, el cerebro tiene la capacidad de aprender de experiencias pasadas y esto se traduce físicamente a mejores conexiones entre neuronas. Cuando una conexión es utilizada mejora, esto es causado porque las sinapsis crean sustancias químicas que mejoran la conexión entre las neuronas, de esta manera alterando la relación entre ellas. Este proceso hace posible que el cerebro reaccione de diferente manera ante alguna situación dependiendo de la información que tiene almacenada.

Para nuestro estudio existen dos funciones de las cuales es capaz el cerebro que nos interesan mucho, el cerebro es capaz de aprender y de generalizar. Por medio de la generalización es posible dar solución a problemas que no hemos visto antes basándonos en

la resolución de otros problemas similares. Vamos a ver este concepto en detalle más adelante.

5.2 Redes neuronales

Ahora que tenemos la base sobre la cual están creadas las redes neuronales podemos seguir. Vamos a analizar la neurona, una neurona es una unidad de procesamiento de información, al igual que el modelo visto para las neuronas orgánicas.

Una neurona posee un conjunto de entradas sobre las cuales debe tomar una decisión y transmitir dicho cálculo por medio de las salidas que también posee. Como en una red neuronal orgánica, la fuerza que adquiere una red neuronal proviene de las conexiones que logra crear con otras neuronas.

Esta es la estructura de una neurona:

57

Inteligencia Artificial

58

Inteligencia Artificial

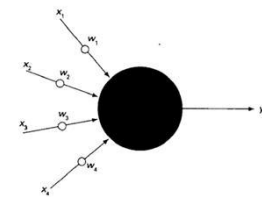


Figura No. 13 – Modelo de una neurona artificial
Fuente: Analysis and application of artificial neural networks por L.P.J. Veelenturf

Conexiones de entrada (inputs): Una neurona puede tener n número de entradas, las entradas de una neurona están ligadas al peso (o weight) que posee la conexión de donde provienen la información. Las neuronas tienen una entrada extra llamada bias que no está ligada a ningún weight y tiene un valor constante.

Función de entrada: es una función por medio de la cual se calcula el valor de entrada neto (net input signal). El modelo más sencillo es la sumatoria del valor de cada una de las entradas. Donde cada entrada está formada por la multiplicación de la entrada misma con su peso ($x_i * w_i$).

Señal de activación: tomando los valores de entrada netos y combinándolos con el estado actual de la neurona se calcula el nuevo nivel de activación de la neurona, aquí hablamos del estado de la neurona. En la mayoría de modelos la señal de activación y la función de salida es la misma.

Función de salida (output): en cada neurona existe también una función de salida, esta función transforma el estado actual de activación en una salida. Para calcular esto se utilizan las entradas netas de la neurona. Este valor es transmitido por el axón de la neurona hacia afuera, para nuestro diagrama el valor de y.

Si recordamos las sinapsis y la función que tienen en la relación entre las neuronas orgánicas es más fácil comprender el objetivo del peso o weight de la neurona artificial. Al igual que una sinapsis el weight de una neurona tiene como objetivo alterar la “calidad” de la relación que existe entre dos neuronas. Es por esta razón que para calcular la entrada neta de una neurona se multiplica la entrada el weight de la relación de cada una de las entradas para luego dársele a la función de entrada.

Los valores que puede manejar una red neuronal puede ser cualquier rango de números que necesita la red neuronal. Las estructuras más comunes son: valores binarios (0 y 1), bivalentes (1 y -1) y continuos (rango entre 0 y 1).

Ahora que sabemos la información básica sobre una neurona vamos a ver un ejemplo sencillo:

59

Inteligencia Artificial

60

Inteligencia Artificial

Figura 48 – Páginas 61-64 Libro “Inteligencia Artificial”

5.2.1 Perceptrón:

Es uno de los modelos de redes neuronales más comunes, se le llama de esta manera por la finalidad que tenía. El modelo originalmente buscaba investigar la manera en que se perciben las imágenes. La arquitectura para este modelo es unidireccional y realmente solo tiene dos capas. Las neuronas de este modelo utilizan una sumatoria sencilla para calcular las entradas netas y su función de activación es ya sea la función escalón o la función de limite lineal (las funciones de activación se detallan más adelante).

En la primera capa se tienen buffers para contener las entradas y en la segunda capa se tiene la neurona llamada perceptrón.

Lo que buscamos realizar con este ejemplo es simplemente clasificar un grupo de coordenadas. Tenemos dos clasificaciones para dichas coordenadas, de manera que dado un set de coordenadas debemos clasificarlo ya sea bajo el grupo A o B.

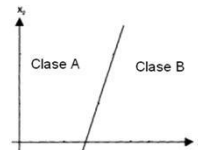


Figura No. 14 - División Lineal

Aquí representamos gráficamente lo que queremos. Para comenzar con el problema necesitamos saber cuántas entradas tenemos, para este caso tenemos las entradas (x_1, x_2) dos coordenadas. Aparte de esto necesitamos saber cuántas salidas tendrá el problema para saber interpretar la misma, si la salida regresa 0 pertenece a la clase A y si regresa 1 pertenece a la clase B.

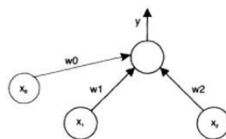


Figura No. 15 - Perceptrón

Fuente: Analysis and application of artificial neural networks por L.P.J. Veelenturf

Con estos datos modelamos nuestro perceptrón y obtenemos la gráfica que vemos arriba. Tenemos dos entradas x_1 y x_2 , también tenemos un bias representado por x_0 que siempre tendrá un valor fijo que es usualmente 1. Tenemos un perceptrón y por lo tanto una salida representada por y .

El bias en términos estadísticos nos permite tener un poco más de libertad en cuanto a la función que se crea a partir de las entradas, si pensamos en términos de funciones el bias nos permite moverla a través del eje x ajustando la misma a los datos mejorando la exactitud.

Pero el perceptrón tiene una gran limitación, únicamente puede resolver problemas que sean linealmente separables. El problema que acabamos de plantear es linealmente separable. Por



ejemplo si tuviéramos un problema con tres entradas se podría trazar un plano que dividiera en dos grupos las posibles respuestas.

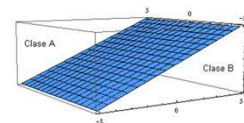


Figura No. 16 - División lineal con tres variables

A los perceptrón también se les llama perceptrons binarios ya que únicamente pueden ser usados para obtener dos valores de salida posibles o dos clasificaciones diferentes.

Para resolver problemas que no son linealmente separables debemos usar otro modelo como el perceptrón multicapa también conocido como MLP (Multi-layer perceptron).

5.2.1.1 Aprendizaje y generalización

Con este ejemplo podemos explicar más a fondo la finalidad de los weights en la red neuronal. Una red neuronal no almacena realmente los datos nuevos que obtiene, pero a pesar de esto si tiene un sistema de aprendizaje. La manera en la cual aprende es por medio de la actualización de sus weights. De esta manera

Figura 49 – Páginas 65-68 Libro “Inteligencia Artificial”

mientras más datos aprenda la red neuronal, más acertada será la respuesta a futuros valores ya que sus weights van a estar más equilibrados. Esta habilidad de aprender que posee la red neuronal es su característica más importante.

Al grupo de datos que se utilizan para que una red neuronal aprenda se le llama un Set de Entrenamiento, y cada uno de los elementos del set de entrenamiento se le llama Ejemplos de entrenamiento. Existen varios métodos por los cuales se puede realizar el proceso de aprendizaje pero se dividen en tres grandes grupos:

Aprendizaje supervisado: cada ejemplo de entrenamiento tiene su vector de entradas y junto a este su vector de salida. El proceso de aprendizaje consiste en hacer que la red neuronal haga las asociaciones partiendo del vector de entrada y luego viendo la salida que debe obtener. Durante este proceso la red neuronal actualiza sus pesos luego de cada asociación.

Aprendizaje no supervisado: aquí solo se tiene un vector de entradas para el entrenamiento. La red neuronal en base a todos los vectores de entrada encuentra características internas sobre los datos para luego actualizar su propia estructura.

Aprendizaje de refuerzo: aquí se realiza una mezcla entre el aprendizaje supervisado y el aprendizaje no supervisado. Se tiene un Set de entrenamiento y se revisa la salida para cada uno de los elementos. Si la salida esta correcta se aumenta el peso de las conexiones en la red neuronal, pero si la respuesta es incorrecta se reduce el peso de las conexiones.

Durante esta etapa puede haber ruido en los datos de entrada, el ruido se interpreta como imperfecciones en los datos. Por ejemplo si estamos entrenando una red neuronal para identificar si una imagen es o no una X, el ruido podría ser un punto negro puesto al azar en algún lugar de la imagen. Si en los datos que vamos a utilizar a menudo existen imperfecciones es necesario incluir ruido durante esta etapa.

5.2.1.2 Aplicando aprendizaje supervisado:

Vamos a explicar paso a paso como se ejecuta el proceso de aprendizaje supervisado, ya que es el más utilizado para redes neuronales. Debemos seguir los siguientes pasos:

- 1- Asignar valores aleatorios pequeños a todos los weights de la red neuronal. Luego pasar un set de entrenamiento a la red neuronal. Para cada elemento del set debemos tener un conjunto de entradas, pesos y la salida correcta.
- 2- Para cada elemento de entrenamiento calcular el error de cada weight. El error es la diferencia entre el weight actual de la conexión y el weight provisto por el elemento de entrenamiento. Únicamente se deben de actualizar los elementos de entrenamiento que no han tenido una salida satisfactoria.
- 3- Para cada conexión en la red tomar el error y multiplicarlo por un coeficiente de aprendizaje. $Weight = Weight + error * CA$. El coeficiente de aprendizaje se utiliza para no alterar mucho los pesos en un solo ciclo. Por ejemplo si el error es igual a



0.3 el peso aumentaría en 0.3, pero multiplicado por un coeficiente de 0.1 el aumento es únicamente de .03.

- 4- Repetir el proceso para cada uno de los elementos del set de entrenamiento.

Luego de la fase de aprendizaje podemos hablar de otra de las características más importantes de una red neuronal, la generalización. Una vez que la estructura de la red neuronal ha sido actualizada con un set de entrenamiento es posible encontrar respuestas para vectores de entrada que aun no han sido probados.

Este concepto es llamado generalización, la red neuronal basándose en el conocimiento adquirido posee la capacidad de calcular la respuesta para un vector de entradas que la red nunca ha visto antes. Este proceso es la culminación de la labor de aprendizaje, es el objetivo de la red neuronal. La red neuronal compara los valores de entrada que le damos contra los valores de entrada de un ejemplo de entrenamiento, basándose en este principio nos da una respuesta similar a la respuesta de la entrada con la cual se hizo la comparación. Gráficamente esto se explica de la siguiente manera:

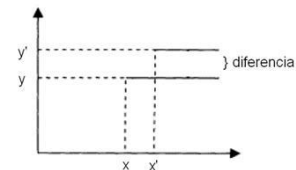


Figura No. 17 – Generalización
Fuente: Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering por Nikola K. Kasabov

X es un vector de entrada y su valor de salida y que ha sido aprendido por medio del proceso de aprendizaje. Tenemos un nuevo valor X' y para este valor de entrada se tiene la salida y' . La diferencia entre ambos pares de entrada y salida se interpreta como la generalización.

5.3 Función de activación

La función de activación como habíamos explicado antes nos sirve para encontrar el nuevo estado de la neurona, esto a partir del estado anterior de la neurona y las entradas netas. Vamos a ver las funciones de activación más comunes.



Figura 50 – Páginas 69-72 Libro “Inteligencia Artificial”

5.3.1 Función escalón:

En esta función se tiene un límite establecido. Si el valor de entrada supera este límite se interpreta como activado (1). Si el valor de entrada es menor al límite establecido se interpreta como desactivado (0).

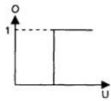


Figura No. 18 – Función escalón

Fuente: Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering por Nikola K. Kasabov

5.3.2 Función de límite lineal:

La función utilizada aquí tiene un crecimiento lineal, mientras mayor sea el valor de entrada mayor será el nivel de activación, pero una vez que se cruza un límite establecido la función se considera completamente activada (1) o desactivada (0) ya que los límites se encuentran a ambos extremos.

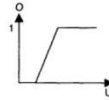


Figura No. 19 – Función Límite Lineal

Fuente: Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering por Nikola K. Kasabov

5.3.3 Función sigmoideal:

Esta función en forma de hélice puede ser utilizada en un espacio positivo (rango entre 0 y 1) o en espacios positivos y negativos (rango entre -1 y 1). Los valores para esta función crecen a medida que crece la entrada siempre en la misma dirección. Para lograr esta forma se puede usar la función de logaritmo natural, la función de tangente hiperbólica, la función logística y otras funciones. Esta función es la más utilizada ya que es la que más se asemeja al funcionamiento de las neuronas orgánicas.

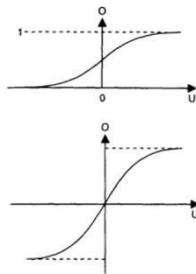


Figura No. 20 – Función sigmoideal

Fuente: Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering por Nikola K. Kasabov

5.3.4 Función Gausiana:

Para esta función mientras más cerca este el valor de entrada al 0 la función se aproxima a 1 y mientras más lejos este la entrada de 0 la respuesta se aproxima a 0.

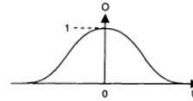


Figura No. 21 – Función Gausiana

Fuente: Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering por Nikola K. Kasabov

5.4 Tipos de redes neuronales

Las conexiones entre las neuronas pueden estar configuradas de varias maneras. Las neuronas pueden estar “completamente conectadas” donde todas las neuronas están conectadas entre sí. Pueden también estar “parcialmente conectadas”, en donde nosotros decidimos que neuronas deben estar conectadas.

Existen dos clasificaciones en las redes neuronales dependiendo del número de capas que posee.

5.4.1 Auto-asociativas:

Aquí las neuronas de entrada son procesadas en una sola dirección y van directamente a la salida, no hay comunicación con neuronas adyacentes.



Figura 51 – Páginas 73-76 Libro “Inteligencia Artificial”

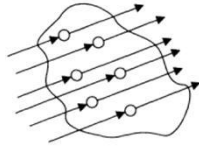


Figura No. 22 – red auto asociativa
Fuente: Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering por Nikola K. Kasabov

5.4.2 Hetero-asociativas:

Para este tipo de redes es posible crear un conjunto de neuronas en diferentes capas, de esta manera es fácil crear un grupo de entradas y de salidas.

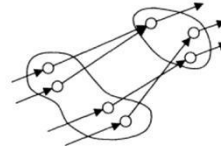


Figura No. 23 – Red hetero asociativa
Fuente: Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering por Nikola K. Kasabov

Existe otra manera de clasificar las redes neuronales, esta clasificación está basada en la existencia o ausencia de recursión en la misma, se tienen dos clasificaciones.

5.4.3 Arquitectura unidireccional (Feedforward):

Aquí las neuronas viajan únicamente en una dirección por lo que no hay conexiones entre neuronas de un nivel posterior a un nivel anterior. En esta red las neuronas no recuerdan valores de salida generados anteriormente y aquí no se lleva control de los estados de activación. El estado de activación para esta arquitectura es siempre el estado anterior y nunca cambia.

73

Inteligencia Artificial

74

Inteligencia Artificial

5.4.4 Arquitectura recursiva o de retroalimentación (feedback):

Aquí si es posible tener conexiones entre niveles superiores a niveles inferiores. Para este tipo de arquitectura es crucial llevar control de los estados de activación de las neuronas, ya que de otra manera nunca se podría salir del ciclo creado en la red.

5.5 MLP Multi Layer Perceptron

A pesar de la gran capacidad que tienen los perceptrón para resolver problemas que pueden ser separados linealmente siempre tuvieron su limitación, a raíz de esto nacen los perceptrón multi capa.

Si recordamos la estructura del perceptrón normal podemos identificar dos capas claramente, la capa donde se tienen los nodos de entrada y la capa del perceptrón. Para este modelo tenemos una capa extra llamada capa oculta, formando así una tercera capa. La capa oculta a su vez puede estar compuesta de una o más capas.

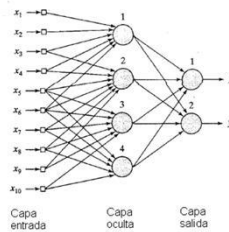


Figura No. 24 – MLP
Fuente: Neural networks a Comprehensive Foundation por Simon Haykin

Si consideramos las múltiples capas de la capa oculta como sub capas podemos identificar tres capas principales. Tenemos la capa de entrada donde se tienen los nodos de entrada, la capa oculta y la capa de salida donde tenemos las neuronas que producen los valores de salida.

Al igual que con el perceptrón normal los MLP son unidireccionales y utilizan como cálculo de las entradas netas la sumatoria. La diferencia principal radica en la función de activación ya que utilizan funciones no lineales, en la mayoría de los casos la

75

Inteligencia Artificial

76

Inteligencia Artificial

Figura 52 – Páginas 77-80 Libro “Inteligencia Artificial”

función sigmoidal. Otra de las grandes diferencias es el algoritmo de aprendizaje que utiliza, para los MLP se creó uno nuevo llamado algoritmo de error de propagación inversa.

Este algoritmo utiliza la regla de gradiente descendiente, lo que hace esta regla por nosotros es que nos ayuda a converger a un punto de equilibrio más rápido al equilibrar los weights de nuestra red. La regla se basa en encontrar los puntos mínimos de la función de error, por lo que para encontrar estos puntos mínimos es necesario utilizar la derivada de la función.

La manera en que calculábamos el cambio en los weights para el perceptrón era la siguiente:

$\Delta w = CA * error$, recordemos que CA es el coeficiente de aprendizaje.

El cambio entre los weights usando el algoritmo de error de propagación inversa y la regla gradiente descendiente es:

$$\Delta w = CA * error_j * Fa'(x_j) * y_i$$

Para comprender esta fórmula debemos saber que el cálculo de Δw está basado en la relación entre dos neuronas que se encuentran en diferentes capas, llamamos i a la neurona de la izquierda y j a la neurona de la derecha.

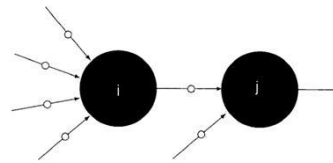


Figura No. 25 – Neuronas i,j

Para la fórmula tenemos que Fa' es la derivada de la función de activación. X_j es el valor de entrada para la neurona j. Y_i es el valor de salida para la neurona i.

Vamos a asumir que estamos usando la función logística como función de activación que es la siguiente: $y = 1/(1 + e^{-(c*x)})$

Esta función es muy popular para modelar la función sigmoidal ya que tiene una derivada sencilla que es la siguiente:

$$C/(1 + e^{-(c*x)}) * (1 - 1/(1 + e^{-(c*x)}))$$

que equivale a: $C * Y(1-y)$, donde y es la función logística misma y si $C=1$ entonces tenemos la función $Y(1-y)$.

Por lo tanto podemos decir que el cambio entre los weights se reduce a esta fórmula:



$$\Delta w = CA * error_j * y_j (1-y_j) * y_i$$

Para llegar a esta fórmula únicamente reemplazamos la derivada de la función de activación por la derivada de la función logística. Por supuesto esto solo es válido si usamos la función logística como función de activación, esto se hizo con el fin de dejar la fórmula más fácil de interpretar.

Así que con esta fórmula únicamente estaremos trabajando con las salidas de las neuronas i, j. Esta es la ventaja de usar la función logística. Notemos que para este algoritmo es crucial que la función de activación sea derivable.

Como podemos ver estamos usando el error y la salida de la neurona j para calcular el weight de i. Es por esta razón que se dice que el error se está propagando hacia atrás ya que las neuronas de capas anteriores heredan el error de neuronas en capas posteriores.

5.5.1 Algoritmo de Propagación Inversa:

Vamos a detallar cuáles son los pasos a seguir para el algoritmo de error de propagación inversa:

- 1- Asignar valores aleatorios pequeños para todos los weights.
- 2- Tomar el primer valor del set de entrenamiento y correrlo a través de todas las capas de la red neuronal.

3- Calcular para todas las salidas cual es el error. $Error = Valor\ obtenido - Valor\ esperado$.

4- Aquí comenzamos a utilizar la nueva fórmula para obtener el valor de Δw . Calculamos los nuevos weights para las neuronas de las capas ocultas utilizando el error de la capa de neuronas de salida (neuronas i, j).

5- Una vez que se ha propagado el error hacia la capa oculta se debe repetir el proceso para la capa de entrada. Ahora calculamos el error para las neuronas de la capa oculta, donde el error sería: $Error_i = \sum Error_j * W_{ij}$. El error de la capa oculta se calcula en base a la capa de salida y los pesos entre estas dos capas.

6- Ya con el error calculado para la capa oculta utilizamos la misma fórmula para calcular el nuevo ajuste en los weights. En este caso la capa de entrada es identificada como i y la capa oculta se identifica como j.

5.5.2 Ejemplo de MLP:

Vamos a darle solución a un ejemplo clásico no separable linealmente, solucionar la función booleana XOR:



Figura 53 – Páginas 81-84 Libro “Inteligencia Artificial”

x1	x2	y
1	1	0
1	0	1
0	1	1
0	0	0

Figura No. 26 – Tabla de verdad 3

Para este ejemplo tenemos dos variables y dos posibles resultados verdadero y falso. Este problema no es linealmente separable por lo tanto no es posible solucionarlo con un perceptrón sencillo.

Para este MLP vamos a utilizar la función logística como la función activación sigmoideal en cada uno de los nodos. Para que sea más fácil la explicación vamos a etiquetar los nodos:

Vamos a realizar una iteración completa para el proceso de aprendizaje utilizando el algoritmo de propagación inversa.

El primer paso es asignar valores aleatorios pequeños para cada uno de los weights de la red neuronal.

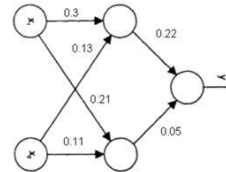


Figura No. 27 – Ejemplo XOR antes de balance

La primera parte del algoritmo de propagación inversa nos dice que debemos correr un elemento de entrenamiento a través de la red completa, vamos a utilizar los valores $x1 = 1$ y $x2=1$ con el resultado $y = 0$.

Vamos a calcular la entrada neta para los nodos A y B, notemos que estamos sumando un valor fijo 1 que viene a ser el bias del nodo. Siempre se debe utilizar un bias como una entrada extra al calcular la entrada neta.

$$A = 1 * 0.3 + 1 * 0.13 + 1 = 1.43$$

$$B = 1 * 0.21 + 1 * 0.11 + 1 = 1.32$$

Ahora como las entradas netas para los nodos A y B vamos a calcular su salida con la función de activación.

81

Inteligencia Artificial

82

Inteligencia Artificial

$$FA(1.43) = 1 / (1 + e^{-(1 * 1.43)}) = 0.8069$$

$$FB(1.32) = 1 / (1 + e^{-(1 * 1.32)}) = 0.7892$$

Ahora tenemos las nuevas entradas para el nodo C, calculamos su entrada neta y también su salida.

$$C = 0.8069 * 0.22 + 0.7892 * 0.05 + 1 = 1.2170$$

$$FC(1.2170) = 1 / (1 + e^{-(2 * 1.2547)}) = .7715$$

Ya tenemos la salida generada para la red con los valores de entrada 1 y 1. Aquí termina la primera fase del algoritmo en la cual vamos hacia la derecha, la segunda parte es la propagación inversa.

Ahora calculamos el error para la salida esperada 0.

$$\text{Error C} = 0 - 0.7715 = -0.7715$$

Con el error de la salida podemos calcular el reajuste de los weights que van hacia el nodo C de salida.

Ya que estamos utilizando la función logística vamos a usar la función que describimos anteriormente para encontrar el cambio a efectuar para los weights, ya que hay dos conexiones hacia C debemos efectuar dos ajustes, uno para A y otro para B. Aquí vamos a utilizar un Coeficiente de Aprendizaje igual a 0.5

$$A-C \Delta w = CA * \text{errorj} * yj (1-yj) * yi, \text{ y para } yj = 0.7715, yi = 0.8069$$

$$A-C \Delta w = 0.5 * -.7715 * .7715 (1-.7715) * .8069 = -.0549$$

$$B-C \Delta w = CA * \text{errorj} * yj (1-yj) * yi, \text{ y para } yj = 0.7715, yi = 0.7892$$

$$B-C \Delta w = 0.5 * -.7715 * .7715 (1-.7715) * .7892 = -.0537$$

Por lo tanto los nuevos weights que van en dirección de C serán:

$$A-C = 0.22 - .0549 = 0.1651$$

$$B-C = 0.05 - .0537 = -0.0037$$

Una vez completado el reajuste de weights entre la capa oculta y la capa de salida vamos a calcular cual es el error para cada nodo de la capa oculta.

$$\text{Error A} = \text{Error C} * \text{Weight A-C} + \text{Error C} * \text{Weight B-C}$$

$$\text{Error A} = -.7715 * .1651 - .7715 * -.0037 = -.1245$$

$$\text{Error B} = \text{Error C} * \text{Weight A-C} + \text{Error C} * \text{Weight B-C}$$

$$\text{Error B} = -.7715 * .1651 - .7715 * -.0037 = -.1245$$

Con los errores definidos para los nodos A y B podemos definir el cambio en los weights para la capa de entrada.

$$X1-A \Delta w = CA * \text{errorj} * yj (1-yj) * yi, \text{ y para } yj = 0.8069, yi = 1$$

$$X1-A \Delta w = 0.5 * -.1245 * .8069 (1-.8069) * 1 = -.0097$$

$$X1-B \Delta w = CA * \text{errorj} * yj (1-yj) * yi, \text{ y para } yj = 0.7892, yi = 1$$

$$X1-B \Delta w = 0.5 * -.1245 * .7892 (1-.7892) * 1 = -.0104$$

$$X2-A \Delta w = CA * \text{errorj} * yj (1-yj) * yi, \text{ y para } yj = 0.8069, yi = 1$$

$$X2-A \Delta w = 0.5 * -.1245 * .8069 (1-.8069) * 1 = -.0097$$

$$X2-B \Delta w = CA * \text{errorj} * yj (1-yj) * yi, \text{ y para } yj = 0.7892, yi = 1$$

$$X2-B \Delta w = 0.5 * -.1245 * .7892 (1-.7892) * 1 = -.0104$$

83

Inteligencia Artificial

84

Inteligencia Artificial

Figura 54 – Páginas 85-88 Libro “Inteligencia Artificial”

Con estos valores podemos equilibrar los pesos para la capa de entrada:

X1-A=0.3-.0097=.2903
 X1-B=0.21-.0104=.1996
 X2-A=0.13-.0097=.1203
 X2-B=0.11-.0104=.0996

Y con estos valores la grafica queda de la siguiente manera:

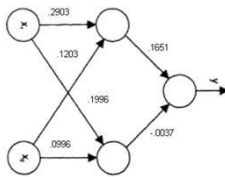


Figura No. 28 – Ejemplo XOR balanceado

Este proceso se repite para cada uno de los elementos del set de entrenamiento.

5.5.3 Teorema de Kolmogorov

El teorema de Kolmogorov más allá de su comprobación matemática nos interesa por la afirmación que realiza, “Cualquier función continua compuesta por valores reales en un cubo n-dimensional puede ser representada como una suma de funciones cuyos componentes son la suma de funciones continuas de una variable”.

$$f(x_1, x_2, \dots, x_n) = \sum_{k=1}^{2n+1} g_k \left(\sum_{i=1, n} \eta_{k,i}(x_i) \right)$$

Figura No. 29 – Teorema de Kolmogorov

Fuente: Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering por Nikola K. Kasabov

Este teorema es importante porque nos asegura que para cualquier función continua se puede crear un MLP que pueda llegar a aproximarse lo suficiente a dicha función. En este caso la suma de funciones es representada por cada nodo en la red neuronal por medio de su función de activación que debe ser sigmoideal.

Es fácil poder apreciar a partir del primer ejemplo que realizamos cuantos nodos debe haber en la capa de entrada y cuantos nodos en la capa de salida. El número de nodos de la capa de entrada es igual al número de entradas que posee el problema, y lo mismo se puede decir para la capa de salida en cuanto al número de salidas.



85

Inteligencia Artificial



86

Inteligencia Artificial

El problema surge a la hora de calcular cuántos nodos deben de haber en la capa oculta e incluso cuantas subcapas debe tener esta. Si nosotros tenemos un exceso de nodos ocultos vamos a tener un porcentaje de error bajo durante el entrenamiento pero durante la generalización vamos a tener una tasa de error alta, esto se debe a un sobre ajuste para la función. Si nosotros tenemos escasez de nodos vamos a tener una tasa de error alta tanto durante el entrenamiento como para la generalización, esto es porque no hemos logrado ajustar a la función lo suficiente.

Encontrar este punto de equilibrio no es tan fácil ya que no existe una convención sobre cómo tratar este problema. Existe una gran cantidad de formulas y opiniones diferentes sobre este tema pero en general tenemos dos opciones principales:

Una opción es realizar varias pruebas con diferentes redes neuronales, y luego calcular el error de generalización para cada una de ellas luego de la fase de entrenamiento. Utilizamos algunas funciones heurísticas como punto de partida para encontrar este balance:

Una fórmula que es utilizada por muchos autores para encontrar el número de nodos ocultos es la siguiente:

$$\text{Nodos: } (\text{numero entradas} + \text{numero salidas}) * 2/3$$

La mayoría de autores recomiendan únicamente utilizar una subcapa en la capa oculta, ya que al tener más capas usando vector gradiente la red neuronal se vuelve inestable.

Debemos hacer notar que existen otras formulas y en general ninguna nos llevara a una respuesta acertada sobre el numero de nodos ocultos, la única manera de llegar al punto optimo es realizar varias pruebas con diferente numero de nodos.

La otra opción es un proceso en el cual comenzamos con una cantidad elevada de nodos ocultos, luego comenzamos la fase de entrenamiento. Durante el entrenamiento en intervalos fijos vamos verificando que conexiones tienen los weights más bajos, estas conexiones son las que no están siendo utilizadas y por lo tanto se van eliminando estos nodos.

Artículos Relacionados

- **Incremental Evolution in ANN's: Neural Nets which grow**
 Christopher MacLeod and Grant M. Maxwell
 Ref: http://www.rgu.ac.uk/files/paper_text_b.pdf
- **Statistical Arbitrage Stock Trading using Time Delay Neural Networks**



87

Inteligencia Artificial



88

Inteligencia Artificial

Figura 55 – Páginas 89-92 Libro “Inteligencia Artificial”

Chris Pennock
Ref: http://cs.nyu.edu/~ccp252/media/Statistical_Arbitrage_TDNN.pdf

• *Event detection and localization for small mobile robots using reservoir computing*

E. A. Antonelo, B. Schrauwen and D. Stroobandt

Ref:
http://snn.elis.ugent.be/sites/snn/files/robotlocalization_antonelo_nn2008_0.pdf

• *Java Object Oriented Neural Engine*

Paul Marrone
Ref: <http://www.jooneworld.com/docs/documentation.html>

Capítulo 6

AGENTES

En este capítulo estudiamos la rama de la Inteligencia Artificial Macro Distribuida. Vemos el tema de agentes inteligentes, comenzamos con definiciones y clasificaciones para los mismos. Vemos la importancia de sus características y el medio en el que se encuentran. Por último vemos las diferentes arquitecturas con las cuales se pueden implementar.

89

Inteligencia Artificial

90

Inteligencia Artificial

ALCANCE

6. Agentes

6.1. Agentes

- 6.1.1. Agentes de Software
- 6.2. Características de los agentes
 - 6.2.1. Autonomía
 - 6.2.2. Inteligencia
 - 6.2.3. Pro actividad
 - 6.2.4. Reactividad
 - 6.2.5. Habilidad de aprender
 - 6.2.6. Cooperación
 - 6.2.7. Movilidad
- 6.3. Clasificación de Agentes
 - 6.3.1. Agentes Reactivos
 - 6.3.1.1 Agentes orientados a una meta
 - 6.3.1.2 Agentes basados en la utilidad
 - 6.3.2. Agentes de interfaz
 - 6.3.3. Agentes móviles
 - 6.3.4. Agentes de información
- 6.4. Sistema Multi Agentes
 - 6.4.1. Comunicación
- 6.5. Clasificación de las propiedades del medio ambiente

- 6.5.1. Accesibles e Inaccesibles
- 6.5.2. Determinista y no Determinista
- 6.5.3. Episódico y no Episódico
- 6.5.4. Estático y Dinámico
- 6.5.5. Discreto y Continuo
- 6.6. Arquitecturas
 - 6.6.1. Arquitectura basada en la lógica
 - 6.6.2. Arquitectura reactiva
 - 6.6.3. Arquitectura BDI

6.1 Agentes

Comenzamos con la definición literal de agente. Según la real academia española esta es:

- “ adj. Que obra o tiene virtud de obrar.
- m. Persona o cosa que produce un efecto.
- m. Persona que obra con poder de otra.”

91

Inteligencia Artificial

92

Inteligencia Artificial

Figura 56 – Páginas 93-96 Libro “Inteligencia Artificial”

Parecerá un poco trivial comenzar con la esta definición, pero existe cierto grado de confusión a la hora de definir lo que es un agente de software. El propósito de un agente, pensando del nivel más bajo, es el de realizar una acción en nombre de otra entidad. Parece una extensión de la definición literal de la palabra ya que ahora hablamos de una entidad. Esta definición más amplia se da en este momento para hacer evidente que no solamente personas pueden tener a su cargo agentes. Un agente en si también puede tener a su cargo a otro agente.

De la misma manera un agente no necesariamente tiene que ser un programa o una aplicación. Según Ben Coppin, un agente puede ser biológico, robótico o computacional. Al decir que puede ser biológico estamos diciendo que un agente puede ser por ejemplo un perro guía, que ayuda a las personas ciegas a caminar. Este tipo de agentes es menos evidente, pero cumple con las características de lo que es un agente. Los agentes robóticos son más fáciles de imaginar en este contexto, cualquier robot que sea capaz de realizar una acción sin intervención humana es un agente robótico.

Según la AAAI (Asociación para el avance de la Inteligencia Artificial), obtenido del LCI de la Universidad de British Columbia: "un agente es algo que actúa en un ambiente, como un robot móvil, un web crawler, un sistema médico de diagnóstico automático o un personaje en un video juego". Esta definición nos

muestra de una mejor manera que tipo de "entidades" puede tomar forma como un agente. Se habla de seres tanto físicos como lógicos, incluso seres que solo existen en juegos de computadora. El área de juegos interactivos es uno de los lugares donde más se ha estado usando el concepto de agentes, para diseñar y lograr crear escenarios realistas.

Existen varias características importantes que identifican a los agentes, un agente debe tener cierto grado de autonomía. Esta es la característica más importante en un agente, ya que le da a un agente la capacidad de demostrar cierto grado de creatividad. La autonomía que se le da a un agente, permite que este encuentre múltiples maneras de resolver un problema o un objetivo. De esta manera puede encontrar soluciones que nunca podrían haber sido propuestas por la entidad que está a su cargo.

Otra característica muy importante es el grado de pro actividad y el grado de reactividad de un agente. Estas características parten de la autonomía, pero está en si no es suficiente. Un agente debe ser proactivo para poder anticiparse a ciertas situaciones. Y también deber ser reactivo, ya que debe responder en cierta manera a su medio ambiente. Debe tener la capacidad de obtener información valiosa que se encuentra en el medio en el que se desarrolla.



93

Inteligencia Artificial



94

Inteligencia Artificial

Otras características que un agente debe tener son: capacidad de aprendizaje, cooperación y movilidad. Se pueden considerar como características terciarias comparadas con las enumeradas anteriormente, pero como veremos en las secciones siguientes son de gran importancia tanto en la definición de agentes como en su clasificación. La cooperación da lugar a que surja una de las clasificaciones más importantes sobre agentes que son los sistemas multi-agentes. Como lo indica el nombre son sistemas que están basados en múltiples agentes, los cuales tienen como característica principal la cooperación.

6.1.1 Agentes de Software

A diferencia del término literal agente, los agentes de software no tienen una definición tan específica. Por esta razón vamos a dar una serie de definiciones y al final vamos a crear una definición con todos estos términos.

Según la AAAI: "Un agente inteligente es un agente que actúa para lograr satisfacer sus objetivos. El agente debe percibir su ambiente, decide que acción tomar y luego procede a ejecutar la acción". Esta definición nos describe varias de las características que debe tener un agente entre ellas: reactividad y pro actividad. La definición comienza con una referencia clara a la propiedad de autonomía.

Según el departamento de Computación, Imperial College, en Londres Inglaterra obtenemos dos definiciones, (Fariba Sadri): "un agente es un sistema de computadora que es capaz de exhibir alguna forma de inteligencia artificial y acción independiente de parte de su dueño". Esta definición nos muestra nuevamente la importancia de la autonomía y nos muestra la importancia de mostrar rasgos de inteligencia artificial.

La segunda definición de la universidad es (Stan Franklin y Art Graesser): "los agentes son software capaz de acción autónoma, flexible y con propósito en la búsqueda de una o más metas". "Diseñados para tomar acciones a estímulos externos". "Al haber varios agentes en un sistema, se espera que interactúen juntos para lograr sus metas". Estos tres extractos nos muestran todas las características principales que hemos expuesto en la sección anterior. Habla de reactividad, autonomía y colaboración.

J.J. Merelo nos dice: "Un agente autónomo es un sistema situado en y parte de un entorno que siente ese entorno y actúa sobre él, a través del tiempo, persiguiendo sus propios objetivos de forma que afecte lo que siente en el futuro". Nuevamente vemos la característica autonomía que están importante, vemos la importancia de percatarse de su ambiente y de las acciones que puede tomar para cambiarlo. Vemos también la importancia que tienen las metas en el comportamiento de un agente.

Según Shoham se tiene esta definición: "un agente es una entidad cuyos estados consisten de componentes mentales (por ejemplo: creencias, capacidades, decisiones y compromisos). Así



95

Inteligencia Artificial



96

Inteligencia Artificial

Figura 57 – Páginas 97-100 Libro “Inteligencia Artificial”

que en concepto de agente o la agencia está en la mente del programador”. Esta definición es más conceptual, nos dice que las limitaciones y alcances del agente dependen completamente del programador. Y que el agente se basa en estados mentales para llevar a cabo sus objetivos, o para realizar un plan de cómo llevarlos a cabo.

A partir de estas definiciones podemos dar la siguiente definición. Un agente de software es una entidad cuyo único propósito es lograr las metas que le han sido impuestas por parte de su dueño. Debe exhibir un alto grado de autonomía para la resolución creativa de problemas y debe ser capaz de poder interactuar con su medio ambiente.

De esta nueva definición resaltamos algunas observaciones. La razón de ser de un agente es cumplir con su objetivo impuesto. Si un agente no muestra un alto grado de autonomía deja de ser un agente. Simplemente es un programa, lo que separa a los agentes de otros programas o rutinas es la manera en que resuelven los problemas. Se dice que muestran creatividad en la resolución de los mismos porque la manera en que van a llegar al objetivo no va a ser la misma. El agente va a tomar los recursos que tiene a su disposición y va a tomar una decisión acerca de cómo va a utilizar estos recursos para resolver el problema. Un agente debe ser capaz de interactuar con su medio ambiente ya que es la fuente principal de información en la mayoría de los casos. En otras ocasiones el agente debe ser capaz de alterar el medio ambiente en el que se

encuentra, siempre que sea una manera efectiva de llegar a su objetivo.

6.2 Características de los agentes

Existe una serie de características que los agentes tienen en común. Algunas de las características son básicas para su existencia, otras son de menor importancia, por lo que los agentes no deben mostrar necesariamente todas las características que se enumeran en esta sección.

6.2.1 Autonomía

La autonomía es la habilidad de tomar decisiones sin la necesidad de tener una interacción con el usuario del agente. Se considera con uno de los aspectos más importantes a la hora de identificar a los agentes.

Si un agente no muestra cierto grado de autonomía, no es un agente de software es simplemente un programa. Existe una serie de características en las cuales está basada la autonomía, las cuales se explican en páginas posteriores. Pero si sobre estas características base no se tiene autonomía, no se puede clasificar a este programa como agente.

Se debe tomar en cuenta también el grado de autonomía que debe tener un agente. Ya que su utilidad se puede ver limitada por un grado de autonomía muy alto. Por ejemplo se tiene el caso de

97

Inteligencia Artificial

98

Inteligencia Artificial

tener un agente a cargo de comprar y vender acciones en base al precio que tiene en ese momento, la historia del comportamiento de la acción, etc. Puede llegar un momento en el cual realizar una compra que supere los 100,000 Q se considere peligrosa. Por esta razón el grado de autonomía que tiene un agente debe ser revisado cuidadosamente, por esta razón el usuario debe decidir qué grado de autonomía es apropiado para su caso específico.

6.2.2 Inteligencia

Un agente tiene como objetivo realizar una tarea por parte de su usuario. La característica de la inteligencia va mas allá de la manera en que funciona un programa normal. Un agente tiene información sobre el ambiente en el cual está trabajando, esto le permite llevar a cabo sus objetivos incluso cuando las variables del problema cambian o cuando surgen eventos para los cuales no está preparado en su codificación inicial.

Por ejemplo tomamos el caso de un agente que está a cargo de revisar el tráfico de internet de los usuarios de una oficina. Tiene una serie de clasificaciones para ciertas páginas y se encarga de dar un reporte sobre el acceso a todas las páginas de internet. Este agente usa el historial de internet para obtener su información. Pero en un caso en particular uno de los usuarios desactiva el historial. El agente es capaz de detectar una reducción en la actividad de un usuario y luego tomar medidas. Puede volver a activar el historial y bloquear su acceso, o mantenerse en una

estación una vez que el historial fue desactivado tabulando las páginas hasta que el historial sea reactivado.

6.2.3 Pro-Actividad

Existe una propiedad derivada de la inteligencia llamada la pro-actividad. Se define como la capacidad de crear nuevas sub-metas que derivan de los objetivos principales. Como se ve claramente no se puede llegar a esta propiedad sin mostrar signos de inteligencia.

Siguiendo el caso del agente que revisa tráfico en internet, revisando su meta principal se percató de que primero debe realizar una tarea importante. Debe buscar una manera de almacenar dicha información, el agente decide crear un archivo binario para almacenarla. Esta es una sub-meta derivada del objetivo principal.

6.2.4 Reactividad

Se define como la capacidad de realizar acciones en base a estímulos externos que se encuentran en su medio ambiente. Esta es otra característica que se deriva directamente de la inteligencia. La capacidad de estar consciente del medio en el que se encuentra un agente se ha enfatizado en varias ocasiones.

99

Inteligencia Artificial

100

Inteligencia Artificial

Figura 58 – Páginas 101-104 Libro “Inteligencia Artificial”

Un ejemplo claro de esta propiedad se encuentra en el ejemplo del agente que recolecta el historial de páginas. El estímulo externo que él toma en cuenta es la acción de un usuario entrando a una página. El ambiente en el cual se desarrolla es la red LAN interna de la organización. Se puede apreciar claramente la importancia de tomar en cuenta la información externa al agente para tomar decisiones.

6.2.5 Habilidad de aprendizaje

La habilidad de aprendizaje se puede definir como, tener la capacidad de alterar su funcionamiento interno con el objetivo de mejorar su desempeño. Un agente puede obtener retroalimentación sobre una tarea específica de parte de su usuario. En caso de haber realizado mal una tarea, el agente desvía la mayoría de recursos para mejorar esta tarea antes de seguir con otras actividades.

Tomemos por ejemplo un programa que clasifica nuestro correo electrónico como deseado y no deseado o spam. Este programa se basa en una serie de parámetros para identificar que correo es útil, esto es un programa de filtro de correos. Pero ahora tomemos como ejemplo el mismo programa agregándole ciertas características. El programa no necesita una serie de parámetros para identificar que correo es útil y cual es basura. Existe un proceso detrás de la selección de correo basura, cada vez que el usuario marca un correo electrónico como no deseado el programa

obtiene las características de este correo y en base a selecciones anteriores crea una serie de reglas para clasificar los correos.

6.2.6 Cooperación

La cooperación es la habilidad de compartir información entre agentes compañeros, o de tener cierto grado de comunicación. La cooperación es crucial cuando se habla de sistemas multi-agentes, ya que este concepto se basa en la premisa de que la colaboración de agentes individuales logra un resultado efectivo.

Tomemos como base el ejemplo del agente que se encarga de tabular el uso de páginas web en la empresa. Este agente trabaja para un agente que está a cargo de la seguridad. El agente de seguridad ha escuchado la noticia sobre el usuario que ha estado desactivando su historial de páginas. Esta información fue transmitida por el agente que revisa el acceso a las páginas. También ha obtenido información de un agente encargado de monitorear virus y aplicar la vacuna. Le ha informado que ha aumentado el grado de actividad de virus en ese mismo usuario. Gracias a la interacción con otros agentes, el agente encargado de seguridad ahora tiene la información necesaria para tomar una decisión sobre este usuario.

El escenario anterior nos muestra que la efectividad de los agentes se incrementa de gran manera al utilizar la cooperación. De manera análoga cuando se está trabajando en proyectos grandes y

101

Inteligencia Artificial

102

Inteligencia Artificial

se divide el mismo en una serie de módulos, cuando se habla de agentes se está dividiendo el proyecto en una serie de agentes cooperativos.

Es importante notar que no todos los agentes son cooperativos. De hecho algunos agentes pueden haber sido creados con el objetivo de no compartir información. Algunos agentes pueden haber sido creados con el único propósito de obstruir el camino de otros agentes. Incluso algunos agentes tienen la capacidad de mentir para lograr estos objetivos. Este tipo de agentes puede ser creado, pero en la mayoría de los casos los agentes tienen objetivos benignos.

6.2.7 Movilidad

El grado de movilidad se define como la capacidad de viajar en una red local o en internet. Para muchos agentes la movilidad es muy importante para lograr sus objetivos, por lo que se toma como una característica importante.

En uno de los casos expuestos ya se utilizó esta propiedad. El agente encargado de revisar el historial de acceso a páginas tiene la habilidad de viajar hacia las estaciones de trabajo de usuarios específicos.

Como una propiedad complementaria a la movilidad se tiene la habilidad de replicación. Tenemos el mismo caso del agente que revisa el historial de páginas cuando tiene que quedarse en una estación esperando a un usuario. Este agente puede ser capaz de

dejar una copia de él mismo en esa estación y luego volver al lugar donde estaba trabajando. Con la limitación opcional que luego de cumplir su tarea de esperar que el historial se cierre, el programa replicado debe borrarse a sí mismo. Esta propiedad se aprecia claramente al ver cómo funcionan los virus de computadora.

6.3 Clasificación de Agentes

Los agentes se pueden clasificar de acuerdo a una serie de parámetros. Los tipos de agentes que veremos no son mutuamente excluyentes, un agente de interfaz puede ser reactivo y de interfaz a la vez. Vamos a desarrollar las diferentes clases de agentes que existen:

6.3.1 Agentes Reactivos

Este tipo de agentes se describe al explicar la característica de reactividad. A este tipo de agentes también se les llama agentes de reflejo porque actúan automáticamente en reacción a un estímulo externo.

Este tipo de agentes está siempre atento a las variables del medio ambiente en el que se encuentra, las entradas que recibe son los cambios de estas variables, estos se consideran como estímulos. Los agentes reactivos por lo tanto toman esta información y en base a reglas preestablecidas reaccionan o realizan una acción.

103

Inteligencia Artificial

104

Inteligencia Artificial

Figura 59 – Páginas 105-108 Libro “Inteligencia Artificial”

Ya que la mayoría de agentes de este tipo se especializan en un medio determinado, cuando su medio cambia no logran lidiar bien con este cambio. La especialización del medio incrementa su desempeño, pero para casos en los cuales el medio es variable se debe preparar al agente para este cambio.

Existen tres sub clasificaciones para los agentes reactivos estos son:

6.3.1.1 Agentes orientados a una meta

Este tipo de agentes tiene un grado más alto de autonomía comparado a un agente reactivo normal. En lugar de basarse en un conjunto de reglas preestablecidas para reaccionar ante su medio un agente orientado a una meta realiza un análisis para lograr su objetivo. Este análisis está basado ya sea en búsquedas o en la planeación. Por la manera en que se comporta este tipo de agente no considera cuantos recursos ha consumido, lo único que le importa es llegar a la meta impuesta.

6.3.1.2 Agentes basados en la utilidad

Es muy similar a los agentes orientados a una meta pero a diferencia de estos los agentes orientados a la utilidad si toman en cuenta los recursos que tienen a su disposición. Aquí se tienen dos objetivos principales. La primera es llegar a la meta impuesta por su usuario y la segunda es optimizar la utilización de recursos para

lograrlo. El recurso más común que se llega a optimizar en la mayoría de los casos es el tiempo.

Podemos ver ahora la diferencia entre los agentes reactivos, orientados a una meta y orientados a la utilidad. La diferencia entre estos es básicamente la motivación que tienen para llegar a su meta. Aquí se introduce lo que se considera el nivel de felicidad del agente, la felicidad se traduce al grado de éxito que tiene el mismo. La felicidad del agente es un factor que se puede llegar a utilizar en muchas maneras, por ejemplo para agentes orientados a la utilidad un recurso que se puede llegar a optimizar es la felicidad (o la probabilidad de éxito).

6.3.2 Agentes de Interfaz

Aquí la palabra interfaz realmente se refiere al usuario. Los agentes de interfaz interactúan directamente con el usuario, usualmente tienen una interfaz amigable o una buena animación para interactuar. Los asistentes interactivos caen bajo esta clasificación, por ejemplo asistentes de escritorio que ayudan en las búsquedas.

Es común que este tipo de agentes no interactúen con otros agentes únicamente con el usuario pero no es una regla, es posible de ser necesario que interactúen con otros agentes. Este tipo de agentes es utilizado comúnmente para realizar labores repetitivas

105

Inteligencia Artificial

106

Inteligencia Artificial

por parte del usuario, luego de ver varias veces las acciones del usuario se le puede indicar que este las realice por su cuenta.

6.3.3 Agentes Móviles

La movilidad tiene un significado diferente dependiendo del tipo de agente que usamos. Para agentes robóticos el medio en el cual son móviles son en la vida real, pueden caminar en un lugar específico. Para agentes de software la movilidad es en la red, ya sea en una red interna o en internet.

Utilizando agentes móviles tenemos la habilidad de distribuir el trabajo en varias computadoras e incluso usar los recursos de una computadora más poderosa a la computadora donde reside normalmente. Este tipo de movimiento también reduce el consumo de ancho de banda al realizar operaciones directamente en otras computadoras en vez de hacer solicitudes en la red. Los agentes que no tienen la capacidad de movilidad se les llaman estáticos.

6.3.4 Agentes de Información

Los agentes de información son agentes utilizados para buscar información en internet. Por esto se les llama también agentes de internet aunque también se les llama agentes de recopilación de información. Este tipo de agentes son muy útiles ya que en algunos casos alivian la labor de búsqueda de información por su gran rapidez y la posibilidad de trabajar de manera asincrónica.

Un ejemplo sencillo de este tipo de agentes son los search bots utilizados por los motores de búsqueda para actualizar sus bases de datos. Agentes de información más sofisticados son capaces de aprender e incluso trabajar en conjunto con otros agentes.

6.4 Sistemas multi-agente

Una característica importante de los agentes dentro de un sistema multi-agente es que por su cuenta cada agente no puede resolver el problema completo. Cada agente aporta una parte de la solución y sus esfuerzos unidos le dan la solución al problema. De hecho cada uno de los agentes no necesita saber cuál es el objetivo final del sistema completo, lo único que debe tener claro es cuál es su sub meta dentro del sistema. Los agentes pueden ser colaborativos o competitivos, esto depende del objetivo final del sistema.

Una de las características más interesantes para agentes de este tipo es la posibilidad de poder trabajar en paralelo, en un sistema bien sincronizado se puede lograr reducir considerablemente el tiempo de procesamiento para la resolución del problema.

107

Inteligencia Artificial

108

Inteligencia Artificial

Figura 60 – Páginas 109-112 Libro “Inteligencia Artificial”

6.4.1 Comunicación

La comunicación en un sistema multi-agente es muy importante, la manera en que se comunican es por medio de mensajes. Se debe establecer un protocolo a seguir para la comunicación en el sistema.

En general existen dos tipos principales de mensajes, tenemos las afirmaciones y tenemos las preguntas. A su vez tenemos otras subdivisiones que son la capacidad de recibir y proponer dichos mensajes. Todo tipo de agente debe tener por lo menos la primera habilidad que es tener la capacidad de recibir afirmaciones. En este contexto una afirmación es la transmisión de información.

Tenemos agentes que llegan a un nivel más alto de comunicación, los agentes que son capaces de recibir tanto afirmaciones como preguntas y tienen la capacidad de responder preguntas son llamados agentes pasivos. Existe un agente que tiene un nivel de comunicación similar pero con un diferente enfoque, un agente que puede recibir afirmaciones, es capaz de mandar afirmaciones y también es capaz de realizar preguntas es un agente activo.

El nivel más alto de comunicación lo alcanzan los agentes compañeros (peer agents), este tipo de agentes es capaz tanto de proponer afirmaciones y preguntas como recibir afirmaciones y preguntas.

Basados en estas características podemos explicar esta clasificación de manera gráfica.

	Básico	Pasivo	Activo	Compañero
Propone preguntas			+	+
Propone afirmaciones		+	+	+
Recibe preguntas		+		+
Recibe afirmaciones	+	+	+	+

Figura No. 30 – Comunicación

6.5 Clasificación de las propiedades del medio ambiente

Como se ha discutido anteriormente el medio en el cual se encuentra un agente es muy importante. Existen características específicas con las cuales podemos clasificar un medio, es importante considerar estas variables ya que existen medios que son menos o más hostiles para aceptar un nuevo agente.

6.5.1 Accesibles e inaccesibles

Un medio accesible se describe como un lugar en el cual se encuentra información completa y exacta, es decir un medio en el cual la información es constante. Podríamos considerar este tipo de



109

Inteligencia Artificial



110

Inteligencia Artificial

ambientes como más controlados, es por esta razón que es más fácil introducir un agente en un medio accesible. Por ejemplo si tenemos un agente robótico su medio sería el mundo real, este tipo de medio complejo es catalogado como inaccesible.

6.5.2 Deterministas y no determinista

Un entorno determinista es un medio en el cual se sabe con certeza cuál será el resultado tras realizar determinada acción. En un medio no determinista es posible que tras realizar una acción ocurran imprevistos, es por esta razón que es más difícil introducir agentes en este tipo de medios.

6.5.3 Episódicos y no episódicos

La palabra episódico se refiere aquí a una sola situación o a una instancia. Los agentes siempre deben lidiar con problemas, cada vez que se le introduce un nuevo problema se tiene una nueva situación. En un medio episódico cada nueva situación se trata de manera independiente, para resolver un problema no se toma en cuenta otras posibles situaciones o episodios. En un ambiente no episódico al planear como resolver un problema se debe considerar la posibilidad de afectar situaciones futuras.



111

Inteligencia Artificial

6.5.4 Estático y dinámico

Existe una regla sencilla que divide los medios estáticos de los dinámicos. Si se trabaja en un medio que no sufre ningún cambio a menos que sea causado por el agente mismo se tiene un ambiente estático. En los medios dinámicos existen otros factores aparte del agente que pueden realizar cambios sobre este.

6.5.5 Discreto y continuo

Aquí debemos observar la cantidad de acciones que deben realizar los agentes. En un medio discreto se tiene un número determinado de acciones que un agente debe realizar. En un medio continuo no se sabe con exactitud el número de acciones que se deben realizar, en este tipo de medio es más difícil llevar el control.

6.6 Arquitecturas

En esta sección examinamos la estructura interna de un agente, la arquitectura define los detalles internos para crear, manipular y mantener funcionando a los agentes. En esta sección nos estamos alejando de la vista abstracta que hemos llevado hasta ahora y estamos detallando la manera en que se puede implementar, existen varias arquitecturas para este propósito.



112

Inteligencia Artificial

Figura 61 – Páginas 113-116 Libro “Inteligencia Artificial”

6.6.1 Arquitectura basada en la lógica

Esta arquitectura toma los principios de la IA simbólica, nos dice que en un sistema se representa la inteligencia al proveerle una representación simbólica de su entorno. Esta representación se realiza por medio de afirmaciones lógicas y luego esta información se puede manipular por medio de una deducción lógica basándose en las reglas creadas anteriormente.

Existe un lenguaje de programación que ejemplifica perfectamente la arquitectura basada en la lógica. PROLOG nos permite crear una base de datos de hechos y reglas.

Por lo tanto es común ver instrucciones de este tipo en PROLOG:

```
Nutritivo(crema)
Hecho_de(crema, leche)
Contiene(leche, calcio)
```

Estamos afirmando lo siguiente: la crema es nutritiva, la crema está hecha de leche y la leche contiene calcio.

A partir de las reglas que estamos creando podemos entonces crear reglas de deducción en la cual obtenemos valores booleanos (V o F) en base a una premisa.

Podríamos presentar el siguiente ejemplo sencillo:

Indicamos al agente que clasifique una lista de comida en dos clasificaciones diferentes saludable y no saludable.

113

Inteligencia Artificial

El agente verificara los alimentos que contengan calcio y este tipo de alimento será clasificado como saludable.

Utilizando una base de datos de reglas o afirmaciones lógicas el agente puede realizar deducciones lógicas con esta información dándole un mayor entendimiento de la información, por lo tanto podemos decir que la toma de decisiones desde este punto de vista es interpretada como la deducción. Esta es la base de la arquitectura basada en la lógica.

6.6.2 Arquitecturas reactivas

Esta arquitectura se basa en rechazar las ideas de la IA simbólica y la resolución de problemas por medio de deducciones lógicas. Aquí se cree que la inteligencia se manifiesta por medio de la interacción que mantiene el agente con su medio y por lo tanto considera al medio como un elemento esencial.

Se le llama una arquitectura reactiva porque simplemente reacciona ante un estímulo externo, podríamos decir que por instinto ya que realmente no hay una serie de deducciones lógicas a realizar antes de ejecutar cierta acción.

Lo que se hace aquí es analizar situaciones, ante determinada situación se realiza determinada acción. La situación en la cual nos encontramos está dictada por los factores externos o variables externas que se encuentran en el medio.

114

Inteligencia Artificial

A este comportamiento se agrega el uso de niveles de prioridad, las acciones a realizar ante determinada situación siguen una especie de jerarquía. Existen situaciones que requieren más atención que otras o que de darse dos situaciones simultáneas las situaciones con mayor grado de prioridad se ejecutaran primero.

Vamos a dar un ejemplo sencillo, tenemos un agente robótico encargado de limpiar el piso.

La orden de mayor prioridad es siempre esquivar obstáculos.

La segunda es verificar los sensores para revisar su nivel de energía.

La tercera es revisar los sensores que detectan la necesidad de limpiar algo.

Situaciones:

Si el nivel de energía es bajo se debe regresar a la estación a recargar.

Si se está cargando algún contenido y no se ha llegado a la estación se debe seguir hacia la estación.

Si se está cargando algún contenido y ya se llegó a la estación se debe descargar el contenido.

Aquí lo que estamos haciendo en las primeras tres líneas es establecer el nivel de prioridad para las acciones principales del agente, seguido de esto se le dan instrucciones o situaciones sencillas y la acción a realizar ante estas situaciones.

115

Inteligencia Artificial

Por lo tanto en este escenario ¿qué pasa si se detecta basura en un lugar pero se tiene un nivel bajo de energía? Lo primero que se debe hacer es ir a recargarse de energía y luego ir a traer la basura.

6.6.3 Arquitectura BDI (belief, desire, intention)

Esta arquitectura tiene una base filosófica sobre la teoría de razonamiento práctico en el cual se razona paso a paso hacia una meta. Existen tres factores principales detrás de esta arquitectura.

Belief o creencia, es la base teórica sobre la cual un agente actúa. Considera toda la información que tiene almacenada como creencias, se les llama creencias ya que no se consideran como hechos porque esta información puede variar.

Desire o deseo, se puede interpretar como las acciones que el agente está planeando hacer pero que aún no se han conceptualizado. El proceso de razonamiento comienza con el análisis de las opciones que se tienen disponibles, los deseos se interpretan como opciones viables a las cuales no se ha hecho un compromiso.

Intention o intención, es el plan de ejecución o la serie de sub metas que ha decidido realizar el agente para llegar a su meta final. Luego de un análisis se toman las opciones más viables y se decide llevarlas a cabo creando así un plan de ejecución.

116

Inteligencia Artificial

Figura 62 – Páginas 117-120 Libro “Inteligencia Artificial”

El modelo BDI es utilizado fuertemente por teorías derivadas de esta arquitectura como lo son los estados mentales y las categorías mentales, utilizadas por la programación orientada a agentes.

Es importante tener la habilidad de poder abandonar una intención en determinado momento, no podemos esperar que todos los planes a los cuales se haya llegado sean los correctos. Si nos damos cuenta que una intención no puede llegar a ser realizada debe ser abandonada. Pero aquí debe haber un balance ya que si se dedica poco tiempo a seguir una intención y se desecha fácilmente nunca se llegara a realizar ningún objetivo. Viendo el otro extremo si no se abandona nunca una intención y se llega a una meta irrealizable llegamos al mismo problema.

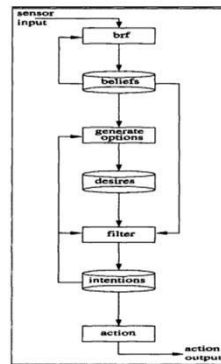


Figura No. 31 – Arquitectura BDI
Fuente: Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence, The MIT Press.

117

Inteligencia Artificial

118

Inteligencia Artificial

Podemos ver a partir de la grafica que el proceso comienza con un sistema con el cual se procesan las entradas, luego de este proceso se toma la base de creencias base y se actualiza con la información provista por el buffer de entradas.

A partir de la base de creencias actualizada procedemos a analizar las opciones que tenemos disponibles, como base tomamos tanto la base de creencias como las intenciones que se tienen actualmente. Todas estas opciones deben de ser evaluadas y eventualmente se convierten en deseos. Luego de esta etapa existe un filtro de intenciones el cual toma como base los deseos actuales, la base de creencias e incluso las intenciones actuales.

Pasado el filtro de intenciones llegamos a la etapa de intenciones establecidas. En base a las intenciones que se han desarrollado realizamos acciones para satisfacerlas. Esta es la última etapa de la arquitectura BDI.

La base de creencias, los deseos e incluso las intenciones son representados como afirmaciones lógicas. Cada uno de los pasos de transformación por ejemplo de creencias a deseos se realizan por medio de verificaciones lógicas.

Artículos relacionados

- **Web 3.0 Artificial Intelligence Agents will be conversation agents**
Valeria Maltoni
Ref: <http://www.conversationagent.com/2007/11/web-30-artifici.html>
- **Artificial Intelligence in games**
James Wexler
Ref: <http://www.cs.rochester.edu/ubrown/242/assts/termproj/games.pdf>
- **A Decision-Theoretic Model of Assistance - Evaluation, Extensions and Open Problems**
Sriram Natarajan and Kshitij Judah and Prasad Tadepalli and Alan Fern
Ref:
<http://www.alsrl.com/~nysmith/organizing/sss07/papers/SS04NatarajanS.pdf>
- **Artificial Intelligence means never having to say you're sorry**
Todd Sundsted
Ref:
<http://edition.cnn.com/TECH/computing/9809/26/gets.mart.idg/index.html>

119

Inteligencia Artificial

120

Inteligencia Artificial

Figura 63– Páginas 121-124 Libro “Inteligencia Artificial”



Glosario

IA: abreviatura para Inteligencia Artificial.

AAAI: Association for the Advancement of Artificial Intelligence, es una asociación dedicada al desarrollo e investigación de la inteligencia artificial.

Perceptron: o perceptrón en español, es el modelo de redes neuronales más sencillo.

MLP: Multi Layer Perceptron o Perceptrón Multi capa.

Web Crawler: Es un agente móvil usualmente encargado de navegar a través de la red actualizando la información para todas las páginas.

Robot: Agente artificial mecánico, es utilizado para realizar una tarea de manera automática.

BDI: Son las iniciales para Belief, Desire, Intention. El modelo BDI es una arquitectura utilizada para la implementación de agentes inteligentes.

Entropía: Es sinónimo de desorden y caos.

Cromosoma: Crepúsculos que existen en el núcleo de las células en los cuales residen los factores hereditarios.

Gen: Fragmento de ADN dispuesto en un orden fijo en los cromosomas.

Derivada: Concepto fundamental del cálculo diferencial. Es la pendiente del gráfico de una función en un punto específico.

Weight: peso en inglés, se le llama así a uno de los componentes que conforma el esquema de una red neuronal.

Bias: en inglés, una preferencia o una inclinación imparcial hacia un punto u opinión. Componente que se utiliza como una entrada fija en toda neurona artificial.



121

Inteligencia Artificial



122

Inteligencia Artificial



Referencias Bibliográficas

- L.P.J. Veelenturf, *Analysis and application of artificial neural networks*.
- Nikola K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*.
- Simon Haykin, *Neural networks a Comprehensive Foundation*.
- Gérard Dreyfus, *Neural Networks Methodology and Application*.
- Nils J. Nilsson, *Introduction to Machine Learning*.
- Samuel Orozco, *Algoritmos Genéticos*.
- Karina Gibert, Rodas, Gramajo, *AI Versus Statistics: Some common topics*.
- Adolfo Lozano Tello, *Ontologías en la web semántica*.
- Miguel Angel Abian, *El futuro de la web*.

- The MIT Press, *Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence*.
- Nils J. Nilsson, *Artificial Intelligence a new synthesis*.
- Ben Coppin, *Artificial Intelligence Illuminated*.



123

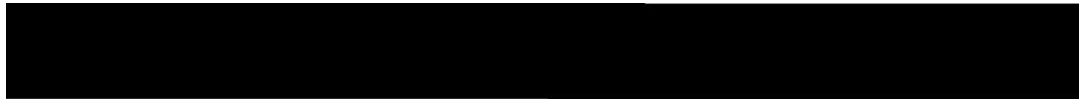
Inteligencia Artificial



124

Inteligencia Artificial

Figura 64 – Páginas 125-128 Libro “Inteligencia Artificial”



APÉNDICE A: LICENCIA

Este trabajo se encuentra definido bajo la licencia Creative Commons Attribution-ShareAlike License. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-sa/3.0/>.

A continuación se presenta un resumen de la licencia.

1.1.1 You are free:

- to **Share** — to copy, distribute and transmit the work
- to **Remix** — to adapt the work

1.1.2 Under the following conditions:

- **Attribution** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

What does “Attribute this work” mean?

The page you came from contained embedded licensing metadata, including how the creator wishes to be attributed for re-use. You can use the HTML here to cite the work. Doing so will also include metadata on your page so that others can find the original work as well.

- **Share Alike** If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.
- For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.



125

Inteligencia Artificial



126

Inteligencia Artificial



themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.

- c. **“Creative Commons Compatible License”** means a license that is listed at <http://creativecommons.org/licenses/by-sa/3.0/> and that is approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.
- d. **“Distribute”** means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
- e. **“License Elements”** means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
- f. **“Licensor”** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
- g. **“Original Author”** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
- h. **“Work”** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.
- i. **“You”** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who



127

Inteligencia Artificial

- Any of the above conditions can be waived if you get permission from the copyright holder.
- Nothing in this license impairs or restricts the author’s moral rights.

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN “AS-IS” BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE (“CCPL” OR “LICENSE”). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED. BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

- a. **“Adaptation”** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-representation with a moving image (“synchronizing”) will be considered an Adaptation for the purpose of this License.
- b. **“Collection”** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in

has received express permission from the Licensor to exercise rights under this License despite a previous violation.

- j. **“Publicly Perform”** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
 - k. **“Reproduce”** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.
2. **Fair Dealing Rights.** Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.
3. **License Grant.** Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:
- a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
 - b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked “The original work was translated from English to Spanish,” or a modification could indicate “The original work has been modified.”;
 - c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
 - d. to Distribute and Publicly Perform Adaptations.
- e. For the avoidance of doubt:
- i. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
 - ii. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

Figura 65 – Páginas 129-132 Libro “Inteligencia Artificial”

iii. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.
- b. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the “Applicable License”), you must comply with the terms of the Applicable License generally, and the following provisions: (i) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (ii) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (iii) You

must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (iv) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.

- c. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author or pseudonym, if applicable (if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution (“Attribution Parties”) in Licensor’s copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 5(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., “French translation of the Work by Original Author,” or “Screenplay based on original Work by Original Author”). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of an Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.
- d. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author’s honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action

prejudicial to the Original Author’s honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability, EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

- a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

- b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however, that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

- a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
- b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
- c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this

License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

- d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
- f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1992, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark “Creative Commons” or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons’ then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of the License. Creative Commons may be contacted at <http://creativecommons.org/>.

8. DOCUMENTACIÓN DE APOYO DE “SEMINARIO DE SISTEMAS 2”

A continuación se presenta el desarrollo de la documentación de apoyo del curso de Seminario de Sistemas 2, el cual se realizó en formato de un libro conteniendo las unidades más importantes impartidas a lo largo del curso.

8.1 Datos generales

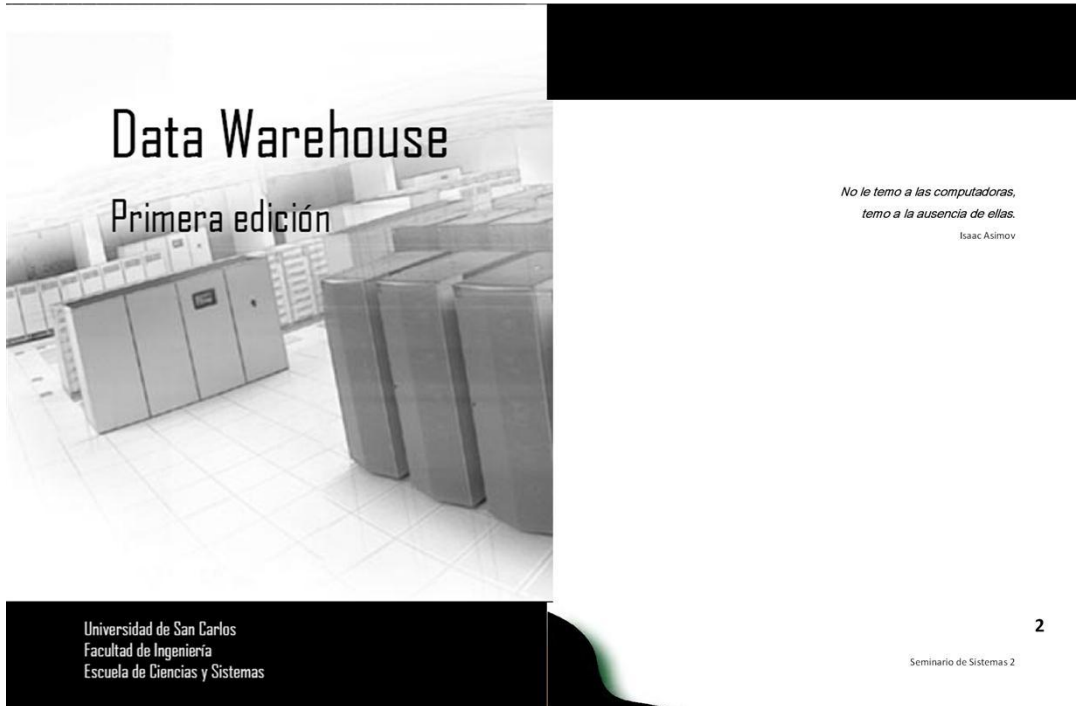
NOMBRE DE CURSO: Seminario de Sistemas 2 (798)

PRE- REQUISITOS: Seminario de Sistemas 1 (797)

POST – REQUISITOS: Seminario de Investigación (799)

OBJETIVO: Este libro está destinado para los estudiantes del curso de Seminario de Sistemas 2, conteniendo teoría, ejemplos, diagramas, resumen, glosario y artículos de interés relacionados a cada una de las unidades del curso.

Figura 66 – Páginas 1-4 Libro “Data Warehouse”



DATA WAREHOUSE

Primera Edición

OBJETIVOS: Este curso busca que el estudiante comprenda de forma práctica y sencilla la teoría básica y los conceptos más importantes al crear una base de datos para usarse como Data Warehouse.

3

Seminario de Sistemas 2

Prefacio

El área de desarrollo de bases de datos orientadas a la creación de Data Warehouse es un área importante dentro del ámbito de las bases de datos. Actualmente su rol en lo que se conoce como Business Intelligence es crucial para proveer información de gran valor para una empresa y mantenerla con una ventaja competitiva.

A lo largo de este libro se desarrolla un curso teórico sobre los conceptos, técnicas y algoritmos más importantes relacionados a la creación de un Data Warehouse.

Tiene un enfoque genérico en cuanto a la herramienta a utilizar se refiere, los conceptos se dan de una manera genérica con el objetivo de servir como base en la creación de un Data Warehouse sin importar que herramienta se utilice.

4

Seminario de Sistemas 2

Figura 67 – Páginas 5-8 Libro “Data Warehouse”

Prerrequisitos

Se debe tener un conocimiento básico sobre estructuras de datos y bases de datos. A pesar de nombrar algunas técnicas que pertenecen al área de Inteligencia Artificial en el último capítulo no es necesario tener ningún conocimiento sobre esta área.

CONTENIDO

1. Capítulo 1 – Data Warehouse	08
A. Alcance	09
1.1 Conceptos Básicos	09
1.1.1 OLAP	10
1.1.2 ROLAP y MOLAP	11
1.2 Data Warehouse y Data Mart	12
1.2.1 Data Warehouse	12
1.2.2 Data Mart	13
1.3 Hardware y Software	13
1.4 Business Intelligence	14
B. Artículos relacionados	15

5

Seminario de Sistemas 2

2. Capítulo 2 – Diseño	17
A. Alcance	18
2.1 Esquema base de un Data Warehouse	18
2.1.1 Modelo Estrella	19
2.1.2 Modelo Snowflake	20
2.2 Componentes de un Data Warehouse	21
2.2.1 Fact Table	22
2.2.2 Dimensiones	22
2.2.3 Niveles	24
2.2.4 Métricas	25
B. Artículos relacionados	26
3. Capítulo 3 – ETL y DSS	28
A. Alcance	29
3.1 Proceso de ETL	29
3.1.1 Extracción	30
3.1.2 Transformación	31
3.1.2.1 Surrogate keys	33
3.1.3 Carga	34
3.1.3.1 Automatización	37
3.2 DSS	37
3.2.1 Usar el DSS	37
3.2.2 Reportes	39
B. Artículos relacionados	40
4. Capítulo 4 – Data Mining	42
A. Alcance	43
4.1 Definición	43

6

Seminario de Sistemas 2

4.2 Aprendizaje Supervisado	44
4.2.1 Clasificación	44
4.2.2 Modelos de regresión	45
4.3 Aprendizaje no Supervisado	46
4.3.1 Reglas de asociación	46
4.3.2 Clustering	47
4.3.3 Extracción de atributos	48
B. Artículos relacionados	49
I. Glosario	51
II. Referencias Bibliográficas	53
III. Apéndice A: Licencia	54

7

Seminario de Sistemas 2

ILUSTRACIONES

I. Figuras	
Figura No. 1 – Modelo Estrella	19
Figura No. 2 – Modelo Snowflake	20
Figura No. 3 – Dimensión en niveles	23
Figura No. 4 – Dimensión	24
Figura No. 5 – ETL	30
Figura No. 6 – ETL Complejo	35
Figura No. 7 – DSS	38

Capítulo 1

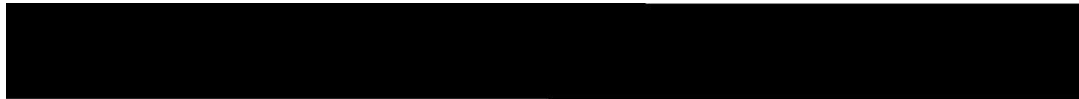
DATA WAREHOUSE

Este capítulo presentamos el concepto general de Data Warehouse y una serie de términos que se utilizan comúnmente. Explicamos el tema de Business Intelligence y también varios conceptos relacionados con este proceso.

8

Seminario de Sistemas 2

Figura 68 – Páginas 9-12 Libro “Data Warehouse”



ALCANCE

- 1. Data Warehouse
 - 1.1. Conceptos Básicos
 - 1.1.1. OLAP
 - 1.1.2. ROLAP y MOLAP
 - 1.2. Data Warehouse y Data Mart
 - 1.2.1. Data Warehouse
 - 1.2.2. Data Mart
 - 1.3. Hardware y Software
 - 1.4. Business Intelligence

1.1 Conceptos Básicos

Comenzamos el capítulo explicando algunos términos básicos que se manejan en este tema. Debemos primero tener claro que un sistema de Data Warehouse es diferente a un sistema de bases de datos normal no solamente en su objetivo sino en su funcionamiento interno.

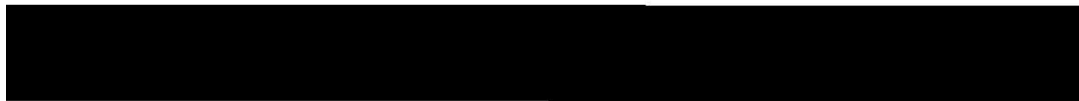
1.1.1 OLAP y OLTP

Una base de datos que va a ser utilizada como Data Warehouse tiene un sistema de almacenamiento de información diferente al que se utiliza con una base de datos relacional o con un modelo relacional.

Para optimizar el proceso completo que trae la realización de un Data Warehouse debemos utilizar una base de datos OLAP (Online analytical processing). La opción para soportar OLAP es usualmente una opción extra en la mayoría de bases de datos y debe ser activada por separado. Una base de datos OLAP trabaja de manera distinta a una base de datos relacional y por esta razón debemos tener claro cuál es la finalidad de la base de datos antes de activar esta opción. Las bases de datos que se manejan más comúnmente son las OLTP, aquí la T es por Transaccional o transaccional. Estas dos iniciales A y T nos dicen cual es la prioridad de la base de datos y lo que es más importante que trabajan de maneras muy diferentes.

El modelo OLAP nos provee un sistema de almacenamiento especializado y un modelo de análisis optimizado para consultas sobre los datos. En una base de datos relacional lo que se está optimizando es el espacio que ocupa la información por esta razón se deben normalizar los datos. Cuando usamos el modelo OLAP se tiene otro objetivo en mente y no se pone mucho énfasis en el espacio que se utiliza en el disco.

Por ejemplo nosotros estamos realizando una consulta sobre la base de datos y estamos buscando por ejemplo el promedio de



ventas de cada empleado. Si estamos usando un sistema relacional de base de datos buscar este tipo de información requiere varias pasadas sobre los datos usando el modelo OLAP el tiempo de respuesta es mucho menos ya que aquí la información esta optimizada para realizar consultas.

1.1.2 ROLAP y MOLAP

En general existen dos enfoques principales para lograr el modelo OLAP.

La primera opción que se nos da son las bases de datos ROLAP, la primera letra que se antepone la R es por la palabra Relacional. Con las bases de datos ROLAP estamos utilizando todavía un modelo relacional para trabajar con los datos, por esta razón se utiliza SQL para manipular los datos.

Luego tenemos las bases de datos MOLAP, la M es por multidimensional. En las bases de datos MOLAP ya no estamos usando el modelo relacional para almacenar los datos, aquí estamos usando un modelo multidimensional. El tiempo de respuesta ante consultas es mucho menor a una base de datos ROLAP porque los datos están replicados en diferentes lugares o lo que llamamos dimensiones. Otra de las grandes diferencias que existe en este modelo es la utilización de MDX como lenguaje de manipulación para los datos. El mayor inconveniente que se tiene usando este modelo es que cargar la base de datos tarda un poco

más de tiempo por la necesidad de conversión de los datos, y por lo tanto la información no está actualizada en todo momento. De hecho en algunos casos el proceso de actualizar el Data Warehouse se puede volver una labor complicada, esto lo determina principalmente el numero de fuentes de información que se tienen y la manera en que se mantiene la información almacenada.

1.2 Data Warehouse y Data Mart

1.2.1 Data Warehouse

Un Data Warehouse es utilizado para un propósito general, obtener datos y tendencias de manera muy rápida basándonos en datos históricos de una empresa.

Cuando hablamos de datos históricos nos referimos a datos de al menos unos 10 años atrás, ya que es de esta manera que trabaja un Data Warehouse. Podemos afirmar que mientras más antiguos sean los datos (manteniendo siempre la continuidad) más efectivo será el análisis de los datos.

Si no se tienen suficientes datos históricos se pierde efectividad utilizando el sistema. En el caso de realizar modelos para predecir el comportamiento futuro de los datos la importancia de tener datos históricos es aun más grande.



Figura 69 – Páginas 13-16 Libro “Data Warehouse”

1.2.2 Data Mart

Un Data Mart es una porción del Data Warehouse completo, que puede funcionar independientemente. Es un subconjunto del conjunto completo que es el DW y este subconjunto está enfocado a un área específica. El único problema es que no se tienen todos los datos, pero se puede trabajar con el de la misma forma que con un Data Warehouse.

Este tipo de enfoque es muy común cuando se tienen varias sucursales grandes ya que cada sucursal usualmente trabaja con los datos locales, únicamente los cargos más altos trabajan con la totalidad de los datos.

1.3 Hardware y Software

El hardware es una de las características más importantes a la hora de implementar nuestro sistema de Data Warehouse. Debemos tener en cuenta que vamos a estar almacenando grandes cantidades de información, y también debemos hacer un estimado de la carga de usuario que esperamos tener sobre la base de datos.

En la mayoría de los casos lo más conveniente es hacer un cluster de computadoras para soportar la carga de trabajo necesaria. También debemos tomar consideraciones en cuanto a la disponibilidad que va a requerir el sistema, es posible que se tenga que implementar un sistema RAID para protección de los datos.

1.4 Business Intelligence

El concepto de business intelligence se refiere a la ventaja competitiva que logra una empresa al tener la información más actualizada y más rápida que le permite tomar decisiones cruciales sobre su negocio.

La clase de información que se puede tener con un buen sistema de Bi (Business Intelligence) puede influir en todos los aspectos de la estrategia de una empresa.

Los niveles altos de la gerencia utilizan estas herramientas para obtener información variada, por ejemplo tomemos el caso de una empresa de venta de ropa.

Con un sistema de Bi podemos averiguar datos como:

Que ropa se vende más.

Que sucursales venden más.

Quien es el mejor vendedor.

Que proveedores no nos están sirviendo.

Por qué un mes se vende más que otro.

Y otros tipos de datos que normalmente obtenemos por medio de reportes, la diferencia aquí es que estos datos se obtienen de manera dinámica y en cuestión de segundos.

Pero tal vez la herramienta más poderosa está en la predicción basada en modelos de regresión. Podemos predecir cómo serán las ventas para el siguiente año, que prendas se venderán más, etc. Y este es el tipo de información que le da una

13

Seminario de Sistemas 2

14

Seminario de Sistemas 2

ventaja competitiva a una empresa, influyendo en decisiones importantes.

Existen otras herramientas que se utilizan para realizar operaciones similares a esta, este tema se desarrolla más en el capítulo dedicado a Data Mining.

- **Panning Data Gold: A guide to information management**

Linda More

Ref: <http://www.computing.co.uk/computing/features/2233359/panning-gold>

- **Business Intelligence gets smarter**

Alice Dragoon

Ref:

http://www.cio.com/article/296999/Business_Intelligence_Gets_Smart_er_

Artículos relacionados

- **Business Intelligence not just for bosses anymore**

Meridith Levinson

Ref:

http://www.cio.com/article/16544/Business_Intelligence_Not_Just_for_Bosses_Anymore

- **The Green Data Warehouse**

Rick Abbot

Ref: <http://www.b-eye-network.com/view/9409>

15

Seminario de Sistemas 2

16

Seminario de Sistemas 2

Figura 70 – Páginas 17-20 Libro “Data Warehouse”



Capítulo 2

DISEÑO

Comenzamos este capítulo explicando los diferentes esquemas que se utilizan como base para nuestro Data Warehouse. Luego de establecer un esquema podemos proceder a explicar a la estructura básica, explicamos conceptos como fact table, dimensiones, niveles y métricas.

ALCANCE

- 2. Diseño
 - 2.1. Esquema base de un Data Warehouse
 - 2.1.1. Modelo Estrella
 - 2.1.2. Modelo Snowflake
 - 2.2. Componentes de un Data Warehouse
 - 2.2.1. Fact Table
 - 2.2.2. Dimensiones
 - 2.2.3. Niveles
 - 2.2.4. Métricas

2.1 Esquema Base de un Data Warehouse

Existe una serie de consideraciones a tomar en cuenta al realizar el diseño de nuestro Data Warehouse, lo primero que debemos saber es el esquema que vamos a utilizar. Dependiendo del modelo inicial que vamos a escoger el número de pasos al formar el Data Warehouse cambia.



17

Seminario de Sistemas 2



18

Seminario de Sistemas 2



Debemos notar en este punto que el modelado para un Data Warehouse es diferente al modelado Entidad Relación utilizado en bases de datos transaccionales. Aquí se utiliza lo que se llama el Modelado Dimensional que está basado en dimensiones y la Fact Table.

En general existen dos esquemas con los cuales trabaja un modelo de Data Warehouse, tenemos el modelo estrella y el modelo Snowflake.

2.1.1 Modelo estrella

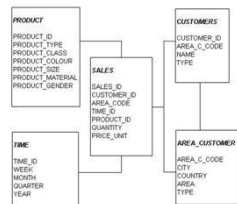


Figura No. 1 – Modelo Estrella.

Fuente: http://www.pafis.shh.fi/~argarg03/SF15/WS1/sales_star.jpg



19

Seminario de Sistemas 2

En este tipo de modelo la información se almacena en dimensiones en las cuales todos los datos están des normalizados. Este modelo es el que más se utiliza, y en algunas herramientas de Data Warehouse la única opción que se tiene.

Se recomienda tener a este modelo siempre como primera opción ya que para la mayoría de los casos es el modelo que mejor desempeño muestra.

2.1.2 Modelo Snowflake

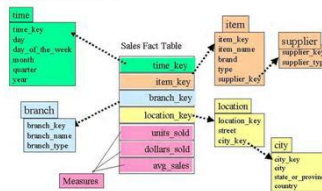


Figura No. 2 – Modelo Snowflake.

Fuente: http://www.bipminstitute.com/dipm_images/DB2-%20Snow%20flake-Schema.jpg



20

Seminario de Sistemas 2

Figura 71 – Páginas 21-24 Libro “Data Warehouse”

En este modelo las dimensiones, a diferencia del modelo estrella, sí están normalizadas. A pesar de esto el objetivo de esta base de datos sigue siendo el mismo. Este modelo no es muy popular entre desarrolladores ya que en cierta manera se aproxima más a un modelo OLTP y se aleja de la teoría del diseño multidimensional.

En general se recomienda usar el modelo estrella a menos que exista alguna razón concreta para escoger este modelo. Por ejemplo si tenemos problemas de almacenamiento y una de las dimensiones en concreto ocupa mucho espacio podemos considerar normalizar un nivel o más para facilitar el almacenamiento. Otro caso que no es muy común es cuando se tiene información que ocupa muchos registros y que no se utiliza mucho. Si normalizamos esta dimensión con esta información en concreto podemos obtener un incremento en el desempeño de las consultas que no buscan esta información extra, pero para las consultas que sí buscan esta información el tiempo de respuesta baja por lo que tenemos que tener cuidado al hacer esto.

2.2 Componentes de un Data Warehouse

Existe una serie de componentes que conforman todo Data Warehouse sin importar el esquema que se vaya a utilizar. Estos

elementos son esenciales por lo que debemos familiarizarnos con ellos.

2.2.1 Fact Table

Es la tabla en la cual se centraliza la información, aquí es importante notar que prácticamente solo se manejan datos numéricos y en mayor parte llaves primarias que son unidas a otras dimensiones.

Esta es la tabla que más datos contiene en todo nuestro modelo ya que es donde se unifica toda la información. Ya que aquí está contenida toda la información es importante analizar en qué grado de detalle queremos que dicha información esté allí, a este concepto se le llama la Granularidad. La Fact Table por lo tanto se apoya delegando a las dimensiones la información más detallada que para un análisis numérico no es utilizada inmediatamente, aunque para consultas más específicas esta información debe estar disponible.

2.2.2 Dimensiones

Cuando hablamos de dimensiones, estamos hablando de elementos de datos que usualmente están conformados por varios niveles. Como habíamos dicho anteriormente, los datos están desnormalizados para ahorrar tiempo, es por esto que se crean las dimensiones. Una manera fácil de separar la información que debe ir en la Fact Table y la que debe ir en una Dimensión es observando

21

Seminario de Sistemas 2

22

Seminario de Sistemas 2

la información que no cambia. En general en las dimensiones se guarda información estática y en la Fact Table información que cambia constantemente.

Un ejemplo claro de la creación de una dimensión puede ser este modelo sencillo:



Figura No. 3 – Dimensión en niveles

Tenemos cuatro tablas, ninguna de estas entidades tiene razón de existir por sí sola realmente, por esta razón vamos a crear la dimensión Especialidad.

Cabe notar que los códigos de las tablas que fusionamos no se incluyen, solamente los campos que contienen datos. En general la manera en que creamos nuestras dimensiones está dirigida a optimizar el modelo multidimensional.



Figura No. 4 – Dimensión

Este es el resultado de la creación de la dimensión basada en el conjunto de tablas utilizadas en la imagen número 3, aquí estamos creando una dimensión pensando en un esquema estrella.

Cuando tenemos un modelo relacional a partir del cual vamos a crear un Data Warehouse es una buena práctica de diseño aislar las tablas que están relacionadas y considerarlas como posibles dimensiones. Luego de esto incluso nos resulta más fácil ver cuáles tablas conforman la Fact Table también.

2.2.3 Niveles

Los niveles como podemos intuir de la imagen anterior, son la jerarquía en la cual se puede configurar la dimensión. Al hablar de jerarquía estamos diciendo que una serie de especialidades están contenidas en un hospital y que una serie de hospitales están contenidas en un departamento, etc. Es básicamente la estructura establecida en el modelo entidad relación pero de una manera más compacta.

23

Seminario de Sistemas 2

24

Seminario de Sistemas 2

Figura 72 – Páginas 25-28 Libro “Data Warehouse”



Para la dimensión Especialidad los niveles son:
Región
Departamento
Hospital
Especialidad

ocasiones son valores que se tienen directamente en la base de datos, debemos saber que todas las métricas que se utilizan se guardan en la fact table como parte de la información en esta tabla.

En esta etapa de la construcción del Data Warehouse no es tan fácil apreciar cual es la utilidad de establecer niveles en las dimensiones, los niveles que establecemos realmente los utilizamos durante la etapa de DSS en la que buscamos datos concretos manipulando las dimensiones. Este tipo de agrupaciones puede ayudarnos a volver esta etapa en un proceso más intuitivo.

2.2.4 Métricas

La combinación de las métricas adecuadas con las dimensiones necesarias nos dan los datos que necesitamos.

Ejemplos usuales de métricas pueden ser: # de ventas, precio, utilidad, ganancia, etc. En general valores que son relevantes al hacer consultas.

Las métricas usualmente son valores que no se tienen directamente en la Fact Table, estos valores son calculados a partir de la información que se tiene allí ya sea por medio de conteos o por medio de la creación de promedios u otros medios. En otras

Artículos relacionados

- **Fact Tables**
Tod McKenna
Ref: <http://blog.todmeansfox.com/2007/08/27/fact-tables/>
- **Identifying Fact Tables**
Craig Borysowich
Ref: <http://it.toolbox.com/blogs/enterprise-solutions/identifying-fact-tables-data-warehouse-20810>



25

Seminario de Sistemas 2



26

Seminario de Sistemas 2



- **Slowly Changing Dimensions Are Not Always as Easy as 1, 2, 3**
Margy Ross y Ralph Kimball
Ref: <http://www.intelligententerprise.com/showArticle.jhtml?articleID=59301280>
- **Best Practices for Business Intelligence Using the Microsoft Data Warehousing Framework**
Microsoft Staff Writers
Ref: [http://msdn.microsoft.com/en-us/library/aa902663\(SQL.80\).aspx](http://msdn.microsoft.com/en-us/library/aa902663(SQL.80).aspx)

Capítulo 3

ETL Y DSS

En este capítulo se ven dos de los últimos procesos que se realizan para utilizar el Data Warehouse. Durante el proceso de ETL se realiza la carga de datos y en esta sección detallamos el proceso completo. Como segundo tema principal se tiene la utilización de un DSS para navegar a través de los datos.



27

Seminario de Sistemas 2



28

Seminario de Sistemas 2

Figura 73 – Páginas 29-32 Libro “Data Warehouse”

ALCANCE

- 3. ETL y DSS
 - 3.1. Proceso de ETL
 - 3.1.1. Extracción
 - 3.1.2. Transformación
 - 3.1.2.1 Surrogate keys
 - 3.1.3. Carga
 - 3.1.3.1 Automatización
 - 3.2. DSS
 - 3.2.1. Usar el DSS
 - 3.2.2. Reportes

3.1 Proceso de ETL

El proceso de ETL tiene como objetivo cargar la información que vamos a utilizar a nuestro Data Warehouse. Consiste básicamente de tres fases. Tenemos la fase de Extracción o extracción, la fase de Transformation o transformación y la fase de Load o carga. A este proceso también se le conoce como ETT (extraction, transportation and transformation), extracción, transportación y transformación.

29

Seminario de Sistemas 2

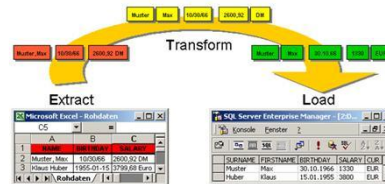


Figura No. 5 – ETL

Fuente: <http://www.et-tool.de/>

De esta imagen podemos extraer la idea principal del proceso de ETL, en este ejemplo los datos están en una hoja de Excel de la cual escogemos solo algunas columnas. A estas columnas le hacemos una serie de cambios para pasar a la siguiente etapa. Finalmente los datos ordenados son cargados a la base de datos. Cada una de estas etapas involucra una serie de detalles que debemos de tomar en cuenta, vamos a desarrollar cada uno de estos pasos.

3.1.1 Extracción

La fase de extracción es el primer paso que debemos tomar. Este proceso se puede llegar a complicar dependiendo del número

30

Seminario de Sistemas 2

de fuentes de información diferentes que se tengan, también es importante el formato en que se encuentra dicha información.

Por ejemplo podríamos tener una gran parte de la información en hojas de Excel, otra parte en una base de datos Oracle y otra parte en una base de datos que no es relacional. No importa de donde venga la información, este proceso debe ser capaz de extraer la misma en un formato que pueda ser utilizado.

En algunos casos la información que vamos a necesitar se encuentra en un sistema en producción, en la mayoría de los casos no se tiene mucho tiempo para trabajar con esta base de datos. Lo más probable es que se genere un archivo de salida en un formato plano para exportar toda la información.

3.1.2 Transformación

Una vez que la información ha sido extraída de las fuentes necesarias podemos pasar a esta fase. Aquí se realiza una serie de transformaciones con la información cruda para llegar a un formato que pueda ser cargado al data warehouse. Aquí no solamente estamos hablando de un cambio de formato sino también de una reestructuración de la misma ya que la información contenida en las dimensiones y en la fact table se guardan de una manera particular. Existe muchas maneras en las cuales se puede transformar la información antes de ser ingresada al data warehouse.

31

Seminario de Sistemas 2

Durante esta fase, como parte de la información a cargar se puede también incluir valores calculados que pueden ser de gran utilidad para hacer análisis posteriores. Por ejemplo si tenemos una tabla que maneja productos y queremos incluir el precio en dólares en la tasa que se manejaba en ese momento, se puede incluir también como parte de los datos principales. También es posible que no todos los datos nos sean útiles para el Data Warehouse y o por esta razón sea necesario seleccionar únicamente algunas columnas de toda la información.

Es posible que algunos de los datos estén muy detallados y no necesitemos este nivel de exactitud, en todo caso es posible realizar proyecciones sobre los datos y seleccionar únicamente algunos registros, es muy similar a seleccionar únicamente ciertas columnas pero aquí se realiza en el sentido contrario.

Es posible que la forma en que se presenta la información esté dividida en varias secciones y en algunos casos es más fácil trabajar con dicha información si esta se fusiona en una sola columna, como un dato atómico. Se puede dar el caso contrario, tener una columna con varios datos importantes y puede ser necesario que nosotros tengamos que dividir esta información en dos o más columnas para facilitar su manejo.

En algunos casos por ejemplo para designar el sexo de una persona se utiliza ya sea la palabra completa o las primeras cuatro letras, (Masculino, MASC, Femenino, FEM), para ahorrar espacio es posible que nosotros tengamos que transformar esta información únicamente a una letra (M y F). También es posible que alguna de

32

Seminario de Sistemas 2

Figura 74 – Páginas 33-36 Libro “Data Warehouse”

la información que se incluye tenga que ser validada de alguna manera antes de ser incluida en el Data Warehouse, de no ser aceptada dicha información simplemente no se carga.

Estos son algunos ejemplos de cómo se realizan transformaciones con la información, esto es muy común durante esta fase pero no es necesariamente obligatorio durante esta fase. Si los datos con los cuales trabajamos están bien, simplemente nos limitamos a crear consultas para cargar varias tablas en las diferentes dimensiones que hemos diseñado. Es importante notar que durante esta etapa se realizan las agrupaciones de información que dan lugar a las dimensiones, aunque este tipo de fusiones puede llegar a darnos algunos problemas.

3.1.2.1 Surrogate keys

El concepto de Surrogate keys, o llaves sustitutas nace de la necesidad de reestructurar información que antes se manejaba de manera individual. La idea es básicamente volver a asignar llaves primarias a los datos que estamos manejando.

Por ejemplo se puede dar el caso de que una empresa que no tiene una política de mantener datos históricos en la base de datos de producción más allá de algunos años. En este tipo de empresas se puede dar el caso de que reutilicen llaves primarias, por ejemplo se maneja una serie de productos y una vez que un producto se elimina su llave primaria se reutiliza. Esto nos puede causar serios

problemas cuando cargamos la dimensión de productos con todos los productos que han existido.

Una alternativa a este caso es reestructurar las llaves primarias una vez que se carga la información, otra alternativa puede ser fusionar la llave primaria con la fecha de creación creando una nueva llave primaria más larga. En general lo que se hace es reasignar el identificador primario de los elementos que manejamos.

3.1.3 Carga

Luego de la fase de transformación la información ya se encuentra en el formato adecuado y listo para ser cargado hacia las dimensiones y la fact table que hemos diseñado.

El proceso de carga a un Data Warehouse puede llegar a ser una labor bastante laboriosa y debemos considerar con qué frecuencia queremos invertir este tiempo en mantener el data warehouse actualizado. Debemos tomar en cuenta que la información con la cual trabajamos no puede estar actualizada en todo momento, porque la información usualmente viene de varios lugares diferentes y es en sus fuentes propias donde esta es actualizada.

Como habíamos explicado anteriormente las bases de datos OLAP están optimizadas para realizar consultas sobre ellas, por lo tanto la carga de datos se tarda más tiempo que en una base de datos OLTP. Se tiene también el problema agregado que

dependiendo la manera en que está diseñado el sistema en algunos casos debemos utilizar MDX para poder realizar estas actualizaciones. Por esta razón en algunos casos resulta más factible eliminar la información completa del Data Warehouse y cargar toda la información nuevamente cada vez que se tenga que actualizar, siempre se debe realizar un análisis para saber qué decisión resulta más conveniente.

Esta es una imagen del proceso de ETL más completa. Comenzamos con la fuente de los datos, si estos se encuentran en un sistema operacional se debe pasar primero a un archivo de salida. Reservamos también espacio para otras posibles fuentes de datos en la primera capa.

Luego de esto presentamos lo que se llama el área de preparación o Staging Area. Este recurso se utiliza en algunas ocasiones cuando la transformación de los datos es muy complicada y agregar un área de preparación facilita el proceso. En esta área tenemos una base de datos de un tamaño similar a la base de datos final del Data Warehouse. Es común que aquí se tenga información aún normalizada, también es posible que aún contenga la información con las llaves primarias originales.

Debemos tener claro que esta área nunca va a ser visible para los usuarios finales, como lo indica su nombre contiene información en un estado intermedio de transformación. En esta área también se realizan labores de limpieza de información y en algunos casos desecho de registros que no cumplen con ciertos requerimientos.

Luego las preparaciones finales que se realizan sobre la información son para preparar y ajustar los datos en el formato apropiado para ingresarse después a las dimensiones y a la fact table.

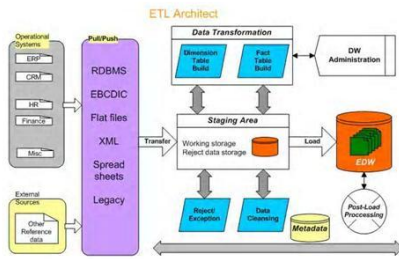


Figura No. 6 – ETL complejo
Fuente: <http://www.dwmantra.com/dwconcepts.html>

Figura 75 – Páginas 37-40 Libro “Data Warehouse”

3.1.3.1 Automatización

Es importante que el proceso completo de ETL sea un proceso automático porque se debe realizar con cierta frecuencia y en algunos casos en horarios donde no se tiene mucha carga de usuarios. En un sistema completamente autónomo se puede incluso programar un horario fijo para que el sistema actualice la información sin intervención alguna periódicamente.

3.2 DSS

El sistema DSS (Decision Support System) o sistema de soporte para toma de decisiones es en algunos casos la última etapa del Data Warehouse. Un DSS es una interfaz amigable por medio de la cual podemos obtener la información que necesitamos del Data Warehouse. Dependiendo de la base de datos que se está utilizando se tiene una serie de opciones de diferentes DSS disponibles, por ejemplo para la base de datos SQL es común utilizar Excel como DSS para navegar a través de la información.

3.2.1 Usar el DSS

Sin importar que DSS se está utilizando la manera de utilizarlo es siempre la misma, lo que estamos buscando es que combinación de dimensiones y métricas necesitamos para encontrar la información que queremos.

Comenzamos tomando una dimensión, y luego tomamos otra dimensión que usualmente es la dimensión tiempo ya que es la que más se utiliza como segunda dimensión. Luego de esto necesitamos escoger una métrica con la cual unir las dos dimensiones. Vamos a ver un ejemplo realizado en SQL Server:

En esta imagen estamos analizando cuantos empleados han sido empleados en cada departamento durante varios años. Por lo tanto como dimensión principal escogemos una dimensión llamada Empleados. Como dimensión secundaria tenemos la dimensión tiempo. Ahora lo que escogemos como métrica es un Conteo de Empleados.

		Department			
		Year			
Year	Métrica	1	2	3	4
		Count Employee ID	Count Employee ID	Count Employee ID	Count Employee ID
1998	Departamento	1	1		
1998	Departamento	1			1
1998	Departamento	2			
1998	Departamento	4			
1998	Departamento	5			
1998	Departamento	10			
1998	Total	138	5		1
1999	Departamento	20			2
2000	Departamento	20			2
2001	Departamento	1			1
2002	Departamento				1
2003	Departamento				1
Grand Total		180	6		13

Figura No. 7 – DSS

Cuando queremos realizar alguna otra consulta a la información simplemente cambiamos las dimensiones que estamos

37

Seminario de Sistemas 2

38

Seminario de Sistemas 2

utilizando y si es necesario las métricas también. En general el proceso de DSS se puede reducir a esto, claro una persona que no está familiarizada con los datos no sabrá qué hacer con ellos.

Debemos comprender bien las métricas que estamos utilizando porque si las intentamos asociar a las dimensiones incorrectas es posible que nos den valores erróneos. En algunos casos algunas métricas no nos dan ningún valor hasta que sean asociadas a las dimensiones correctas. Por estas razones es importante saber interpretarlas.

Como podemos ver el proceso para obtener información es bastante intuitiva y no requiere mayor conocimiento de bases de datos para poder utilizarlo. Si se está familiarizado con el objetivo de la entidad que utiliza la base de datos se puede usar el DSS para obtener la información que se necesita.

3.2.2 Reportes

Debemos tener en cuenta que el objetivo de un Data Warehouse es la generación de reportes instantáneos, a pesar de que la mayoría de los casos lo que se busca es información inmediata en algunos casos vamos a necesitar generar reportes formales de la información que estamos analizando.

Para toda base de datos que maneja un Data Warehouse existen siempre un módulo o una utilidad extra para generar este tipo de reportes. Este tipo de programas funcionan de la misma

manera en que lo hacen generadores de reportes para bases de datos OLTP, se manejan las mismas variables de establecer un formato y lo que cambia aquí es la fuente de información que especificamos.

Artículos relacionados

- Additional thoughts: ETL and ELT, RDBMS**
 Dan Linstedt
 Ref: <http://www.teradata.com/t/page/127099/index.html>
<http://www.teradata.com/t/page/128181/index.html>
- ETL Tool selection for Data Warehousing**
 David Haertzen
 Ref: http://www.infogoal.com/datawarehousing/etl_tool_select.htm
- Data Warehouse and the ETL Tool**
 Art Taylor
 Ref: <http://www.tdan.com/view-articles/5259>

39

Seminario de Sistemas 2

40

Seminario de Sistemas 2

Figura 76 – Páginas 41-44 Libro “Data Warehouse”

- *Tomorrow's real time ETL Imperative*
Dan Linstedt
Ref: <http://www.teradata.com/?page=127085/index.html>

Capítulo 4

DATA MINING

En este capítulo explicamos una de las herramientas más útiles de todo Data Warehouse, explicamos el concepto de data mining y enumeramos algunas técnicas utilizadas para este fin. Dividimos estas técnicas entre aprendizaje supervisado y aprendizaje no supervisado.

41

Seminario de Sistemas 2

42

Seminario de Sistemas 2

ALCANCE

- 4. Data Mining
 - 4.1. Definición
 - 4.2. Aprendizaje Supervisado
 - 4.2.1. Clasificación
 - 4.2.2. Modelos de regresión
 - 4.3. Aprendizaje no Supervisado
 - 4.3.1. Reglas de asociación
 - 4.3.2. Clustering
 - 4.3.3. Extracción de atributos

4.1 Definición

Este es un tema que involucra las áreas de inteligencia artificial y la estadística, el objetivo es encontrar tendencias escondidas en la información. Durante el proceso de DSS nosotros movemos dimensiones y escogemos métricas para asociar la información que nosotros queremos ver. Durante el proceso de data mining no se busca una dimensión específica, estamos buscando entre varias fuentes de información y utilizando algunos parámetros base buscamos algún tipo de relación entre los datos.

43

Seminario de Sistemas 2

Podemos explicar esto con ejemplos más concretos. Durante la etapa de DSS podemos responder a preguntas como: ¿Quién ha vendido más este semestre? ¿Qué tienda me da las mayores ganancias? ¿Qué producto no se vende? Pero en la etapa de data mining podemos responder a preguntas como ¿Qué producto venderá más si es puesto en oferta? ¿Será buena idea comenzar una promoción de ventas? ¿Qué lugar sería apropiado para crear una nueva sucursal?

Existen dos grandes grupos bajo los cuales podemos clasificar las técnicas asociadas a data mining, para cada una de estas clasificaciones mostramos algunos algoritmos que caen bajo esta categoría.

4.2 Aprendizaje Supervisado

Tenemos primero el aprendizaje supervisado o dirigido, aquí tenemos una idea de lo que buscamos pero no sabemos realmente como llegar a la información. Por ejemplo cuando buscamos responder a las preguntas propuestas en el párrafo anterior, estamos buscando una solución por medio del aprendizaje supervisado.

4.2.1 Clasificación

La técnica de clasificación es muy similar a la técnica de clustering ya que ambas toman grandes cantidades de información y observando sus características las dividen en una serie de

44

Seminario de Sistemas 2

Figura 77 – Páginas 45-48 Libro “Data Warehouse”

agrupaciones o categorías. La diferencia aquí es que las agrupaciones las definimos nosotros, ya sabemos de antemano cuales son las agrupaciones que queremos pero no las variables que definen a qué grupo pertenecen.

Por ejemplo manejamos una empresa de seguros y queremos dividir a nuestros clientes en 3 clases preferenciales diferentes, la gran cantidad de variables que se manejan por persona hace que esta tarea sea bastante complicada si se quiere realizar manualmente. Por medio de la técnica de clasificación, una vez que hemos definido las agrupaciones en las cuales vamos a clasificar a los clientes, podemos solucionar este problema.

4.2.2 Modelos de regresión

Los modelos de regresión son básicamente técnicas de predicción, se utiliza la información histórica almacenada para dar un estimado bastante acertado del futuro comportamiento de la información.

El modelo más sencillo de regresión que podemos crear es el modelo de regresión lineal, se utiliza como base la ecuación de la línea ($y = mx + b$) y luego se intenta resolver las variables m y b para ajustar la línea a los datos.

Podemos crear también modelos más complejos que involucren más de una variable independiente, estos modelos son llamados modelos de regresión múltiples. En algunas ocasiones los modelos que involucran dos variables se les llama modelos de

regresión cuadrática. Este tipo de modelo es más difícil de armar pero podemos crear un ajuste más exacto hacia los datos.

Para ambos casos lo que se está haciendo es que se crea un modelo matemático que se ajuste a los datos que tenemos. El objetivo final de crear este modelo matemático es observar cómo se comporta la gráfica para valores de entrada que aun no han sido resueltos, de esta manera podemos predecir el comportamiento futuro de los datos.

4.3 Aprendizaje no supervisado

En la segunda clasificación tenemos el aprendizaje no supervisado o no direccionado. Se le llama así ya que no se tiene una premisa bajo la cual trabajar, lo único que tenemos es un grupo de datos y lo que buscamos es algún tipo de asociación entre los mismos que sea de valor. Podríamos considerar a este enfoque como minería de datos pura.

4.3.1 Reglas de asociación

Esta técnica está basada en encontrar relación entre dos artículos, en la mayoría de los casos se emplea este conocimiento en el área de ventas.

La mayor parte de nosotros hemos sido víctimas de esta técnica y tal vez no nos hayamos dado cuenta. Cuando nosotros entramos a un sitio y este tiene la capacidad de guardar un record

45

Seminario de Sistemas 2

46

Seminario de Sistemas 2

de las acciones que realizamos esta información se guardan en una base de datos. Basándose en ella el sitio nos da una serie de sugerencias, por ejemplo si compramos una película de acción es posible que nos recomiende otra película similar la próxima vez que entremos.

De esta manera por medio de reglas de asociación el sitio nos da sugerencias con una probabilidad alta de aceptación a diferencia de los anuncios enviados al público en general. A esta técnica también se le llama análisis basado en el mercado.

La manera en que este proceso funciona es por medio de la asociación de dos elementos. Siguiendo el caso que se da en sitios donde venden productos, si un usuario compra determinado objeto esta acción se guarda para luego encontrar los elementos que se relacionan con este y generar posibles ofertas.

Para que esto funcione se necesitan dos elementos base. El primero es lo que llamamos el apoyo, consisten en cuantas ocasiones la compra de A ha sido seguida de la compra de B. El segundo elemento es la confianza, que es un porcentaje de la probabilidad de que se lleguen a comprar ambos elementos juntos.

4.3.2 Clustering

Esta técnica se utiliza con grandes cantidades de información, dicha información maneja una gran cantidad de variables. Lo que se está buscando es crear una serie de agrupaciones en las cuales

deben caer los datos basándose en sus características o los valores de sus variables. Estas agrupaciones o categorías no son fáciles de encontrar cuando se está lidiando con una gran cantidad de variables diferentes.

Por ejemplo tenemos una aplicación que maneja miles de candidatos para miles de puestos de trabajo, una especie de buscador de empleo. En esta aplicación se maneja una gran cantidad de variables como edad, sexo, experiencia, preferencias, historial de empleo, nivel de educación, etc. Por medio de técnicas de clustering podemos dividir a miles de candidatos en una serie de grupos bien definidos y de esta manera poder manejarlos de una manera más fácil.

Un buen algoritmo de clustering nos hace ver que los grupos creados están bien definidos y que cada elemento pertenece a un solo grupo sin mucha posibilidad o duda de que pueda pertenecer a otro.

4.3.3 Extracción de atributos

Es un proceso por medio del cual se extraen los atributos más importantes de un conjunto de datos. Existen cierta similitud entre este proceso y el proceso de clustering, pero la diferencia radica en que aquí no definimos cuales son los atributos mientras que en el clustering si se hace.

Por ejemplo tenemos un conjunto de cartas o correos electrónicos dirigidos a una organización. Esta organización desea

47

Seminario de Sistemas 2

48

Seminario de Sistemas 2

Figura 78 – Páginas 49-52 Libro “Data Warehouse”

crear una especie de filtro para saber si el correo es una petición de compra, una queja, publicidad, etc. Aquí no sabemos qué variables o características de cada correo nos ayudarían a diferenciarlos y por medio de extracción de atributos podemos encontrar estas variables.

- *Mining a Large Database – A case study*
Herb Edlestein
Ref: <http://www.twocrows.com/largedb.pdf>

Artículos relacionados

- *Data Mining on the web*
Dan R. Greening
Ref: <http://www.webtechniques.com/archives/2000/01/greening/>
- *Government increasingly turning to data mining*
Arshad Mohammed y Sara Kehaulani
Ref: <http://www.washingtonpost.com/wp-dyn/content/article/2006/06/14/AR2006061402063.html>
- *Description is not prediction*
Herb Edlestein
Ref: <http://www.dnreview.com/issues/20030301/6388-1.html>

49

Seminario de Sistemas 2

50

Seminario de Sistemas 2

Glosario

OLAP: Online Analytical Processing o Procesamiento analítico en línea.
OLTP: Online Transactional Processing o Procesamiento transaccional en línea.
ROLAP: Relational OLAP, tiene el mismo significado que OLAP pero aplicado a bases de datos relacionales.
MOLAP: Multidimensional OLAP, tiene el mismo significado que OLAP pero aplicado a bases de datos multidimensionales.
Warehouse: Almacén en inglés, utilizado en el término Data Warehouse.
Fact Table: La traducción literal a español es tabla de hechos, es un componente utilizado en el diseño de un Data Warehouse.
Snowflake: En inglés copo de nieve. El esquema snowflake es una de las dos posibles opciones a utilizar en un Data Warehouse.
ETL: Proceso por medio del cual se carga información a un Data Warehouse a partir de una o más fuentes de datos.
Surrogate: Sustituto en inglés, se utiliza en el término Surrogate keys utilizado en el proceso de ETL.
DSS: Decision Support System o sistema de soporte para toma de decisiones en inglés. Medio por el cual se interactúa con la información de un Data Warehouse.

51

Seminario de Sistemas 2

BI: Siglas para Business Intelligence, este término se puede traducir literalmente a Inteligencia de Negocio.

Data Mining: En español se puede traducir a minería de datos. Proceso por medio del cual se encuentran tendencias escondidas en los datos.

52

Seminario de Sistemas 2

Figura 79 – Páginas 53-56 Libro “Data Warehouse”

Referencias Bibliográficas

- Lillian Hobbs, Susan Hillson, Shilpa Lawande, Pete Smith. *Oracle Database 10g Data Warehousing*.
- Maria Sueli Almeida, Missao Ishikawa, Joerg Reinschmidt, Torsten Roerber. *Getting started with Data Warehouse and Business Intelligence*.
- Vincent Rainardi, *Building a Data Warehouse with examples in SQL Server*.
- Fon Silvers, *Building and maintaining a Data Warehouse*.
- W. H. Immon, *Building the Data Warehouse*.

APÉNDICE A: LICENCIA

Este trabajo se encuentra definido bajo la licencia Creative Commons Attribution-ShareAlike License. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-sa/3.0/>.

A continuación se presenta un resumen de la licencia.

1.1.1 You are free:

- to Share — to copy, distribute and transmit the work
- to Remix — to adapt the work

1.1.2 Under the following conditions:

- **Attribution** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

What does "Attribute this work" mean?

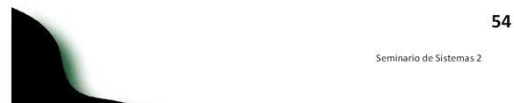
The page you came from contained embedded licensing metadata, including how the creator wishes to be attributed for re-use. You can use the HTML here to cite the work. Doing so will also include metadata on your page so that others can find the original work as well.

- **Share Alike** If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.
- For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.



53

Seminario de Sistemas 2



54

Seminario de Sistemas 2

- Any of the above conditions can be waived if you get permission from the copyright holder.
- Nothing in this license impairs or restricts the author's moral rights.

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED. BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

- "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-representation with a moving image ("synchronizing") will be considered an Adaptation for the purpose of this License.
- "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in

themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.

- "Creative Commons Compatible License" means a license that is listed at <http://creativecommons.org/compatiblelicenses> that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License, and; (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.
- "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
- "License Elements" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
- "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
- "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and; (iii) in the case of broadcasts, the organization that transmits the broadcast.
- "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.
- "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who



55

Seminario de Sistemas 2



56

Seminario de Sistemas 2

Figura 80 – Páginas 57-60 Libro “Data Warehouse”



has received express permission from the Licensor to exercise rights under this License despite a previous violation.

j. **“Publicly Perform”** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

k. **“Reproduce”** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. **Fair Dealing Rights.** Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. **License Grant.** Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked “The original work was translated from English to Spanish,” or a modification could indicate “The original work has been modified.”;

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

i. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License.

ii. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and

iii. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. **Restrictions.** The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.

b. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the “Applicable License”), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You

57

Seminario de Sistemas 2

58

Seminario de Sistemas 2



must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (v) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.

c. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution (“Attribution Parties”) in Licensor’s copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI for, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., “French translation of the Work by Original Author”) or “Screenplay based on original Work by Original Author”). The credit required by this Section 4(c) may be implemented in any reasonable manner, provided, however, that in the case of an Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

d. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author’s honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action

prejudicial to the Original Author’s honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. **Representations, Warranties and Disclaimer.** UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. **Limitation on Liability.** EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. **Termination**

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however, that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. **Miscellaneous**

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this

59

Seminario de Sistemas 2

60

Seminario de Sistemas 2



Figura 81 – Páginas 61-62 Libro “Data Warehouse”



License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1992, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice
Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.
Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark “Creative Commons” or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons’ then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of the License.
Creative Commons may be contacted at <http://creativecommons.org/>.

CONCLUSIONES

1. Se realizó la estructuración de los laboratorios de los cursos Sistemas de Bases de Datos 1, Sistemas de Bases de Datos 2 y Arquitectura de Computadores y Ensambladores 2. El trabajo se basó en los contenidos actuales, sobre esto se hizo una revisión y actualización del contenido guiado por el profesor titular de cada curso.
2. Para cada *instructor guideline* se detallaron puntos importantes como: detalle de horario de sesiones, laboratorios, prácticas, exámenes, ejercicios sugeridos y la ponderación en puntos.
3. Detallando cada sesión, se realizaron presentaciones con el contenido para cada una. Para las prácticas, exámenes y ejercicios sugeridos se proveen ejemplos así como los objetivos que se buscan de cada actividad.
4. Se realizó la guía de clase para los cursos Sistemas de Bases de Datos 2, Seminario de Sistemas 2 e Inteligencia Artificial. El contenido fue realizado con base en el contenido viejo, al igual que con los *instructor guidelines* se realizaron revisiones y actualizaciones guiadas por el profesor titular de cada curso.
5. Al final de cada unidad de la guía se incluyen artículos de interés como apoyo al contenido base.

6. Con la creación de estos documentos se está creando una guía detallada para impartir los laboratorios y clases de los cursos de Ciencias y Sistemas. Se provee la plataforma para estandarizar los contenidos y para facilitar la actualización de los mismos.

7. Esta guía resulta esencial para el instructor del curso ayudándole en la planeación de impartir sesiones, realizar pruebas, ejercicios y prácticas.

RECOMENDACIONES

1. Es crucial para el éxito de este proyecto tomar en cuenta que el contenido debe ser actualizado constantemente, es importante tomar en cuenta que la estructura básica debe mantenerse y todos los módulos en la guía deben conservarse, ya que cada uno tiene un objetivo específico.
2. Es importante para conservar la integridad de estas guías, que las personas encargadas de realizar las actualizaciones tengan experiencia en cada curso y que sean guiadas por el profesor titular del mismo.
3. Finalmente, se recomienda llevar control de las diferentes versiones de las guías al realizar actualización de contenido, para evitar pérdida de información.

BIBLIOGRAFÍA

1. Mrutunjaya Panda. Curso de Planificación Educativa a través de *Instructor Guidelines*. India-Guatemala *IT Education Centre of Excellence*. Guatemala, 2008.
2. Araham Silberschatz & James. *Operating Systems Concepts*.
3. Date, C.J. Introducción a los sistemas de bases de datos.
4. Elmasri y Navathe. *Sistemas de Bases de datos*.
5. Ron Soukup. *Microsoft SQL Server a Fondo*.
6. Andy Opper. *Databases Demystified*.
7. Kevin Loney. *Oracle Database 10g: the complete reference*.
8. Almeida, Ishikawa, Reinschmidt y Roeber. *Getting started with Data Warehouse and Business Intelligence*.
9. Fon Silvers. *Building and maintaining a Data Warehouse*.
10. Ben Schneiderman. *Leonardo's Laptop: Human Needs and the New Computing Technologies*.

11. Marshall McLuhan. *The Medium is the message.*

12. Peter L. Berger. *Construcción Social de la Realidad.*

ANEXOS

La participación del profesor titular de cada curso fue crucial para el éxito de este proyecto, por esta razón se considera importante validar su participación.

Por lo cual, en el anexo se adjuntan las cartas autorizando el contenido por parte del profesor titular de cada curso. Para cada curso el profesor titular revisó la guía de su curso y verificó que los objetivos detrás del proyecto se cumplieran.

Se da a continuación la lista de los profesores que participaron en el proyecto:

- Ing. Javier Gramajo
- Ing. Armín Mazariegos
- Ing. Luis Espino




Guatemala 20 de julio de 2009

Ingeniero
Jorge Armin Mazariegos
Asesor de EPS
Facultad de Ingeniería
USAC

Por este medio me dirijo a usted para informarle que he revisado el **instructor guideline** de “*Sistemas de Bases de datos 1*” del alumno Franklym Algaba, del Proyecto “*Aplicación de la metodología de la empresa TATA CONSULTANCY SERVICES a las áreas Bases de datos, Inteligencia Artificial e Innovación de la carrera de Ciencias y Sistemas*”, y a mi parecer cumple con los objetivos de guía de laboratorio.

Agradeciendo su atención a la presente,

Atentamente,



Ing. Luis Espino
Catedrático de la Escuela de Ciencias y Sistemas
Facultad de Ingeniería, USAC





Guatemala 20 de julio de 2009

Ingeniero
Jorge Armin Mazariegos
Asesor de EPS
Facultad de Ingeniería
USAC

Por este medio me dirijo a usted para informarle que he revisado el **instructor guideline de "Sistemas de Bases de datos 2"** del alumno Franklym Algaba, del Proyecto *"Aplicación de la metodología de la empresa TATA CONSULTANCY SERVICES a las áreas Bases de datos, Inteligencia Artificial e Innovación de la carrera de Ciencias y Sistemas"*, y a mi parecer cumple con los objetivos de guía de laboratorio.

Agradeciendo su atención a la presente,

Atentamente,

Ing. Luis Espino
Catedrático de la Escuela de Ciencias y Sistemas
Facultad de Ingeniería, USAC



Guatemala 20 de julio de 2009

Ingeniero
Jorge Armin Mazariegos
Asesor de EPS
Facultad de Ingeniería
USAC

Por este medio me dirijo a usted para informarle que he revisado la **guía de clase de "Clase de Seminario de Sistemas 2"** del alumno Franklym Algaba, del Proyecto *"Aplicación de la metodología de la empresa TATA CONSULTANCY SERVICES a las áreas Bases de datos, Inteligencia Artificial e Innovación de la carrera de Ciencias y Sistemas"*, y a mi parecer cumple con los objetivos de guía de clase.

Agradeciendo su atención a la presente,

Atentamente,

Ing. Luis Espino
Catedrático de la Escuela de Ciencias y Sistemas
Facultad de Ingeniería, USAC




Guatemala 20 de julio de 2009

Ingeniero
Jorge Armin Mazariegos
Asesor de EPS
Facultad de Ingeniería
USAC

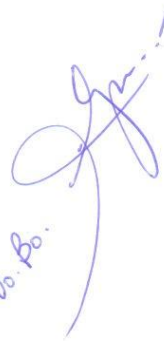
Por este medio me dirijo a usted para informarle que he revisado la **guía de clase de "Clase de Sistemas de Bases de datos 2"** del alumno Franklym Algaba, del Proyecto "*Aplicación de la metodología de la empresa TATA CONSULTANCY SERVICES a las áreas Bases de datos, Inteligencia Artificial e Innovación de la carrera de Ciencias y Sistemas*", y a mi parecer cumple con los objetivos de guía de laboratorio.

Agradeciendo su atención a la presente,

Atentamente,



Ing. Luis Espino
Catedrático de la Escuela de Ciencias y Sistemas
Facultad de Ingeniería, USAC


Vo. Bo.



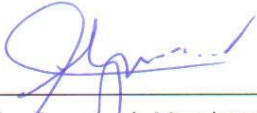
Guatemala 18 de julio de 2009

Ingeniero
Jorge Armin Mazariegos
Asesor de EPS
Facultad de Ingeniería
USAC

Por este medio me dirijo a usted para informarle que he revisado el **Instructor Guideline** de "*Clase de Arquitectura de Computadoras y Ensambladores 2*" del alumno Franklym Algaba, del Proyecto "*Aplicación de la metodología de la empresa TATA CONSULTANCY SERVICES a las áreas Bases de datos, Inteligencia Artificial e Innovación de la carrera de Ciencias y Sistemas*", y a mi parecer cumple con los objetivos de guía de laboratorio.

Agradeciendo su atención a la presente,

Atentamente,



Ing. Jorge Armin Mazariegos
Catedrático de la Escuela de Ciencias y Sistemas
Facultad de Ingeniería, USAC



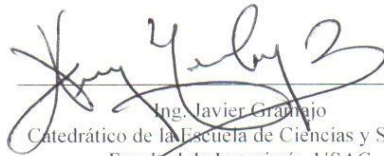
Guatemala 3 de marzo de 2009

Ingeniero
Jorge Armin Mazariegos
Asesor de EPS
Facultad de Ingeniería
USAC

Por este medio me dirijo a usted para informarle que he revisado la **guía de clase de "Clase de Inteligencia Artificial 1"** del alumno Franklym Algaba, del Proyecto "*Aplicación de la metodología de la empresa TATA CONSULTANCY SERVICES a las áreas Bases de datos, Inteligencia Artificial e Innovación de la carrera de Ciencias y Sistemas*", y a mi parecer cumple con los objetivos de guía de laboratorio.

Agradeciendo su atención a la presente,

Atentamente,


Ing. Javier Graña
Catedrático de la Escuela de Ciencias y Sistemas
Facultad de Ingeniería, USAC


Vo. Bo.