



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

**ANÁLISIS Y RECOMENDACIONES PARA EL CURSO DE  
ESTRUCTURA DE DATOS  
DEL PENSUM DE ESTUDIO DE LA CARRERA DE  
INGENIERÍA EN CIENCIAS Y SISTEMAS  
DE LA FACULTAD DE INGENIERÍA, USAC**

**Guillermo Octavio López González**

Asesorado por el Ing. Jorge Armín Mazariegos Rabanales

Guatemala, octubre de 2010



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ANÁLISIS Y RECOMENDACIONES PARA EL CURSO DE ESTRUCTURA DE  
DATOS DEL PENSUM DE ESTUDIO DE LA CARRERA INGENIERÍA EN  
CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERIA, USAC**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**GUILLERMO OCTAVIO LÓPEZ GONZÁLEZ**

ASESORADO POR EL ING. JORGE ARMÍN MAZARIEGOS

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, OCTUBRE DE 2010



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. Luis Pedro Ortiz de León
VOCAL V	Agr. José Alfredo Ortiz Herincx
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Julio González Podszueck
EXAMINADOR	Ing. Jorge Luis Álvarez Mejía
EXAMINADOR	Ing. Carlos Alfredo Azurdia Morales
EXAMINADOR	Ing. Federico Guillermo Velarde Morales
SECRETARIO	Ing. Francisco Javier González López



## **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 graduación titulado:

### **ANÁLISIS Y RECOMENDACIONES PARA EL CURSO DE ESTRUCTURA DE DATOS DEL PENSUM DE ESTUDIO DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA, USAC**

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, en junio de 2006.

Guillermo Octavio López González



Guatemala, 5 de abril de 2010

Escuela de Ciencias y Sistemas  
Facultad de Ingeniería

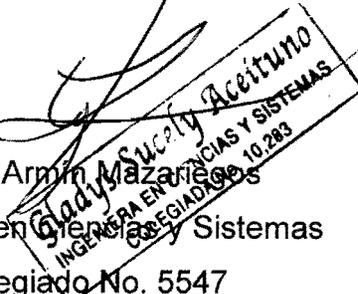
Respetable:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **GUILLERMO OCTAVIO LOPEZ GONZALEZ** Carnet **84-11329**, titulado: **"ANALISIS Y RECOMENDACIONES PARA EL CURSO DE ESTRUCTURA DE DATOS DEL PENSUM DE ESTUDIO DE LA CARRERA DE INGENIERIA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERIA, USAC"**, y a mi criterio el mismo cumple con todos los objetivos propuestos para su desarrollo, según el protocolo.

Sin otro particular, me suscribo de usted.

Atentamente,

X  
Jorge Armin Mazariegos  
Ingeniero en Ciencias y Sistemas  
Colegiado No. 5547



Asesor y Revisor de Trabajo de Graduación



Guatemala, 31 de mayo de 2010.  
REF:EPS:DOC:682.05.10.

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, **Guillermo Octavio López González** Carné No. **8411329** procedí a revisar el informe final, cuyo título es **"ANÁLISIS Y RECOMENDACIONES PARA EL CURSO DE ESTRUCTURA DE DATOS DEL PENSUM DE ESTUDIO DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA, USAC"**.

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

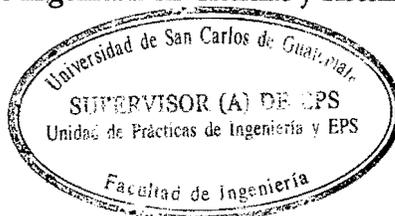
Sin otro particular, me es grato suscribirme.

Atentamente,

"Id y Enseñad a Todos"

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

FFAPdM/RA





Guatemala, 31 de mayo de 2010.  
REF.EPS.D.424.05.10.

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 **“ANÁLISIS Y RECOMENDACIONES PARA EL CURSO DE ESTRUCTURA DE DATOS DEL PENSUM DE ESTUDIO DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA, USAC”**, que fue desarrollado por el estudiante universitario **Guillermo Octavio López González** Carné No. **8411329** quien fue debidamente asesorado por el Ing. Jorge Armin Mazariegos y supervisado por la Ing. Florida 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 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 Todos”

  
Inga. Norma Ileana Sarmiento Zecena de Serrano  
Directora Unidad de EPS

NISZ/ra



UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA

Guatemala, 22 de junio de 2010

Ingeniera  
Norma Ileana Sarmiento Zeceña de Serrano  
Directora de la Unidad de EPS  
Facultad de Ingeniería

Ingeniera Sarmiento:

Por este medio le informo que la INGA. GLADYS ACEITUNO estará co-asesorando el trabajo de Ejercicio Profesional Supervisado (EPS) titulado "ANALISIS Y RECOMENDACIONES PARA EL CURSO DE ESTRUCTURA DE DATOS DEL PENSUM DE ESTUDIO DE LA CARRERA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERIA, USAC", el cual ha sido propuesto por el estudiante GUILLERMO OCTAVIO LOPEZ GONZALEZ carne 8411329.

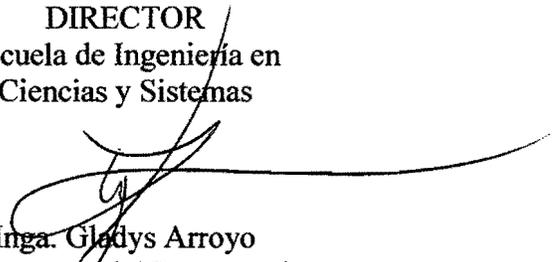
Esta disposición ha sido aprobada con el fin de apoyar al ING. ARMÍN MAZARIEGOS INGENIERO ASESOR del proyecto, ya que actualmente ha acumulado una carga de trabajo muy fuerte dentro del entorno de los EPSs de la Escuela de Ciencias y Sistemas. Cabe mencionar que esta disposición ya fue expuesta a Junta Directiva y nos expresaron su apoyo a nuestra decisión.

Agradeciendo la atención a la presente y quedando a sus órdenes para cualquier información adicional, me suscribo,

Atentamente,

"Id Y Enseñad a todos"

Ing. Marlon Antonio Pérez Türk  
DIRECTOR  
Escuela de Ingeniería en  
Ciencias y Sistemas

  
Inga. Gladys Arroyo  
Co-Asesora del Proyecto de  
E.P.S.

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

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 "ANÁLISIS Y RECOMENDACIONES PARA EL CURSO DE ESTRUCTURA DE DATOS DEL PENSUM DE ESTUDIO DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA, USAC", presentado por el estudiante GUILLERMO OCTAVIO LÓPEZ GONZÁLEZ, aprueba el presente trabajo y solicita la autorización del mismo.*

**"ID Y ENSEÑAD A TODOS"**

*Ing. Marlon Antonio Pérez Turk*

Director, Escuela de Ingeniería Ciencias y Sistemas



Guatemala, 14 de octubre  
2010



DTG. 326.2010

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: **ANÁLISIS Y RECOMENDACIONES PARA EL CURSO DE ESTRUCTURA DE DATOS DEL PENSUM DE ESTUDIO DE LA CARRERA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA, USAC,** presentado por el estudiante universitario **Guillermo Octavio López González,** autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Murphy Olimpo Paiz Reyes  
Decano



Guatemala, 18 de octubre de 2010.

/gdech

**MIS AGRADECIMIENTOS A:**

**Dios**

**Mis Padres**

**La Facultad de Ingeniería**

**La Universidad de San Carlos de Guatemala**



**DEDICO ESTE ACTO A:**

**mis padres**

**mis hermanos**

**mi esposa**

**Y especialmente, a mis hijos**



# ÍNDICE GENERAL

<b>ÍNDICE DE ILUSTRACIONES</b> .....	V
<b>GLOSARIO</b> .....	VII
<b>RESUMEN</b> .....	XXVII
<b>OBJETIVOS</b> .....	XXIX
<b>INTRODUCCIÓN</b> .....	XXXI
<b>1. MARCO TEÓRICO</b> .....	1
1.1. Importancia de la educación .....	1
1.2. Modelo educativo .....	2
1.3. Algunos tipos de modelos educativos .....	3
1.3.1. El modelo tradicional .....	3
1.3.2. El modelo de Ralph Tyler.....	4
1.3.3. Modelo de Popham-Baker .....	6
1.3.4. Modelo de Roberto Mager.....	8
1.3.5. Modelo de Hilda Taba .....	9
1.4. Dinámica del curso .....	11
1.5. Descripción del curso .....	12
1.5.1. ¿Por qué Java? .....	13
1.5.2. ¿Por qué seleccionar a Java como plataforma de desarrollo?.....	13
1.5.3. Ventajas de Java .....	15
1.5.4. Conclusión .....	17
1.6. Atribuciones del catedrático .....	18
1.7. Casos especiales .....	19
1.8. Publicación de notas .....	20
1.9. ITFORCEGT .....	21

<b>2. EVOLUCIÓN DEL CURSO SEGUNDO SEMESTRE 2006</b> .....	23
2.1. Descripción del curso.....	23
2.2. Distribución de la nota.....	23
2.3. Principales Hallazgos.....	25
2.3.1. Problema del lenguaje.....	25
2.3.2. Tiempo insuficiente para impartir la clase magistral, por varios factores ....	25
<b>3. EVOLUCIÓN DEL CURSO PRIMER SEMESTRE 2007</b> .....	27
3.1. Descripción del curso.....	27
3.2. Distribución de la nota.....	27
3.3. Principales hallazgos.....	29
3.3.1. Conocimiento de idioma Inglés .....	29
3.3.2. Tiempo insuficiente para impartir la clase magistral, por varios factores ....	32
<b>CONCLUSIONES</b> .....	33
<b>RECOMENDACIONES</b> .....	35
<b>BIBLIOGRAFÍA</b> .....	39
<b>ANEXOS</b> .....	41
<b>ANEXO 1: PROGRAMA DEL CURSO</b> .....	41
<b>ANEXO 2: DOCUMENTO SOBRE ITFORCEGT</b> .....	49
<b>ANEXO 3: MATERIAL SEGUNDO SEMESTRE 2006</b> .....	55
PRIMERA EVALUACIÓN PARCIAL SEGUNDO SEMESTRE 2006.....	55
SEGUNDA EVALUACIÓN PARCIAL SEGUNDO SEMESTRE 2006 .....	63
EVALUACIÓN FINAL SEGUNDO SEMESTRE 2006.....	71
ENUNCIADO PRIMER PROYECTO SEGUNDO SEMESTRE 2006.....	77

ENUNCIADO SEGUNDO PROYECTO	SEGUNDO SEMESTRE 2006.....	83
NOTAS FINALES SEGUNDO SEMESTRE 2006 .....		87
<b>ANEXO 4: MATERIAL PRIMER SEMESTRE 2007 .....</b>		<b>91</b>
PRIMERA EVALUACIÓN PARCIAL PRIMER SEMESTRE 2007 .....		91
SEGUNDA EVALUACIÓN PARCIAL PRIMER SEMESTRE 2007 .....		97
TERCERA EVALUACIÓN PARCIAL PRIMER SEMESTRE 2007 .....		105
EVALUACIÓN FINAL PRIMER SEMESTRE 2007 .....		109
ENUNCIADO PRIMER PROYECTO PRIMER SEMESTRE 2007 .....		117
EXAMEN DE PRIMERA RETRASADA PRIMER SEMESTRE 2007 .....		133
EXAMEN DE SEGUNDA RETRASADA PRIMER SEMESTRE 2007 .....		135
NOTAS PRIMERA RETRASADA PRIMER SEMESTRE 2007 .....		137
NOTAS SEGUNDA RETRASADA PRIMER SEMESTRE 2007 .....		137



## ÍNDICE DE ILUSTRACIONES

1. Gráfica de resultado de estudiantes aprobados 2° semestre 2006..... 23
2. Gráfica de resultado de estudiantes aprobados 1° semestre 2007..... 27



## GLOSARIO

<b><i>Abstract Windowing Toolkit (AWT)</i></b>	Biblioteca de módulos para representar interfaces gráficos provisto por Sun en la API de Java.
<b>Administrador de seguridad</b>	Parte de la máquina virtual Java responsable de velar por el cumplimiento de las políticas y reglas de seguridad.
<b>Ámbito</b>	Parte de un programa en el que es válida una referencia a una variable.
<b><i>American Standard Code for Information Interchange (ASCII)</i></b>	Sistema de codificación que convierte caracteres a números en el rango de 0 a 127. Es una parte del código ANSI que se amplía hasta los 257 caracteres.
<b>Análisis</b>	Proceso de conocer los requerimientos de <i>software</i> que tienen el cliente y el usuario final.
<b>API</b>	<i>Application Programming Interface.</i>
<b>Aplicación</b>	Programa informático, que se ejecuta sin necesidad de otro programa.

<b>Applet</b>	Programa informático que se ejecuta necesitando de otro programa, normalmente un navegador.
<b><i>Application Programming Interface (API)</i></b>	Conjunto de paquetes y clases Java, incluidos en el JDK que utilizan los programadores Java para realizar sus aplicaciones.
<b>Árbol</b>	Estructura de datos, grafo no cíclico, con forma de árbol (nodos padres e hijos).
<b>Argumentos</b>	Parámetros.
<b>Array</b>	Vector.
<b>ASCII</b>	<i>American Standard Code for Information Interchange.</i>
<b>AWT</b>	<i>Abstract Windowing Toolkit.</i>
<b>BDK</b>	<i>Beans Developer Kit.</i>
<b><i>Beans Developer Kit (BDK)</i></b>	Conjunto de herramientas para desarrollar <i>JavaBeans</i> .
<b>Bit</b>	Unidad mínima de información digital que puede tomar los valores lógicos de 0 o 1.

<b>Bloque</b>	Código localizado entre corchetes.
<b><i>Boolean</i></b>	Tipo de datos bi-estado, que puede tomar valor de cierto ( <i>true</i> ) o falso ( <i>false</i> ).
<b><i>Byte</i></b>	Secuencia de 8 bits.
<b>Cadena</b>	Secuencia de caracteres.
<b>Carácter</b>	Símbolo que representa información, o la codificación en una computadora. Normalmente letras de alfabeto, números o signos ASCII.
<b>Cargador de clases</b>	Parte del JRE de Java responsable de encontrar archivos de clase y cargarlos en la máquina virtual Java.
<b><i>Casting</i></b>	Moldeado
<b>CGI</b>	Common Gateway Interfaz.

<b>Clase</b>	Unidad fundamental de programación en Java, que sirve como plantilla para la creación de objetos. Una clase define datos y métodos y es la unidad de organización básica de un programa Java.
<b><i>Common Gateway Interfaz (CGI)</i></b>	Es un lenguaje de programación que permite dotar a las páginas Web de interactividad, construyendo una página Web correspondiente a un enlace de hipertexto en el mismo momento en que se hace "clic" sobre el enlace. Los <i>script cgi</i> pueden estar escritos en cualquier lenguaje de programación.
<b><i>Common Object Request Broker Architecture (CORBA)</i></b>	Estándar para la conexión entre objetos distribuidos, aunque esté codificados en distintos lenguajes.
<b>Compilador</b>	Programa de software que traduce código fuente en un lenguaje de programación legible por una persona a código máquina interpretable por un ordenador.

<b>Constante</b>	Valor utilizado en un programa de computadoras con la garantía de no cambiar en tiempo de ejecución. La garantía es a menudo reforzada por el compilador. En Java las constantes se declaran como <i>static final</i> .
<b>Constructor</b>	Método que tiene el mismo nombre que la clase que inicia. Toma cero o más parámetros y proporciona unos datos u operaciones iniciales dentro de una clase, que no se pueden expresar como una simple asignación.
<b>Contenedor</b>	En diseño de interfaces de usuario, es un objeto que contiene los componentes (como botones, barras de deslizamiento y campos de texto).
<b>Conversión de tipos de datos</b>	Modificación de una expresión de un tipo de datos a otro.
<b>CORBA</b>	<i>Common Object Request Broker Architecture.</i>
<b>Entero</b>	Un número entero, sin parte decimal, positivo o negativo.

<b>Estructura de datos</b>	Una construcción de software (en memoria o en disco duro) que contiene datos y las relaciones lógicas entre ellos.
<b>Evento</b>	Un mensaje que significa un incidente importante, normalmente desde fuera del entorno de <i>software</i> .
<b>Excepción</b>	Un evento que ocurre durante la ejecución de un programa que interrumpe el flujo normal de las instrucciones.
<b>Flujo</b>	<i>Stream</i> .
<b><i>Graphical User Inteface (GUI)</i></b>	Interfaz gráfica de usuario.
<b><i>Hardware</i></b>	El aspecto físico de un sistema de computadora, como el procesador, disco duro e impresora.
<b>Herencia múltiple</b>	La práctica (permitida en lenguajes como C++ pero no en Java) de derivar una clase de más de una <i>clase base</i> .

<b>Herencia</b>	Mecanismo encargado de relacionar clases entre sí de una manera jerárquica. En Java, sólo existe herencia simple.
<b>Hilo</b>	<i>Thread.</i>
<b>HTML (<i>HyperText Markup Languaje</i>)</b>	Lenguaje que se utiliza para crear páginas Web. Los programas de navegación de la Web muestran estas páginas de acuerdo con un esquema de representación definido por el programa de navegación.
<b>IDE</b>	<i>Integral Development Environment.</i>
<b>IDL</b>	<i>Java Interface Definition Language.</i>
<b>Ingeniería del <i>software</i></b>	Rama de la ingeniería concerniente con el análisis, diseño, implementación, prueba, y mantenimiento de programas de computadoras.
<b>Instancia</b>	Objeto de software construido desde una clase. Por ejemplo, puede tener una clase avión, pero una flota de quince instancias de avión.

<b><i>Integral Development Enviroment (IDE)</i></b>	Una herramienta de desarrollo visual en la que un programa puede ser construido, ejecutado y depurado.
<b>Interbloqueo</b>	Condición en la que dos o más entidades de software se bloquean mutuamente, cada una esperando los recursos que está utilizando la otra.
<b><i>Interface Definition Language (IDL)</i></b>	Herramienta mediante la cual los objetos pueden invocar métodos de otros objetos que se encuentren en máquinas remotas, mediante CORBA.
<b>Interfaz gráfica de usuario (GUI)</b>	Una interfaz entre la máquina y el hombre como el Windows de Microsoft, el Mac OS, o el Sistema X Windows, que depende de pantallas de alta resolución, un recurso gráfico de puntero como un ratón y una colección de controles en pantalla (denominados Widgets) que el usuario puede manejar directamente.

**Interfaz**

Mecanismo Java para decirle al compilador que un conjunto de métodos serán definidos en futuras clases. (Esas clases estarán definidas para implementar la interfaz).

**Java 2D**

Paquete que permite a los desarrolladores incorporar texto, imágenes y gráficos en dos dimensiones de gran calidad.

**Java 3D**

Conjunto de clases para crear aplicaciones y applets con elementos en tres dimensiones. Es parte del JMF.

***Java DataBase Connectivity (JDBC)***

Lenguaje estándar de Java para interactuar con bases de datos, similar al SQL. Es independiente no sólo de la plataforma sino también de la base de datos con que interactúe. Desde la versión 1.2 del JDK se permite interactuar con ODBC.

***Java Developer Connection (JDC)***

Conexión de desarrollo en la que se publican las versiones beta de las bibliotecas de Java que se están desarrollando.

***Java Foundation Classes (JFC)***

Conjunto de componentes y características para construir programas con interfaces gráficas.

***Java Media Framework (JMF)***

Protocolo de transmisión de datos para la reproducción multimedia (vídeo y sonido).

***Java Native Invocation (JNI)***

Capacidad de Java para ejecutar código nativo, es decir, código compilado al lenguaje máquina de un determinado ordenador. Permite a la Máquina Virtual Java (JVM) interactuar con programas o bibliotecas escritos en otros lenguajes (C/C++, ensamblador...). No se puede utilizar en applets, pues viola las directrices de seguridad.

***Java Runtime Environment (JRE)***

*Software* suministrado por *Sun* que permite a los programas de Java ejecutarse en una máquina de usuario. El JRE incluye la Máquina Virtual Java (JVM).

**JRE**

*Java Runtime Environment.*

**JVM**

*Java Virtual Machine.*

***Java Virtual Machine (JVM)***

El intérprete de Java que ejecuta los códigos de byte en una plataforma particular.

***JavaBeans***

Paquete que permite escribir componentes *software Java*, que se puedan incorporar gráficamente a otros componentes.

***JDBC***

*Java DataBase Connectivity.*

***JDC***

*Java Developer Connection.*

***JFC***

*Java Foundation Classes.*

***JMF***

*Java Media Framewok*

***JNI***

*Java Native Invocation.*

***JVM***

*Java Virtual Machine.*

**Llamada por referencia**

Una forma de transferir parámetros a una subrutina en la que se pasa un puntero o referencia a un elemento, de esta forma, la subrutina puede leer y cambiar el valor del elemento referenciado.

**Llamada por valor**

Una forma de transferir parámetros a una subrutina en la que se pasa la copia del elemento; las modificaciones de la copia no afectan al elemento original.

**Método**

Conjunto de sentencias que operan sobre los datos de la clase para manipular su estado.

**Miniaplicación**

*Applet.*

**Modelo**

En diseño orientado a objetos, una representación del mundo real en unas abstracciones de *software* denominadas clases y la relación entre ellas.

**Moldeado**

Suplantación del tipo de un objeto o variable por otro nuevo tipo.

<b>Multiproceso</b>	En sistemas operativos, la habilidad de efectuar dos o más programas independientes, comúnmente en un procesador solo (a través de <i>Multitarea</i> ).
<b>Navegador Web</b>	<i>Software</i> que permite al usuario conectarse a un servidor de Web utilizando <i>Hypertext Transfer Protocol</i> (HTTP). Microsoft Internet Explorer, Netscape Navigator, HotJava de Sun, son populares navegadores de Web.
<b>Navegador</b>	Navegador Web.
<b><i>Null</i></b>	Valor de Java que significa <i>vacío</i> .
<b><i>Object DataBase Connectivity (ODBC)</i></b>	Lenguaje estándar de Microsoft; que utiliza un driver del fabricante de una base de datos, para interactuar con ella, más orientado a C/C++ que a Java.
<b>ODBC</b>	<i>Object DataBase Connectivity</i> .
<b>Paquete</b>	Nombre de Java para una biblioteca de clases.

**Parámetros formales**

Nombres utilizados dentro de una subrutina por sus parámetros.

**Parámetros**

Valores u objetos pasados entre una subrutina y la rutina de llamada.

***Plug-in***

Un programa de una plataforma específica diseñado para ser llamado por un navegador Web. Utilizado con frecuencia para mostrar información que el mismo navegador no puede mostrar.

**Poliforfismo**

En diseño orientado a objetos, la habilidad de utilizar una clase derivada en lugar de su clase base. Por ejemplo, un programador puede escribir un método *expresarse()* para la clase mamífero. Un perro, una vaca y un gato se derivan de mamífero y todos pueden *expresarse()*, aunque sus voces sean bastantes diferentes.

**Proceso**

Instancia de un programa ejecutable. Por ejemplo, si inicia dos copias de un intérprete de Java, tiene dos procesos de la máquina virtual de Java ejecutándose en su computadora.

**Recolector de basura**

En Java, el mecanismo por el cual se recobra y libera la memoria asociada con objetos no utilizados.

***Remote Method Invocation (RMI)***

Herramienta que incorpora métodos Java para localizar objetos remotos, comunicarse con ellos e incluso enviar objetos como parámetros de un objeto a otro.

**RMI**

*Remote Method Invocation.*

***Secure Sockets Layer (SSL)***

Sistema para la creación de conexiones seguras en red.

***Servlets***

Módulos que permiten sustituir o utilizar el lenguaje Java en lugar de programas CGI.

**Seudocódigo**

Documentación de diseño que describe el trabajo de un programa en inglés estructurado (o en otro lenguaje) en lugar de un lenguaje de computadora.

<b>Shell</b>	Intérprete de órdenes de un sistema operativo.
<b>Sistema operativo</b>	Software responsable de asignar a los usuarios los recursos de sistemas de computadoras (incluyendo procesos). UNIX, Windows, NT y Mac OS, son ejemplos de sistemas operativos.
<b>SQL</b>	<i>Structured Query Language.</i>
<b>SSL</b>	<i>Secure Sockets Layer.</i>
<b>Static</b>	En diseño orientado a objetos, representa la pertenencia a la clase, en vez de a una instancia. Es un espacio compartido por todas las instancias de una clase.
<b><i>Stream</i></b>	Flujo de datos. Por ejemplo las entradas y salidas de un programa.
<b><i>String</i></b>	Objeto Java estandarizado en el lenguaje, que representa una cadena de caracteres.

***Structured Query Language (SQL)***

Lenguaje para realizar consultas a Bases de Datos relacionales.

**Subclase**

Clase descendiente de otra clase de la que hereda métodos y variables.

**Superclase**

Clase de la cual heredan sus métodos y variables otras clases denominadas subclases.

***Swing***

Paquete que permite incorporar elementos gráficos en las aplicaciones, de una manera más potente que con el AWT. Aparece en la versión 1.2 del JDK. Es no de los componentes que están incluidos en las Java Foundation Classes, o JFC.

***Thread***

Un "proceso ligero" que puede ser arrancado y utilizado más rápidamente que por un *fork* o *spawn*. Véase también: *fork*, *spawn* y proceso.

**Tiempo de vida**

El número de líneas sobre las que una variable es activa, esto es, el número de líneas entre la primera y la última referencia a la variable.

**Tipo primitivo**

En Java, un tipo de dato que no es un objeto. Los tipos primitivos incluyen caracteres, enteros, número de coma flotante y booleanos.

**UML**

*Unified Modeling Language.*

***Unicode***

Conjunto de caracteres de 16 bits, en lugar de los 8 que soportaba ASCII. Así se pueden representar la mayor parte de los lenguajes del mundo.

***Unified Modeling Language (UML)***

Notación estándar de facto utilizada en el análisis y diseño orientado a objetos, basado en el trabajo de Grady Booch, James Rumbaugh e Ivar Jacobson.

**Vector**

Estructura de datos que coloca un tipo de datos en celdas continuas.

**Verificador de código de byte**

Rutinas en la máquina virtual de Java, que aseguran que las instrucciones en el archivo de clase no violan ciertas restricciones de seguridad.



## **RESUMEN**

La Escuela Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, se ha visto afectada por la rotación de catedráticos en diversos cursos del área profesional. Esto se debe principalmente a que esta labor de docencia es una actividad secundaria, un gran número profesionales que tienen interés en la docencia no necesariamente disponen del tiempo para dedicarle a la misma sin afectar su principal actividad económica.

Durante el segundo semestre de 2006 y primer semestre de 2007, se llevó a cabo una consultoría sobre el curso de Estructura de datos, incluyendo evaluaciones parciales, elaboración de proyectos, evaluación final, evaluaciones de primera y segunda retrasada.

El presente informe es un documento completo con toda la información relacionada con el curso impartido. Se incluye el programa del curso tal y como se presentó a los estudiantes, las claves de todas las evaluaciones realizadas en ambos semestres, los enunciados de los proyectos, los cuadros finales de notas, los resultados de evaluaciones de primera y segunda retrasada.

También se dan recomendaciones que pudieran mejorar el contenido y desarrollo del curso, ya que ese fue uno de los objetivos planteados en el anteproyecto del trabajo de Ejercicio Profesional Supervisado.



## **OBJETIVOS**

### **GENERALES**

- Realizar el análisis del curso Estructura de datos por dos semestres consecutivos, de julio 2006 a agosto 2007.

### **ESPECÍFICOS**

1. Recomendar mejoras al contenido oficial del curso impartido.
2. Motivar al estudiante de la carrera de Ingeniería en Ciencias y Sistemas, mediante charlas informativas sobre de la importancia de este curso para su carrera profesional, dado la reciente expansión del uso de java en todos los ambientes de programación.
3. Analizar el contenido del curso y la forma en que se imparte a lo largo de los dos semestres para poder recomendar formas para asegurar que se transmite la mayor parte posible del contenido a los estudiantes.



## **INTRODUCCIÓN**

El presente informe corresponde al Ejercicio Profesional Supervisado o EPS, desarrollado en la Escuela Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería de la Universidad de San Carlos de Guatemala. Dicho EPS nació de la necesidad en la Escuela de personal docente profesional y con experiencia en áreas de la carrera de Ingeniería en Ciencias y Sistemas. Esta escasez de profesionales que puedan transmitir su experiencia y conocimientos en los diversos cursos, se debe a varios factores como el horario de los cursos en jornada matutina que entran en conflicto con los horarios de la actividad económica principal de los profesionales, la elevada cantidad de estudiantes por curso, la oferta de mejores condiciones de trabajo en universidades privadas, entre otros.

El trabajo de EPS consistió en apoyar de forma directa a la Escuela Ingeniería en Ciencias y Sistemas, llevando a cabo una consultoría sobre el curso de Estructura de datos, en el período de julio de 2006 a agosto de 2007. Dado que se dio un cambio de contenido del curso con respecto al semestre anterior, con la intención de dar continuidad al contenido de Introducción a la programación y computación 2, se planteó como uno de los objetivos para el final del trabajo, hacer recomendaciones que pudieran mejorar o ampliar el contenido del curso.

De este modo se sienta un precedente para que otros profesionales no graduados y con pensum cerrado puedan apoyar a la Carrera de acuerdo con su experiencia, impartiendo cursos que sean de su dominio e interés, para cumplir así con su Ejercicio Profesional Supervisado y optar a graduarse.



# **1. MARCO TEÓRICO**

## **1.1. Importancia de la educación**

La educación se ha convertido en una de las piedras angulares del desarrollo económico. De allí que los países avanzados hagan fuertes inversiones para aumentar el volumen y la calidad de la educación. En el mundo industrializado, la adecuación de los estándares educativos a lo que cada país o grupo de interés considera estratégico es motivo de análisis y discusión permanentes. También es frecuente la comparación entre el número de horas que dedica cada alumno a estudiar en cada país o la cantidad de profesionales con que cada uno cuenta en los campos de mayor demanda en la producción o los servicios. La disponibilidad de ingenieros electrónicos o de especialistas en las diversas ramas de la informática es objeto de especial atención. Por contraste, en los países subdesarrollados, dentro de los que -inevitablemente- se incluye al nuestro, las inversiones en educación son mucho menor en relación con el número de alumnos y el producto nacional, y la calidad de la enseñanza también es inferior.

Además de aumentar los recursos destinados a la educación, es necesario prestar atención a otros aspectos clave como la forma más adecuada de distribuir los fondos, la mejora en la calidad y la búsqueda de una mejor articulación entre sus diferentes niveles. De este modo se logrará una mejor utilización de los recursos sociales. La extensión de la educación opera en dos niveles. En el aspecto humano aumenta las posibilidades de trabajo y de progreso de las personas. Paralelamente, pone a disposición del sector público y privado los recursos humanos que necesitan para desarrollar sus actividades.

El nivel educativo de la sociedad siempre fue importante para el desempeño de las instituciones públicas o privadas, pero esa importancia aumenta con la complejidad de los procesos administrativos o productivos. También, con la velocidad con que se desactualizan los saberes y la necesidad de actualizar la oferta educativa. La valorización de la educación en el desarrollo económico o la creación de empleos no debe hacer olvidar el papel determinante de los demás factores de la producción. La buena formación de la población contribuye a mejorar el ambiente económico, pero no alcanza para generar empleos. En nuestro país existen situaciones en las que los conocimientos no son utilizados o aprovechados debidamente e, incluso, hay exportación de materia gris que no encuentra puestos de trabajo.

## **1.2. Modelo educativo**

“Los modelos educativos son visiones sintéticas de teorías o enfoques pedagógicos que orientan a los especialistas y a los profesores en la elaboración y análisis de los programas de estudios;” <sup>(1)</sup> en la sistematización del proceso de enseñanza-aprendizaje, o bien en la comprensión de alguna parte de un programa de estudios. El conocimiento de los modelos educativos permite a los docentes tener un panorama de cómo se elaboran los programas, de cómo operan y cuáles son los elementos que desempeñan un papel determinante en un programa o en una planeación didáctica.

El conocimiento que se tenga de los programas y de sus partes será determinante para que los docentes elaboren planeaciones didácticas eficientes y obtengan resultados mejores en el aula.

## 1.3. Algunos tipos de modelos educativos

### 1.3.1. El modelo tradicional

Este tipo de modelo educativo se refiere principalmente a la elaboración de un programa de estudios. Los elementos que presentan son mínimos, ya que no se hacen explícitas las necesidades sociales, la intervención de especialistas, las características del educando, ni tampoco se observan las instancias de evaluación del programa de estudios. El esquema es muy sencillo. En él destacan los cuatro elementos siguientes:

•**El profesor.** Es el elemento principal en el modelo tradicional, ya que tiene el papel activo.

•**El método.** Se utiliza cotidianamente la clase tipo conferencia, copiosos apuntes, la memorización y la resolución de los cuestionarios que presentan los libros de texto.

•**El alumno.** Su papel es más bien receptivo, es tratado como objeto del aprendizaje.

•**La información.** Los contenidos se presentan como temas, sin acotar la extensión ni la profundidad con la que deben enseñarse. De esta manera, algunos profesores desarrollan más unos temas que otros creando, por ende, distintos niveles de aprendizaje en grupos de un mismo grado escolar.

El modelo tradicional muestra la escasa influencia de los avances científico-tecnológicos en la educación y, en consecuencia, refleja un momento histórico de desarrollo social. No obstante sus limitaciones, este modelo se tomó como base pedagógica para formar diversas generaciones de profesores y de alumnos.<sup>1</sup>

---

<sup>1</sup> Disponible en:

<http://www.es.catholic.net/educadorescatolicos/694/2418/articulo.php?id=22081>

### 1.3.2. El modelo de Ralph Tyler

El modelo que Tyler propone, presenta como aportación fundamental el concepto de objetivos, los cuales se convierten en el núcleo de cualquier programa de estudios, ya que determinan de una manera u otra el funcionamiento de las otras partes del programa. La idea de elaborar un programa o una planeación didáctica teniendo como base a los objetivos, cambia sustancialmente el esquema tradicional de las funciones del profesor, del método, del alumno y de la información, por ejemplo:

•**El profesor.** Sus acciones están determinadas por el objetivo, puesto que señala con claridad la extensión y la profundidad con que se ha de enseñar dicho contenido: Está definida la forma en que el profesor tendrá que impartir la enseñanza y le propone diversas actividades según sea el tipo de objetivo de que se trate.

•**El método.** Como los objetivos mencionan diversas acciones que los alumnos han de desempeñar, la enseñanza no puede dirigirse con un solo método o con una misma forma de dar la clase. Por el contrario, se proponen diversas actividades para los alumnos (actividades de aprendizaje) y actividades para el profesor (actividades de enseñanza), de tal manera que dependiendo el tipo de objetivo serán las acciones a realizar por el docente y los educandos.

•**El alumno.** Los objetivos mencionan acciones que han de realizar los alumnos, por lo cual éstos dejan de ser pasivos u objetos de enseñanza y se convierten en sujetos de aprendizaje realizando diversas acciones que son registradas por el docente. A diferencia del modelo tradicional donde el alumno desconocía la profundidad y extensión de tema, así como las acciones que se esperan de él, en el modelo de Tyler el alumno, desde la lectura del objetivo, conoce las actividades que debe realizar individualmente, en equipo o bien conjuntamente con el profesor.

•**La información.** La información por enseñar ya no se presenta a manera de temas como se hacía en el modelo tradicional, sino por medio de objetivos, es decir, se fragmentan los contenidos en pequeñas porciones, las cuales están acotadas tanto en su extensión como en su profundidad. Un contenido puede dar lugar a varios objetivos con diversas acciones por realizar; dichos objetivos se relacionan y se estructuran lógicamente formando unidades, éstas, a su vez, presentan un orden lógico y una secuencia de lo simple a lo complejo y forman un programa de estudios.

La información así estructurada permite un manejo preciso y homogéneo por parte del profesorado y elimina, en parte, la subjetividad en la enseñanza de los contenidos, ya que las acciones del profesor, del alumno, la extensión, profundidad y tiempo dedicado a cada objetivo están acordados previamente en el programa de estudios.

La planeación didáctica se facilita puesto que el programa de estudios resulta lo suficientemente explícito y el docente sólo necesita hacer un análisis cuidadoso del programa o, en su defecto, consultar al coordinador de área o de estudios.

Otras ventajas que proporciona este modelo son:

•**La evaluación.** Se realiza de manera más sistemática, ya que los tiempos, las formas e instrumentos de evaluación que deben emplearse están predeterminados en el programa de estudios

Los docentes podrán elegir formas alternativas de evaluación con la condición de que se adecuen al objetivo, es decir, que las acciones que el alumno debe realizar, las cuales se mencionan en el objetivo, sean factibles de medir y de registrar.

•**La participación de especialistas.** La elaboración de programas requiere de la participación de especialistas, puesto que se requiere de un conocimiento técnico-pedagógico que demanda rigor y precisión.

La propuesta de objetivos, la selección de los mismos, así como su redacción son tareas complejas que requieren del conocimiento de diversas teorías del aprendizaje, del manejo de diversos métodos y técnicas didácticas y de enfoques taxonómicos de evaluación, entre otros requisitos.

•**La sociedad.** El vínculo entre educación y sociedad se torna más estrecho en el modelo de Tyler, ya que los objetivos sugeridos por los especialistas tienen como marco de referencia las necesidades que demanda la sociedad, de tal manera que, conforme se modifican las necesidades sociales, es necesario cambiar los objetivos de los programas de estudios porque se corre el riesgo de que se vuelvan obsoletos.

Los elementos que el modelo de Ralph Tyler presenta, así como su dinámica, serán comentados por el instructor del curso, de tal manera que los participantes puedan interpretar por sí mismos el programa de estudios que imparten.<sup>2</sup>

### **1.3.3. Modelo de Popham-Baker**

Este modelo se refiere particularmente a la sistematización de la enseñanza; hace una comparación entre el trabajo de un científico y el trabajo de un profesor. La comparación estriba en que el científico tiene un conjunto de hipótesis como punto de partida, selecciona una serie de instrumentos para comprobar su veracidad, con los instrumentos seleccionados somete las hipótesis a experimentación y evalúa los resultados obtenidos.

---

<sup>2</sup> *Disponible en:*

<http://www.es.catholic.net/educadorescatolicos/694/2418/articulo.php?id=22081>

De igual manera, el docente parte de un conjunto de objetivos de aprendizaje, selecciona los instrumentos de evaluación más idóneos y los métodos y técnicas de enseñanza acordes con los objetivos, los pone a prueba durante la clase o en el curso y evalúa los resultados obtenidos.

Desde luego que los niveles de rigor, precisión y conceptualización distan mucho entre un científico y un docente; sin embargo, la propuesta de Popham-Baker es de que en ambos hay sistematización en el trabajo que se realiza, un conjunto de elementos a probar, y la evaluación de resultados, es decir, cada uno de los elementos mencionados ocupa un lugar dentro de una secuencia formando un sistema que tiene una entrada y una salida de productos o resultados, los cuales se modifican por medio de un proceso.

Este modelo incorpora, a diferencia del modelo de Tyler, una evaluación previa de los objetivos de aprendizaje, la cual permite conocer el estado inicial de los alumnos respecto de los objetivos.

Los resultados de la evaluación previa se comparan con los resultados de la evaluación final; de esta manera puede registrarse y compararse el grado de avance en el aprendizaje de los alumnos.

El modelo de estos autores debe motivar a los docentes a realizar planeaciones didácticas rigurosas, bien secuenciadas, y apoyadas con los instrumentos de evaluación más idóneos según los objetivos de aprendizaje que mencionen los programas de estudios que imparten.

El instructor comentará algunos otros aspectos del modelo de Popham-Baker que considere convenientes.<sup>3</sup>

---

<sup>3</sup> *Disponible en:*

<http://www.es.catholic.net/educadorescatolicos/694/2418/articulo.php?id=22081>

### **1.3.4. Modelo de Roberto Mager**

El modelo de Roberto Mager permite a los docentes conocer en detalle una parte importante de los programas: los objetivos.

Los objetivos pueden ser generales, particulares (también llamados intermedios) y específicos (también conocidos con el nombre de operacionales).

Los objetivos han sido estudiados por diversos autores y por consiguiente han surgido distintas nomenclaturas o terminologías.

En el modelo de Mager se hace referencia a los objetivos específicos, es decir, con los que comúnmente opera el profesor en el salón de clase y los que están a la base de su planeación didáctica.

Habitualmente en un programa de estudios los objetivos específicos se presentan redactados, sin hacer mención a cada una de sus partes, para evitar la pérdida de significado o de sentido en el profesor.

Es frecuente que los profesores lean rápidamente los objetivos específicos y no tomen en cuenta todas las acciones y partes que se mencionan en ellos, esto trae como consecuencia que no se distinga con claridad cómo enseñar y evaluar adecuadamente los objetivos. Esta situación impide que se alcancen óptimamente las acciones y los niveles de ejecución que los objetivos demandan.

El modelo de Roberto Mager muestra con claridad las partes que integran un objetivo específico.

•**Presentación:** esta parte se refiere a quién efectuará la conducta solicitada: el alumno, el participante, el practicante, etcétera.

•**Conducta:** se refiere al comportamiento o acción que realiza el alumno o el participante. Por lo regular se redacta utilizando un verbo activo que no dé lugar a diversos significados, por ejemplo: identificar, clasificar, enlistar, etcétera.

•**Contenido:** esta parte hace mención al tema o subtema mediante el cual se logrará el objetivo, por ejemplo: tabla periódica, ecuación lineal, movimiento rectilíneo, huesos del cuerpo humano, efecto invernadero, etcétera.

•**Condiciones:** hace mención a las circunstancias particulares en que la conducta debe manifestarse, por ejemplo: en el laboratorio, en un mapa, con ayuda de un modelo o maniquí, en una maqueta, en la computadora, etcétera.

•**Eficacia:** en esta parte se hace referencia al criterio de aceptabilidad de la conducta, es decir, se hace explícito el nivel o grado de complejidad en que la conducta debe darse.<sup>4</sup>

### 1.3.5. Modelo de Hilda Taba

Este modelo sintetiza los elementos más representativos de los otros modelos que ya se han revisado. Uno de los aportes que presenta es la organización de contenido y las actividades de aprendizaje.

El contenido de un programa de estudios o de una planeación didáctica debe presentar una organización lógica, cronológica o metodológica. Dicha organización permitirá al docente presentar la información a los alumnos de lo simple a lo complejo, de lo que es antecedente a su respectivo consecuente, de la causa al efecto, de lo general a lo particular, etcétera, lo cual redundará en un mejor aprovechamiento.

---

<sup>4</sup> *Disponible en:*

<http://www.es.catholic.net/educadorescatolicos/694/2418/articulo.php?id=22081>

La organización de las actividades también es un factor de mejora en el aprendizaje. Los profesores deben presentar a los alumnos los objetivos mediante una gama de actividades debidamente secuenciadas, considerando cuáles han de ser de manera individual y cuáles de forma grupal, fijando la duración de ambas.

Las actividades que los profesores y los alumnos realizan deben estar claramente diferenciadas y equilibradas, de tal manera que el profesor tenga previsto cuándo exponer, retroalimentar, organizar y supervisar, y en qué momentos el grupo asume el papel protagónico en el aprendizaje y el profesor coordina las actividades y retroalimenta a los alumnos individualmente o a cada uno de los equipos.

La propuesta del modelo de Hilda Taba muestra a los docentes las partes más importantes de un programa y, a su vez, les plantea el reto de elaborar planeaciones didácticas con organización de contenidos y actividades creativas, precisas y eficientes. <sup>5</sup>

---

<sup>5</sup> *Disponible en:*

<http://www.es.catholic.net/educadorescatolicos/694/2418/articulo.php?id=22081>

## **1.4. Dinámica del curso**

En la parte magistral del curso, el objetivo principal es familiarizar a los estudiantes con las diferentes unidades del programa del curso. Para el catedrático esto implicó dar clases magistrales cubriendo el material hasta donde el tiempo lo permitió, dejar tareas orientadas a obligar a los estudiantes a leer el material y familiarizarse con los conceptos de las unidades y dando un peso en puntos a las tareas para motivar a los estudiantes a llevarlas a cabo. En este sentido, la dinámica del curso se aproxima al método de Ralph Tyler, en el sentido de que las acciones del profesor, los estudiantes y el método están determinadas por el objetivo.

El laboratorio tiene contenido propio, planificado por el encargado de laboratorio y el catedrático, enfocado a aspectos prácticos de programación en Java, ejemplos, talleres, y todo lo relacionado con el soporte, revisión y calificación de los proyectos del curso. Adicionalmente, el laboratorio apoya la parte teórica del curso cuando por limitaciones de tiempo no es posible cubrir todo el material en la clase magistral.

## 1.5. Descripción del curso

El curso impartido es Estructura de datos [Código 772], el cual corresponde al quinto semestre del p nsu m de estudios de la carrera de Ingenier a en Ciencias y Sistemas. Adem s corresponde al  rea de Desarrollo de *Software* y tiene un valor de cinco cr ditos, es post-requisito de los cursos Introducci n a la programaci n y computaci n 2 [C digo 771], Lenguajes formales y de programaci n [C digo 796] y Matem tica de c mputo 2 [C digo 962]; y prerrequisito de los cursos Teor a de sistemas 1 [C digo 722], Organizaci n de lenguajes y compiladores 2 [C digo 781], Manejo e implementaci n de archivos [C digo 773].

Tradicionalmente, este es uno de los cursos m s importantes de la carrera, en donde se ense a a los estudiantes a elaborar los conceptos sobre datos, los diferentes tipos de datos y las diferentes estructuras de datos, determinar cu les estructuras de datos son necesarias en un programa, y a analizar el manejo en la memoria del computador de todas las estructuras de datos. Sin embargo, debido a cambios en el pensum de la carrera, los estudiantes se familiarizaron con estos conceptos en cursos anteriores, y ahora en Estructura de datos se les pedir a la aplicaci n pr ctica de los mismos, en ambiente de programaci n java.

### **1.5.1. ¿Por qué Java?**

Java es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general, al grado en ocasiones de ha llegado a denominar como “el lenguaje de Internet” debido a la relevancia que ha conseguido alcanzar en este ámbito. Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. Está desarrollado por la compañía Sun Microsystems y suele ir enfocado a cubrir las necesidades tecnológicas de punta, por lo que se consideró el medio más apropiado para el desarrollo del curso.

Java se distingue de otros lenguajes, en que es una plataforma completa de desarrollo, consta de un gran conjunto de componentes que se pueden reutilizar y mecanismos para extenderlos, facilitando la vida a los desarrolladores. Aunque al mismo tiempo obliga a tener buenas prácticas, buenos patrones de diseño a diversos problemas recurrentes de desarrollo.

La plataforma Java se ha establecido en la industria como una de las principales herramientas de construcción de aplicaciones en las corporaciones, otorgándoles diversos beneficios, así como de un universo de aplicaciones, frameworks y estándares generados alrededor de la plataforma, que la complementan y extienden.

### **1.5.2. ¿Por qué seleccionar a Java como plataforma de desarrollo?**

Aplicaciones que se puedes construir en la plataforma JAVA:

1. Aplicaciones empresariales

- Aplicaciones Web usando tecnologías Java: Servlets, JSP
- Servicios Web
- Aplicaciones cliente/servidor
- Aplicaciones distribuidas

2. Aplicaciones de escritorio con AWT, *Swing*.

3. Aplicaciones móviles

Java posee tres tipos de subestándares, cada una es un mundo de tecnología disponible para desarrollar familias de aplicaciones:

**JSE (*Java Standard Edition*)**. Plataforma base del lenguaje Java donde puedes hacer aplicaciones Stand Alone, tanto JEE y JME se basan en JSE. Es el conjunto de herramientas *software* que permite el desarrollo y la ejecución de programas Java. Es gratuito y de libre distribución. Dentro del JSE se incluyen el compilador y la JVM (*Java Virtual Machine*), que también se conoce como Intérprete de *Java Java Standard Edition*, comprende las librerías básicas del lenguaje, así como las APIs más comunes.

**JEE (*Java Enterprise Edition*)**. Plataforma del lenguaje Java que provee una especificación de cómo debe construirse una aplicación empresarial. Esta especificación describe como los servidores de aplicación deben proporcionar seguridad, escalabilidad, portabilidad, consistencia, manejo transaccional robusto e independencia de la plataforma tanto de hardware como de sistemas operativos, y al mismo tiempo para que el desarrollador final pueda desarrollar aplicaciones empresariales con menor esfuerzo.

**JME (*Java Micro Edition*).** Plataforma del lenguaje Java destinada al desarrollo de aplicaciones para dispositivos móviles. JME es una versión reducida del conjunto de clases de JSE aunque con algunas librerías extra pensando sobretodo en el desarrollo de aplicaciones para dispositivos móviles. Se trata de una edición especial debido a las limitaciones de recursos en diversas familias de dispositivos (celulares, PDAs, Pockets, PCs, televisiones, relojes, sistemas de ayuda para automóviles, tarjetas, robótica).

### 1.5.3. Ventajas de Java

**Licencias:** Java es *Open Source* (*Software* libre o Código abierto). La infraestructura para desarrollar en JAVA es gratuita, aunque algunos proveedores de servidores de aplicaciones o frameworks tienen costo. A partir del 13 de noviembre de 2006, ya es un proyecto completamente libre, tiene la licencia elegida es GPL v2, que cuenta con el apoyo de *Free Software Foundation* (FSF), lo cual permitió contar con una gran variedad de proveedores.

**Multiplataforma.** Es un lenguaje independiente de la plataforma. Una aplicación Java puede correr sobre diversos ambientes sin modificaciones como Windows, Linux, AS400, entre otros.

**Interoperabilidad Multilenguaje.** A partir de la versión 6 JAVA soporta aplicaciones con componentes en otros lenguajes (RUBY).

**Documentación.** JAVA ofrece mucha documentación de ayuda (herramientas, debuggers, editores), apoyado con un gran número de librerías que son soportadas por diversas empresas como IBM, Oracle, Jakarta, además de encontrarse otra gran cantidad desarrolladas por terceros. Esto simplifica el desarrollo y la implementación. De esta

manera, se prevé que las aplicaciones en Java sean cada vez más rápidas, fiables y sencillas de escribir.

**Java es un estándar abierto.** Es abierto a todo el que quiera participar en la evolución de Java, con esa intención se creó el JCP (organismo que controla la evolución de Java), el cual se integra con más de 700 empresas, entre las que se hallan todas las grandes empresas como Sun, IBM, SAP, BEA, IONA, Oracle, Nokia, Siemens, Siebel, Motorola entre otras, es decir Java no pertenece solamente a Sun.

**Reutilización.** Los desarrollos de una aplicación permiten un alto grado de reutilización de componentes, por lo tanto un desarrollo en Java nunca empieza desde cero. Java implementa consigo un gran número de clases, incluidas gratuitamente e su entorno de desarrollo, para realizar muy variadas tareas que permiten al desarrollador centrarse en el negocio y no en la implementación.

**Seguridad.** Provee diversos mecanismos de seguridad tanto a nivel de código (un programa Java no se ve afectado por virus informáticos) como a nivel de autenticación y autorización de usuarios.

**Es multihilo.** El beneficio de ser multihilo consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, etc.), aún supera a los entornos de flujo único de programa (*single-threaded*) tanto en facilidad de desarrollo como en rendimiento.

**Extendible.** Se puede extender para soportar otros servidores de aplicaciones y bases de datos. Adicionalmente, los componentes de *software* generados, se pueden extender acorde a las necesidades requeridas. Los componentes de código generados para los proyectos, se pueden personalizar fácilmente dependiendo de necesidades y estándares particulares de cada aplicación.

**Se puede desarrollar con IDEs muy robustos.** Algunos de ellos *Open-Source* como: *NetBeans* y *Eclipse* .

**Alta conectividad** con bases de datos, ERP, sistemas legados (AS-400, CICS) y otros sistemas.

**Es más solicitado en el mercado de *software*** por su gran cantidad *de frameworks* (*Spring, Struts, Hibernate, etc*) que facilitan el desarrollo de proyectos.

**Escalabilidad.** Puede cambiarse la configuración y fácilmente adaptarlo a nuevas capacidades.

**Soporte.** Esta soportado por empresas como Oracle, BEA, IBM, Sun.

#### **1.5.4. Conclusión**

Java es un lenguaje que permite tener desarrollos bastante completos, ya que es un lenguaje de propósito general con el que se puede programar desde una aplicación Web o distribuida hasta una aplicación Stand Alone sin ninguna conexión a Internet. Por lo que hoy en día, existe una gran variedad de dispositivos que nos rodean en la oficina o en el hogar que tienen Java como teléfonos móviles, impresoras, webcams, automóviles, equipamientos médicos, servidores, motores de aviones, el control de navegación de la Mars Rover de la NASA y muchos otros.

## **Duración del Ejercicio Profesional Supervisado**

La asesoría del curso Estructura de datos se llevó a cabo durante el segundo semestre de 2006 y el primer semestre de 2007. En términos de tiempo, abarcó desde julio de 2006 hasta agosto de 2007, incluyendo la realización de evaluaciones de primera y segunda retrasada de cada semestre.

### **1.6. Atribuciones del catedrático**

Durante el desarrollo del curso impartido en ambos semestres, el catedrático(a) es responsable de:

- Impartir clase magistral para exponer el contenido del curso. El material expuesto fue completo, y se proporcionó documentación electrónica (PDF) de modo que los estudiantes no estuvieran obligados a adquirir el libro de texto.
- Elaborar las evaluaciones parciales, evaluaciones de reposición, evaluación final, evaluaciones de primera y segunda retrasada. Este material está incluido en los apéndices del presente documento.
- Calificar las evaluaciones anteriormente mencionadas.
- Elaborar enunciados de proyecto y juego de datos para estudiantes y ayudante de cátedra.
- Publicar notas de evaluaciones, proyectos, tareas, etc. en el sitio de la Universidad Virtual, para que los estudiantes puedan saber su zona parcial y sus posibilidades de aprobar el curso.

Estas tareas podrían o no realizarse conjuntamente con el catedrático(a) de la otra sección del mismo curso. De esta forma se optimizan esfuerzos en la preparación de evaluaciones y proyectos.

Por su parte, el (la) ayudante de cátedra o auxiliar es el (la) responsable de:

- Impartir laboratorio del curso, el cual cuenta con contenido propio, enfocado a aspectos prácticos de programación en Java.
- Calificar y tabular pruebas cortas realizadas por el catedrático o por él (ella) mismo(a).
- Calificar y tabular proyectos.
- Publicar notas de pruebas y evaluaciones de laboratorio en el sitio de la Universidad Virtual, en tiempo prudencial.
- Proporcionar notas de laboratorio y de proyectos al catedrático(a) en tiempo prudencial.

## **1.7. Casos especiales**

Con frecuencia ocurre que hay estudiantes para los cuales es necesario llevar un control adicional a sus notas durante el semestre. Estos casos incluyen:

- Estudiantes que no aparecen asignados a la sección, pero sí están cursando en la misma. Al final si el estudiante aprueba el curso, debe solicitar una autorización a Junta Directiva de la Facultad, para ser agregado en las actas oficiales del curso y sección correspondiente.

- Estudiantes que están asignados en una sección, pero llevan el curso en otra sección por diferencias de horario o por preferencia en la forma como imparte el curso otro catedrático. Al final del semestre se debe solicitar al catedrático de la otra sección, que proporcione la zona del curso y la nota de examen final, para incluirlas en los cuadros finales. La nota de laboratorio puede ser también proporcionada por el auxiliar de la otra sección o de la sección correspondiente.
- Estudiantes que ya han aprobado el laboratorio del curso con anterioridad. El laboratorio tiene una vigencia de un año y la nota aparece automáticamente en el sistema de Control Académico, cuando el catedrático procede a cargar las notas por Internet.

## **1.8. Publicación de notas**

Las notas de evaluaciones, pruebas cortas, tareas, proyectos, etc. se deben publicar en el sitio de la Universidad Virtual, propio de la Escuela Ingeniería en Ciencias y Sistemas. Tanto catedráticos como ayudantes de cátedra tienen acceso a este sitio por medio de un usuario y una clave. Los estudiantes también tienen la obligación de solicitar acceso a este sitio, para consultar sus notas y poder ser notificados por correo electrónico de la publicación de notas, enunciados de proyectos, tareas, etc.

Se pueden publicar documentos en cualquier formato, siendo los más comunes archivos de Microsoft Excel, texto, HTML, PDF, Microsoft Word. La dirección URL para acceder al sitio de la Universidad Virtual es:

<http://sistemas.ingenieria-usac.edu.gt>

Además, al final del semestre hay un período durante el cual cada catedrático puede cargar las notas o cuadros finales en el sitio de Control Académico de la Facultad de Ingeniería. La Escuela es la encargada de solicitar el acceso para cada catedrático, por medio de un usuario y un PIN.

Las notas se pueden ingresar manualmente en el sitio, o se pueden cargar mediante un archivo en formato CSV con un encabezado específico. Luego Control Académico imprime las actas (cuadros) y los envía a cada Escuela, en este caso la Escuela de Sistemas. El catedrático debe pasar a la Escuela a firmar estas actas, pudiendo hacer correcciones o agregar estudiantes que no estaban asignados al curso pero lo aprobaron y Junta Directiva de la Facultad les autorizó dicha aprobación. La dirección URL para ingresar al sitio de Control Académico es:

<http://ingenieria-usac.edu.gt>

## **1.9. ITFORCEGT**

Durante el desarrollo de ambos semestres, la Dirección de la Escuela Ingeniería en Ciencias y Sistemas solicitó a los catedráticos que promocionaran el sitio de ITFORCEGT entre los estudiantes. Para ello se definió como un requisito para aprobar el curso, que cada estudiante publicara su *home page* en este sitio, incluyendo información personal, como una especie de *curriculum vitae* u hoja de vida totalmente en inglés. Si el estudiante ya había publicado su información, para el siguiente semestre se le solicitaba mejorar la página, tanto en aspecto funcional y estético, como en contenido en inglés. La dirección URL para ingresar a la página de cada estudiante es:

[http://itforcegt.org/<numero\\_de\\_carnet>](http://itforcegt.org/<numero_de_carnet>)

En el Anexo 2 se muestra el documento original que fue proporcionado por la Escuela Ingeniería en Ciencias y Sistemas, sobre el proyecto ITFORCEGT.

## **2. EVOLUCIÓN DEL CURSO SEGUNDO SEMESTRE 2006**

### **2.1. Descripción del curso**

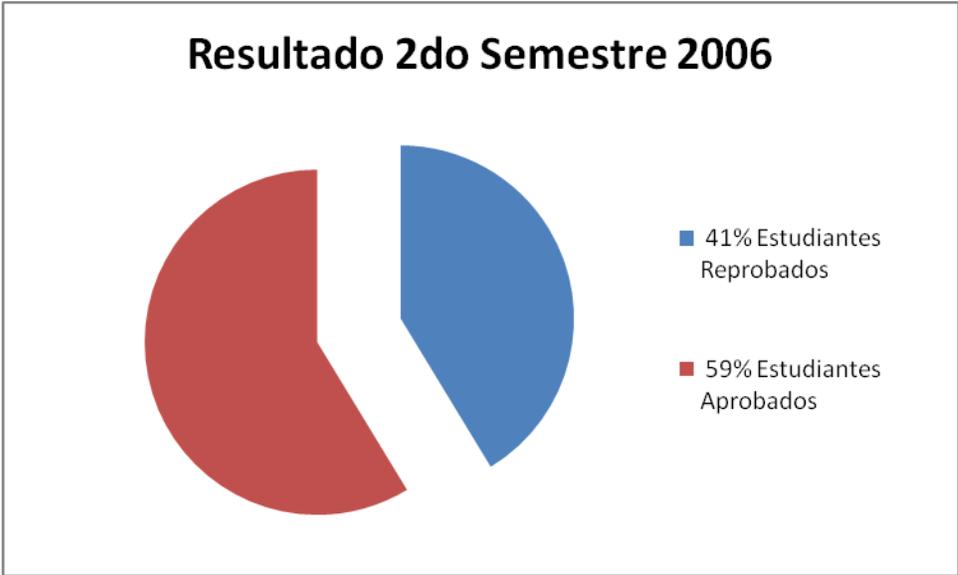
Para el segundo semestre de 2006, hubo tres secciones de Estructura de datos en el horario de viernes de 7:00 a 9:00 h y sábado de 9:00 a 11:00. Cada sección contó con un catedrático y un auxiliar, quienes realizaron conjuntamente todas las evaluaciones parciales, final y de retrasadas, además de los proyectos.

### **2.2. Distribución de la nota**

Durante este semestre, la nota se distribuyó de la siguiente forma:

dos evaluaciones parciales a 15 puntos cada uno	30 puntos
ITFORCEGT y otras tareas en clase	10 puntos
Evaluación final	20 puntos
Laboratorio	
dos proyectos a 15 puntos cada uno	30 puntos
Tareas de laboratorio	10 puntos
TOTAL	100 puntos

Se tomó como muestra la sección B para ilustrar la información de estudiantes que aprobaron el curso. Al final del curso aprobaron 27 estudiantes de un total inicial de 46. En el Anexo 1 se muestra el material desarrollado y utilizado para este semestre: claves de evaluaciones parciales y final, enunciados de proyectos, cuadro de notas finales, exámenes y notas de retrasadas.



**Porcentajes de estudiantes aprobados y reprobados**

## **2.3. Principales Hallazgos**

### **2.3.1. Problema del lenguaje**

Una de las principales dificultades encontradas para impartir el curso fue el hecho de que solo un pequeño porcentaje de los estudiantes lee y comprende inglés técnico.

Esto representa un enorme impedimento para los estudiantes considerando que la gran mayoría del material de referencia disponible (no solamente sobre Java sino sobre todo tema de informática) está en inglés.

A pesar de lo anterior, se decidió que todas las evaluaciones se redactaran en inglés, no solamente para evitar que la traducción de términos pudiera confundir a los estudiantes, sino que al mismo tiempo para motivarlos a mejorar su conocimiento del inglés.

Para facilitar la comprensión de los términos de los exámenes, se decidió utilizar el formato de selección múltiple. Inicialmente se permitió el ingreso de diccionarios y otros medios de traducción electrónica, pero después de un par de incidentes en el primer parcial se decidió no permitir el ingreso de cualquier material para ayudar a la traducción.

### **2.3.2. Tiempo insuficiente para impartir la clase magistral, por varios factores**

Durante este semestre hubo retraso en las clases programadas, por las actividades del congreso COECYS'06, realizado la semana del 18 al 22 de septiembre. Durante esta

semana se suspendieron clases, exámenes, etc., para que los estudiantes participaran en el congreso.

Debido al volumen del material a impartir, el tiempo para impartir el curso es bastante apretado, no hay tiempo para dar ejemplos extensos, y debido a estas clases perdidas, las últimas unidades se cubrieron de forma bastante rápida, dejando lecturas y elaboración de resúmenes y preguntas acerca de las mismas como tareas, para obligar a los estudiantes a leer el material que no se pudo cubrir en clase.

### **3. EVOLUCIÓN DEL CURSO PRIMER SEMESTRE 2007**

#### **3.1. Descripción del curso**

Para el primer semestre de 2007, hubo tres secciones de Estructura de datos en el horario de viernes de 7:00 a 9:00 h y sábado de 9:00 a 11:00. Cada sección contó con un catedrático y un auxiliar, quienes realizaron conjuntamente todas las evaluaciones parciales, final y de retrasadas, además de los proyectos.

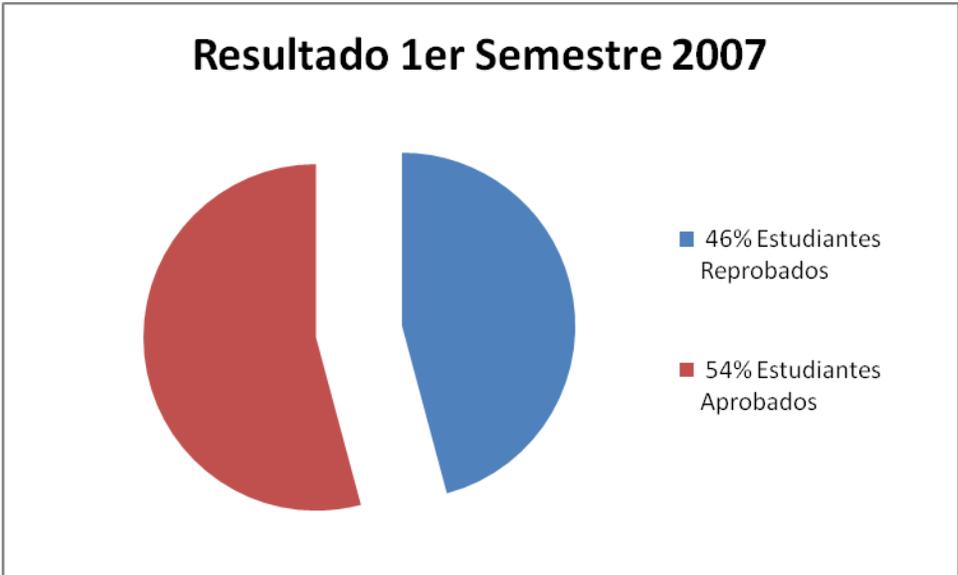
Se apoyó el proyecto ITFORCEGT, solicitando a los estudiantes que mejoraran su página en diseño y principalmente ampliando y mejorando el contenido en inglés.

#### **3.2. Distribución de la nota**

Durante este semestre, la nota se distribuyó de la siguiente forma:

tres evaluaciones parciales a 10 puntos cada una	30 puntos
ITFORCEGT y otras tareas en clase	10 puntos
Evaluación final	20 puntos
Laboratorio	
un proyectos a entregar por etapas	30 puntos
Tareas de laboratorio	10 puntos
TOTAL	100 puntos

Se tomó como muestra la sección B para ilustrar la información de estudiantes que aprobaron el curso. Al final del curso aprobaron 25 estudiantes de un total inicial de 48. En el Anexo 2 se muestra el material utilizado para este semestre: claves de evaluaciones parciales y final, enunciados de proyectos, cuadro de notas finales, exámenes y notas de primera y segunda retrasada.



**Porcentajes de estudiantes aprobados y reprobados**

**3.3.**

## **Principales hallazgos**

### **3.3.1. Conocimiento de idioma Inglés**

Al igual del semestre anterior, una de las principales dificultades encontradas para impartir el curso fue el hecho de que sólo un pequeño porcentaje de los estudiantes lee y comprende inglés técnico. A pesar de lo anterior, nuevamente se decidió que todas las evaluaciones se redactaran en inglés, para evitar que la traducción de términos pudiera confundir a los estudiantes. Nuevamente, para facilitar la comprensión de los términos de los exámenes, se decidió utilizar el formato de selección múltiple.

Aún cuando el curso de Estructura de Datos y la Programación en JAVA tiene que ver más con aspectos meramente técnicos y no directamente con idiomas, el tener un dominio del inglés supone una necesidad más que un lujo: Para los informáticos es obligatorio poseer un inglés funcional que le ayude a alcanzar una mejor curva de aprendizaje y a realizar mejor su trabajo. A continuación cinco razones por las que como programador (y como informático) el estudiante debe tener un inglés funcional o como mínimo un dominio básico de este idioma:

1) El vocabulario informático es de origen anglo.

Los países de habla inglesa han sido históricamente los precursores en el desarrollo de tecnologías, principalmente en el área de la Informática y la Computación. Como potencias tecnológicas han contribuido con innumerables descubrimientos e inventos, los cuales son la base de estas ciencias tal y como las conocemos. Por tal razón, casi la totalidad del vocabulario técnico que se utilizan tiene su fundamento en el idioma inglés.

El influjo de inglés en la informática y la computación tiene grandes implicancias para

nosotros, los de habla no inglesa: las interfaces de *software*, su respectiva documentación, los libros de texto, la sintaxis de los lenguajes de programación, gran parte de los recursos de la Internet y la terminología de cada nueva tecnología que sale al mercado está, al menos en su primera versión, en inglés, por ser el idioma de facto de estas ciencias. La traducción a otros idiomas de los elementos antes mencionados (cuando es posible hacerla) es a menudo insatisfactoria por la riqueza de vocabulario técnico que posee el inglés y la carencia de términos equivalentes en el resto de lenguas.

El inglés es entonces el "*unoficial language*" que los informáticos usan para publicar, comunicarse, conversar sobre tecnología, buscar información, programar, etc. Es evidente entonces que si no se tiene un dominio básico del idioma inglés, mucho del esfuerzo dedicado a aprender, enseñar, trabajar o utilizar tecnología computacional se desperdiciará o no llegará a obtener beneficios rápidos y tangibles.

2) La sintaxis de los lenguajes de programación incluyen palabras en inglés.

Exceptuando un pequeño porcentaje desarrollado por ciertos países y con fines específicos, la inmensa mayoría de los lenguajes de programación utilizan palabras del idioma inglés como parte de su sintaxis. Este conjunto de vocablos llamados *keywords* o palabras reservadas tienen un significado semántico propio, similar al que poseen en el inglés común, lo cual facilita el aprendizaje rápido de la programación así como la comprensión efectiva de la lógica implícita de los programas.

3) Los mensajes de error de los compiladores están en inglés

La aparición en los últimos años de versiones de compiladores en idiomas distintos al inglés y la flexibilidad de poder configurarlos o actualizarlos para que comuniquen sus

hallazgos y reportes en otras lenguas ha hecho que esta afirmación pierda validez. A pesar de ello, existen aún muchos compiladores que sólo "hablan" inglés y no es remoto que algún día se deba obligatoriamente trabajar con uno de estos.

4) La mayoría de las interfaces de desarrollo (IDE) están en inglés.

Aunque cada vez es más frecuente la aparición de versiones traducidas de las IDE (*Interface Development Environment*) así como la facilidad para configurarlas en otros idiomas, aún existe una gran cantidad de estas herramientas en inglés, lo que incide bastante en el lento aprendizaje de la programación y hasta en la productividad y eficiencia de muchos programadores.

5) Más de la mitad de los recursos de la Internet se encuentran en inglés.

Como programadores la Internet nos ofrece una inmensa cantidad de recursos que podemos utilizar para ampliar nuestros conocimientos o resolver problemas específicos de programación. Pero a menudo lo que necesitamos no siempre está en nuestro propio idioma. De hecho, los mejores documentos y artículos técnicos, foros, libros, ejemplos y herramientas están en inglés. Aunque cada día aparece en la Red más material en nuestro idioma natal, no siempre está correctamente traducido y puede conducir a serios errores de aprendizaje o aplicación.

Además de las razones relativas a la preparación y futuro del estudiante como profesional, queda bastante claro que como programador e informático el estudiante debe poseer un inglés funcional, con un buen dominio del vocabulario técnico informático. Por lo menos debe saber lo suficiente como para poder comprenderlo y expresarse apropiadamente con él. El obtener algún nivel académico en este idioma le será de mucha ayuda, pero aún si no logra completarlo, los conocimientos que adquiera le serán de gran utilidad pues podrá documentarse más, realizar mejores búsquedas en la Red, programar, expresar con más claridad y fluidez sus dudas en los foros, y traducir mejor los manuales técnicos.

### **3.3.2. Tiempo insuficiente para impartir la clase magistral, por varios factores**

Durante este semestre hubo retraso en las clases programadas, por las actividades de la Huelga de Dolores y la Semana Santa. Durante este período no se impartieron clases, aunque sí se dejó la primera fase del proyecto 1, para que los estudiantes tuvieran suficiente tiempo. Debido al volumen del material a impartir, el tiempo para impartir el curso es bastante apretado, no hay tiempo para dar ejemplos extensos, y nuevamente debido a estas clases perdidas, las últimas unidades se cubrieron de forma bastante rápida, dejando lecturas y elaboración de resúmenes y preguntas acerca de las mismas como tareas, para obligar a los estudiantes a leer el material que no se pudo cubrir en clase.

# CONCLUSIONES

## **Material del curso**

El material a cubrir en el curso es muy extenso, tanto el material referente a JAVA como el material propio a las diferentes estructuras de datos, y debido a varios factores externos, no se cuenta con suficiente tiempo de clase magistral en el curso para cubrirlo.

Debido a lo extenso del material, se dio el caso de que en clase se enfatizó la parte teórica y conceptual, las últimas unidades del programa se cubrieron muy rápidamente, los estudiantes tuvieron que leer el material por su cuenta, y además se utilizará en forma exclusiva el laboratorio del curso para proporcionar la experiencia práctica.

## **Contenido del laboratorio**

No existe un contenido oficial para el laboratorio del curso, en los dos semestres los auxiliares diseñaron el contenido, asesorados por el catedrático.

Además, dicho laboratorio debería ser completamente configurable, extensible y dinámico para que se ajuste a los enfoques que los diferentes catedráticos pueden dar a dicho curso. Un catedrático debería poder agregar una nueva estructura de datos al laboratorio, o plantear un experimento a los estudiantes, o simplemente adicionar una nueva manera de visualizar un resultado.

## **Deficiencia de conocimiento del idioma Inglés**

Un alto porcentaje de los estudiantes no tienen un suficiente conocimiento del idioma Inglés para acceder a material de referencia actual disponible en este idioma en diversas fuentes, principalmente el Internet, por lo que no cuentan con todas las herramientas y ventajas para el aprovechamiento del curso.

## RECOMENDACIONES

### Contenido del curso

El contenido del curso es muy extenso y el tiempo disponible para poder impartir clases magistrales es a penas suficiente para cubrir el material, y cualquier interrupción por congresos, feriados, etc., disminuye drásticamente el tiempo disponible y hace necesarias clases adicionales para tratar de ponerse al día, dejar trabajos de investigación, lecturas para que los estudiantes puedan complementar fuera de clase los temas que no se pueden cubrir en el aula por falta de tiempo. Se recomienda que las últimas dos unidades del programa incluyan:

#### UNIDAD 6

- Patrones de Diseño Java (Java Patterns)
- Struts (MVC)

#### UNIDAD 7

- Otras tecnologías
- Ant
- Maven
- XDoclet
- Hibernate
- 

Además que no se impartan en clase, sino que sean trabajos de investigación anunciados desde el principio del curso, y que sean incluidos dentro de las evaluaciones parciales y final.

Es necesario analizar el contenido del curso y hacer una nueva planificación para asegurarse de cubrir los puntos más importantes del material.

## **Idioma Inglés**

Requerir a los estudiantes que lean y comprendan material de estudio en inglés. Es importante hacer énfasis en el aprendizaje del idioma Inglés para que los estudiantes tengan acceso al material más reciente y puedan aprovechar la ventaja que presenta java como “lenguaje internet”.

## **Planificación contenido laboratorio**

- Aprovechar el espacio de laboratorio para que, además de proporcionar la práctica al estudiante, pueda complementar el contenido que no se pueda cubrir en la clase magistral debido al corto tiempo disponible.
- Definir objetivos específicos para cumplirse en el laboratorio de Estructura de datos, para proporcionar a los auxiliares una guía para la elaboración de la planificación del laboratorio.

## **Enfoque del curso**

Asegurarse que, al tiempo de que se cubre el material pertinente al lenguaje JAVA, se consigan los siguientes objetivos:

- Que el estudiante entienda la importancia de la programación con tipos abstractos de datos (TAD) como base del diseño modular, diferenciando los conceptos de especificación e implementación de un TAD.
- Dar a conocer al estudiante las estructuras de datos fundamentales y los algoritmos principales que se utilizan para su manipulación.

- Que al finalizar el curso, el estudiante tenga la capacidad de elección de la estructura adecuada para cada tipo de problema.
- Que el estudiante pueda decidir entre dos o más soluciones algorítmicas similares sobre la base de la eficiencia de cada una de ellas.
- Que el estudiante aprenda a combinar diferentes estructuras de datos para resolver problemas complejos.

### **Objetivos específicos para el laboratorio del curso**

Es recomendable que se definan objetivos específicos para el laboratorio, y que la planificación del curso de las actividades del laboratorio apoyen en forma directa el plan de estudios del curso.



## BIBLIOGRAFÍA

1. Ball, Jennifer & Jendrock, Erick, **The Java EE 5 Tutorial** ©2006, **The Java™ EE 5 Tutorial for Sun Java System Application Server Platform**, 3ra. Edición, Estados Unidos: Prentice Hall PTR, 2006, 1360 páginas.
2. Burke, Eric M. & Brian M. Coyner, **Java™ Extreme Programming Cookbook**, Estados Unidos: O'Reilly Media,m, 2003, 352 páginas.
3. **Contenedores de Inversión de Control y el patrón de Inyección de Dependencias**  
'[http://www.programacion.com/java/articulo/jap\\_injection/](http://www.programacion.com/java/articulo/jap_injection/)',  
descargado en junio/2006.
4. **Java desde cero**,  
'<http://www.cybercursos.net>',  
descargado en junio/2006.
5. Lemay, Laura & Charles L. Perkins, **Teach Yourself JAVA in 21 days**, Professional Reference Edition (Sams Teach Yourself), Estados Unidos: Sams, 1996, 1296 páginas.
6. **Modulo 1 – El Lenguaje Java**,  
'[http://seba.wikidot.com/local--files/java/java\\_clase1.pdf](http://seba.wikidot.com/local--files/java/java_clase1.pdf)',  
descargado en junio/2006.
7. **¿Qué es Java?**,  
'<http://www.coit.es/publicac/publbit/bit122/quees.html>',  
descargado en junio/2006.

8. **Tu primera taza en Java (Palos, Juan Antonio),**  
[http://biblioteca.vitanet.cl/colecciones/000/005/primera\\_taza.pdf](http://biblioteca.vitanet.cl/colecciones/000/005/primera_taza.pdf) ,  
descargado en Junio/2006
  
9. Zukowski, John, **Java 6 Platform Revealed**, Estados Unidos: Apress, 2006, 240 páginas.

## **ANEXOS**



## **ANEXO 1: PROGRAMA DEL CURSO**

**VIGENTE EN JULIO DE 2006**

**FUENTE: ESCUELA INGENIERÍA EN CIENCIAS Y SISTEMAS**

<b>No</b>	<b>Código 772</b>	<b>Créditos 5</b>	
<b>1</b>	<b>Escuela</b> <b>Ciencias y Sistemas</b>	<b>Área a la que pertenece</b> <b>Desarrollo de software</b>	<b>Vigencia</b> <b>2º Semestre 2006</b>
<b>2</b>	<b>Horas por semana</b> <b>4</b>	<b>Horario</b> <b>Viernes de 7:00 a 9:00,</b> <b>Sábado de 9:00 a 11:00</b>	
<b>3</b>	<b>Pre-requisitos</b> “Introducción a la programación y computación 2” [Código 771] “Lenguajes formales y de programación” [Código 796] “Matemática de Cómputo 2” [Código 962]		
<b>4</b>	<b>Postrequisitos</b> “Teoría de Sistemas 1” [Código 722] “Organización de Lenguajes y Compiladores 2” [Código 781] “Manejo e implementación de Archivos” [Código 773]		
<b>5</b>	<b>Sección: “A”</b>		
<b>6</b>	<b>I. Descripción General</b> Este curso continua el estudio formal sobre varias técnicas de representación de los datos en la memoria de una computadora y de los algoritmos que los manipulan. Se enfatiza en las características de una buena programación: modularidad, ocultamiento de información, reutilización de código y estilo de programación. Todas estas características se estudian en función de los conceptos de Tipo de Dato Abstracto y Objeto, haciéndose uso de la metodología Orientada a objetos en ambiente Web.		

## **II Objetivos**

Conocer el funcionamiento y saber escribir los algoritmos que manipulan las estructuras de datos más utilizadas.

Utilizar la metodología orientada a objetos para la solución de problemas que utilizan las diferentes estructuras de datos, independientemente de cualquier lenguaje de programación específico.

Implantar las diferentes estructuras de datos estudiadas en la computadora utilizando un lenguaje de programación y siguiendo la metodología orientada a objetos para ambiente WEB

## **III Contenido**

### **ESTRUCTURA DE DATOS**

#### *PROGRAMA DEL CURSO*

#### **UNIDAD 0**

**J2SE y otros Java APIs**

**Java Framework Collection (1)**

**Java Naming and Directory Interface (JNDI)**

**Java Database Connectivity (JDBC)**

**Java Cryptography Extensión (JCE)**

**Java Mail**

**Java Media Framework (JMF)**

**Java Advanced Imaging (JAI)**

#### **UNIDAD 1**

**La capa web (The web tier)**

**Java Servlets**

**Java Server Pages (JSP)**

**Java Standard Tag Libraries (JSTL)**

**Custom Tag Libraries**

**Java Server Faces (JSF)**

**Internationalization**

**AIAX**

**UNIDAD 2**

**Servicios web (Web services)**

**Web Services with JAX-WS**

**XML Schema and Java Classes (binding)**

**JAXB**

**Streaming API for XML**

**SOAP**

**Java API for XML Registries**

**UNIDAD 3**

**Enterprise Beans (8)**

**Entity Beans**

**Session Beans**

**Message Driven Beans**

**Packaging**

**Deploying**

**UNIDAD 4**

**Persistencia (Persistence) (8)**

**Entities**

**Persistence in The Web Tier**

**Persistence in The EJB Tier**

**Query Language**

**UNIDAD 5**

**Servicios (Services) (7)**

**Securing Web Applications**

**Securing Web Services**

**Java Message Services (JMS)**

**Transactions**

**Resource Connections (6)**

**UNIDAD 6**

**Patrones de Diseño Java (Java Patterns) (5)**

**Struts (MVC)**

**UNIDAD 7**

**Otras tecnologías (4)**

**Ant (2)**

**Maven (2)**

**XDoclet (2)**

**Hibernate (3)**

**COMENTARIOS**

(1) El tema del Java Framework Collection debe ser la base del curso, pues es un API que contiene clases para el manejo de distintas estructuras de datos, listas, tablas hash, mapas, sets, listas ordenadas, etc. No tiene clases de manejo de árboles pero en ese punto se pueden tratar este tipo de estructuras.

(2) Herramientas para la automatización del proceso de desarrollo de software (basado en XML).

(3) Mapeo de objetos a tuplas, clases y tablas (basado en XML).

(4) Podría darse en el laboratorio.

(5) Puede ser tema de análisis y diseño.

(6) Me parece interesante que se vea, pues al implementar un pool de conexiones deben de utilizar alguna estructura de datos según lo que necesiten realizar.

(7) Se pueden omitir los servicios de seguridad pues sería tema de redes, análisis y diseño.

También se podría omitir el de transacciones y verse en bases de datos.

(8) Pueden ser temas de algún curso de desarrollo Web avanzado.

La unidad 0 puede darse en el laboratorio, paralela al curso.

**LABORATORIO**

**Proyecto 1: sobre la unidad 1 y 2.**

**Proyecto 2: sobre la unidad 3 y 4.**

**Proyecto 3: sobre la unidad 5, 6 y 7.**

<p>Todos los proyectos deben utilizar la unidad 0 para desarrollar.</p> <p><b>IV Metodología</b></p> <p>El curso se desarrollará intercalando clases magistrales para la exposición de conceptos y clases participativas en las que se asume que el estudiante realiza las lecturas, tareas o ejercicios, previo al inicio de un nuevo día de clases.</p> <p><b>V Evaluación</b></p> <p>2 evaluaciones de rendimiento (15 puntos c/u) ..... 30 pts  Tareas, investigaciones, cortos, asistencia, en clase ..... 10 pts  2 proyectos (15 puntos c/u) ..... 30 pts  Tareas, investigaciones, cortos, asistencia, en laboratorio..... 10 pts  Evaluación final ..... <u>20 pts</u>  Nota total .....100 pts</p>		
7	Bibliografía	*****
8	No. de secciones	3
9	Catedráticos titulares y auxiliares	Titular: Ing. Rene Ornelyz Auxiliar: Mario Gaitán.
10	Coordinador de área	
11	Director de Escuela	Ing. Jorge Armín Mazariegos



## ***ANEXO 2: DOCUMENTO SOBRE ITFORCEGT***

**SEGUNDO SEMESTRE 2006**

**FUENTE: ESCUELA INGENIERÍA EN CIENCIAS Y SISTEMAS**

---

*Guatemala, 08 de julio de 2006*

### **Proyecto “ITFORCEGT.ORG”**

Debido a la importancia del HTML en la creación de páginas WEB y con el afán de que los estudiantes de la carrera Ingeniería en Ciencias y Sistemas de la Universidad de San Carlos de Guatemala, se familiaricen y conozcan este lenguaje, se ha definido la creación del proyecto ITFORCEGT.ORG (Fuerza de Informática GT).

### **OBJETIVO**

El objetivo de este proyecto es que el estudiante se familiarice con HTML y FTP, para esto se ha adquirido el dominio **www.itforcegt.org** y en este dominio se publicará la página de cada estudiante en inglés, de tal forma que se pueda promocionar nuestra fuerza informática guatemalteca. El futuro y visión del sitio [www.itforcegt.org](http://www.itforcegt.org) es convertirse en el sitio guatemalteco [www.rentacoder.com](http://www.rentacoder.com). En esta primera etapa iniciaremos básicamente con páginas promocionando a nuestros estudiantes. Las reglas para desarrollar esta página se describen más adelante en la sección DESCRIPCION.

### **CURSOS INVOLUCRADOS**

Esta iniciativa es **OBLIGATORIA** para los siguientes cursos

- Introducción a la computación y programación I

- Introducción a la computación y programación II
- Estructura de datos
- Manejo e implementación de Archivos
- Bases de datos 1
- Bases de datos 2
- Análisis y diseño de Sistemas I
- Análisis y diseño de Sistemas II
- Software Avanzado

## **DESCRIPCION**

El proyecto consiste en que el estudiante deberá diseñar y desarrollar un sitio WEB personal utilizando HTML, en este sitio podrá mostrar información general acerca de su persona, sus estudios, sus hobbies, sus deportes favoritos, tecnología de hardware y software de interés y podrá mostrar sitios turísticos de Guatemala que le parezcan interesantes. El contenido de la página será 100% en inglés. El diseño de la página WEB **será libre** y el estudiante podrá realizarlo sin ninguna restricción o limitación, lo importante es que el estudiante explote su creatividad en el diseño y contenido de la página.

La herramienta de desarrollo también será libre, podrán elegir cualquier herramienta que permita desarrollar archivos HTML (Dreamweaver, Macromedia Flash, Word, Frontpage, etc.).

Cada estudiante dispondrá de 1 MB de espacio para crear su propio sitio.

El carácter de este proyecto es incremental, es decir, la primera vez el estudiante creará su página WEB personal, la siguiente vez, la actualizará y mejorará, siendo el reto,

incentivar que el estudiante en cada *actualización mejore y aprenda nuevas técnicas que le permitan conocer más acerca de HTML.*

Cada estudiante deberá acceder por medio de FTP su página antes del 30 de septiembre 2006. La información mínima que el sitio de cada estudiante debe contener es:

- Carne del estudiante
- Nombre del estudiante
- Listado de cursos y secciones en que será válido el trabajo (ya que podría estar congelando un curso y llevando el siguiente y en ambos cursos se requerirá este proyecto)
- Descripción general del contenido de la página (por ejemplo, información personal, educación y lago de Amatlán, o puede ser información personal, tecnología dual core de procesadores, etc.).
- *Información de contacto es optativa. Se sugiere colocar correo electrónico.*

No existe NINGUNA LIMITANTE en cuanto a la tecnología a utilizar. Cualquier tecnología para manejo de contenido “estático” es válida como por ejemplo Macromedia Flash, Java Applets, HTML plano con o sin frames, etc. Si existieran algún tipo de requerimientos específicos a nivel del cliente el estudiante debería ser capaz de indicarlos al momento de acceder su página. **En cursos básicos como IPC1, IPC2, Estructuras de datos, la exigencia es mínima y se recomienda simplicidad en el uso del HTML por medio de herramientas de generación de código HTML como Word o Frontpage.**

El estudiante tendrá hasta el **día 30 de septiembre 2006**, para publicar vía FTP su sitio al WEB, este procedimiento es definido en la sección “*ACCEDIENDO AL SITIO*”.

**Está prohibido incluir temas o fotografías que riñan con la moral y las buenas costumbres y no se podrá incluir ningún tipo de información que pueda ser considerada racista, tampoco deben incluirse temas religiosos ni políticos en el contenido de las páginas que desarrolle el estudiante.**

### **DESCRIPCION DEL SITIO**

El sitio será accedido con el siguiente URL: [www.itforcegt.org](http://www.itforcegt.org) Este sitio estará dividido en accesos que tendrán la siguiente forma: [www.itforcegt.org/#####](http://www.itforcegt.org/#####), donde ##### es el número de carné del estudiante, así, por ejemplo, para acceder al sitio personal del estudiante con carné 200112521, deberá utilizar el URL [www.itforcegt.org/200112521](http://www.itforcegt.org/200112521)

### **ACCEDIENDO AL SITIO**

Todos los estudiantes en los cursos mencionados arriba tendrán **un usuario FTP** para acceder al sitio [www.itforcegt.org](http://www.itforcegt.org). Los usuarios y claves de acceso serán otorgados uno a uno cada estudiante avocándose a la escuela durante el periodo comprendido del 10 de Julio de 2006 al 1 de Agosto de 2006 en el siguiente horario:

**Lunes a Viernes: 9 a.m. a 11:00 a.m.**

Ejemplo:

El estudiante Rene Alcázar con carné 9761622 tendría que ir a la escuela de sistemas con el Ing. Pedro Pablo Hernández y/o el Auxiliar Luis Alvarado para obtener su usuario y clave de acceso. Una vez que ha obtenido esta información, el estudiante podrá acceder su espacio disponible (1 MB) mediante una herramienta FTP proveyendo la siguiente dirección [www.itforcegt.org/9761622](http://www.itforcegt.org/9761622).

### **PAGINA PRINCIPAL**

Todos los estudiantes en los cursos mencionados arriba dentro de su directorio deben de crear una página llamada **INDEX.HTML**. **El contenido de esta y otras páginas serán 100% en ingles.**

## **EVALUACION**

Este proyecto tendrá un valor neto de 1 (un) punto en los cursos que lo requieren y **este punto será obligatorio para tener derecho a nota de laboratorio y de examen final** en el curso, la calificación será binaria, es decir, o cumple o no cumple (1 o 0). La forma en que el catedrático evaluará este proyecto, consistirá en acceder el sitio y verificar cada sitio de cada estudiante en su **antes del 10 de Octubre de 2006**. El catedrático será el responsable del contenido y de la presentación de los sitios de los estudiantes asignados al listado que la escuela le entregue. Si el catedrático evalúa que el contenido del sitio NO es apropiado, dará por NO cumplido el proyecto (0 puntos) y deberá reportarlo a la escuela para que el sitio sea bloqueado.



### ***ANEXO 3: MATERIAL SEGUNDO SEMESTRE 2006***

#### **PRIMERA EVALUACIÓN PARCIAL SEGUNDO SEMESTRE 2006**

---

Is a Java programming language class that is used to extend the capabilities of servers that host applications access via a request-response programming model?

- Java server page
- Java servlet**
- Java server face
- Java standard tag library
- none answer is correct

The life cycle of a servlet is controlled by:

- the container in which the servlet has been deployed**
- the web server
- the same servlet
- b and c are correct
- none answer is correct

Name of service methods in a Java Servlet:

- init()
- doGet()
- destroy()
- doPost()

**b and d are correct**

It's a kind of servlet that can transform the header and content (or both) [ ] of a request or response.

Http servlet

Servlet listener

**Servlet filter**

b and c are correct

none answer is correct

To transfer control to another web component:

HttpSession

HttpServletRequest

**RequestDispatcher**

HttpServletResponse

all answers are correct

Is a text document that contains two types of text: static data, which can be expressed in any text-based format (such as HTML, SVG, WML, and XML), which construct dynamic content?

JavaBean

Custom Tag Library

**JSP page**

none answer is correct

all answers are correct

The pageContext for the JSP page, provides access to various objects, these are:

servletContext and session

request and response

header and cookie

**a and b are correct**

none answer is correct

Valid expression language examples:

"\${sessionScope.cart.total}"

\${customer.name}

#{customer.name}

none answer is correct

**all answers are correct**

Syntax for expressions in a Java server page

<%= ..%>

<%! ..%>

<% ..%>

<jsp:expression> .. </jsp:expression>

**a and d are correct**

Allows page authors to use simple expressions to dynamically read data from JavaBeans components:

HttpRequest

**Expression language**

- `HttpResponse`
- `HttpSession`
- Java server page

Encapsulates core functionality common to many JSP applications. This allows you to employ a single, standard set of tags.

**Java standard tag library**

- Custom tag library
- `JavaBeans`
- `SessionBeans`
- Core tag library

This is part of Java standard tag library (JSTL).

- Core tag library
- XML tag library
- Internationalization tag library
- SQL tag library

**all answers are correct**

Valid JSTL example.

- `<c:set var="bookId" value="{param.Remove}"/>`
- `<c:remove var="cart" scope="session"/>`
- `<h3><fmt:message key="Choose"/>`

**all answers are correct**

- none answer is correct

Defines tags for setting the locale for a page, creating locale-sensitive messages, and formatting and parsing data elements such as numbers, currencies, dates, and times in a locale-sensitive or customized manner.

Core tag library

**Internationalization tag library**

XML tag library

SQL tag library

none answer is correct

They are tags for accessing databases; they are designed for quick prototyping and simple applications. For production applications, database operations are normally encapsulated in JavaBeans components.

XML tag library

Core tag library

**SQL tag library**

Internationalization tag library

Custom tag library

Is a server-side user interface component framework for Java technology-based web applications?

Java server page

**Java server face**

Java standard tag library

all answers are correct

none answer is correct

This technology provides a rich architecture for managing component state, processing component data, validating user input, and handling events.

**JSF**

JSP

JSTL

Servlet

none answer is correct

What is a Java server faces application?

A set of JSP pages

A set of *backing beans*

An application configuration resource file, which defines page navigation rules and configures beans and other custom objects.

**all answers are correct**

none answer is correct

This is a technology for building web services and clients that communicate using XML. This technology allows developers to write message-oriented as well as RPC-oriented web services.

JAXR

**JAX-WS**

StAX

JAXB

AJAX

This architecture provides a fast and convenient way to bind between XML schemas and Java representations, making it easy for Java developers to incorporate XML data and processing functions in Java applications.

JAX-WS

AJAX

StAX

**JAXB**

all answers are correct

This is a streaming Java-based, event-driven, pull-parsing API for reading and writing XML documents. This enables you to create bidirectional XML parsers that are fast, relatively easy to program, and have a light memory footprint.

Java architecture for XML binding

**Streaming API for XML**

Java API for XML registries

all answers are correct

none answer is correct

What I need to include a tag library in my Java server page?

Tag library descriptor

URL

**URI**

all answers are correct

none answer is correct

Servlets, Filters, Java server pages and other web resources are mapping here:

- faces-config.xml
- Tag library descriptor
- web.xml**
- struts-config.xml
- deployment descriptor

What do you use for save the state of a user in a Java Web Application?

- Attributes in JavaBeans
- Http Sessions**
- Custom tag library
- Attributes in a Servlet
- Attributes in Java Server Page

What immediately happens to the Java servlet code that was translated from the JSP page?

- The translated code is sent to the browser
- It is compiled as an applet and then delivered to the client to run in its browser
- It is compiled into a class file that is stored in the Web container**
- It is compiled into a class file that is sent to the browser
- none answer is correct

## SEGUNDA EVALUACIÓN PARCIAL SEGUNDO SEMESTRE 2006

---

Java EE security services are provided by \_\_\_\_\_ and can be implemented using declarative or programmatic techniques.

- The application security
- The Annotations
- The component container**
- The deployment descriptors
- Application server

For a Web application, \_\_\_\_\_ is a complete database of users and groups that identify valid users of a Web application (or a set of Web applications) and are controlled by the same authentication policy

- A domain
- A profile
- A database
- A role
- A realm**

An abstract name for the permission to access a particular set of resources in an application. It can be compared to a key that can open a lock. Many people might have a copy of the key. The lock doesn't care who you are, only that you have the right key.

- A Password
- An Authorization
- An Authentication
- A profile

**A role**

The characteristics of a local client are?

**It must run in the same JVM as the enterprise bean it accesses**

- It cannot be a web component of another enterprise bean
- The location of the enterprise bean it accesses is transparent
- A, B y C.
- None correct

Which of the following options is a correct Requirement for Entity Classes?

The class must have a private or protected, no-argument constructor. The class can't have other constructors.

**The class must not be declared final . No methods or persistent instance variables must be declared final .**

- Entities can't extend both entity and non-entity classes, and non-entity classes may extend entity classes.
- All of the above
- None of the above

Represents a single client inside the Application Server:

Enterprise Bean

Entity Bean

**Session Bean**

Interactive Session

Web Service Client

Which is the component that condenses the logic of the business of an application.

Session Bean

**Enterprise Bean**

State Management Modes

Stateful Session Beans

none answer is correct

Creates and removes persistent entity instances, finds entities by the entity's primary key, and allows queries to be run on entities.

**EntityManager API**

Persistence Context

Container-Managed

Entity Instances

None answer is correct

Java Persistence consists in which areas:

The Java Persistence API

The Query Language

Object/relational mapping metadata

A and C are correct

**all answer are correct**

The state of the persistent is given by:

The object/relational

**Persistent fields or persistent properties**

- Public or protected constructor
- Session bean
- None of the above

The persistence runtime accesses entity class instance variables directly, this happens when we use:

- Embeddable classes
- Persistent properties
- Persistent fields**
- B and C are correct
- None answer is correct

\_\_\_\_\_ has the following traits: It can run on a different machine and a different Java virtual machine (JVM) than the enterprise bean it accesses. It can be a web component, an application client, or another enterprise bean. To it, the location of the enterprise bean is transparent.

- Remote Clients**
- Stateless Session
- Session Bean
- Enterprise Bean
- Local client

Is a lightweight persistence domain object. Represents a table in a relational database, and instance corresponds to a row in that table.

- Enterprise Bean
- JSTL

**Entity**

JAX-WS

None is correct

The security information is contained within the soap message and for soap message attachment.

**message layer security**

transport layer security

application layer security

security implementation message

nic

The primary programming artifact of an entity is:

**entity class**

entity manager

entity units

a and b correct

all correct

A primary key class must meet some requirements:

the class must be serializable

the access control modifier of the class must be public

java primitive types

**a and b are correct**

none correct

The state of the persistent is given by:

- the object/relational
- persistent fields or persistent properties**
- public or protected constructor
- session bean
- none answers is correct

The persistence runtime accesses entity class instance variables directly, this happen when we use:

- embeddable classes
- persistent properties
- persistent fields**
- b and c are correct
- none answer is correct

Is a set of managed entity instaces that exist in a particucar data store:

- The entity manager
- Container-manged entity managers
- The Persistence Context**
- Managing entities
- The pesistence

Creates and removes persistence entity instances:

- The persistence context
- The entity manager**

- Container-managed entity managers
- Application-managed entity managers
- Finding entities using the entity manager

The createNamedQuery method is used to create

- dynamic queries
- EntityManager
- static queries**
- Creating Queries
- None answer is correct

The means by which communicating entities (for example, client and server) prove to one another that they are acting on behalf of specific identities that are authorized for access. This ensures that users are who they say they are.

- Access
- Authentication**
- Authorization
- Auditing
- Affirmation

Is an enterprise bean that allows Java EE applications to process messages asynchronously.

- Entity Beans
- Session Beans
- message-driven bean**
- Stateless Session Beans
- none of the above

The primary programming artifact of an entity is?

Entities

**The entity class**

Managing Entities

A y B it is correct

None of the above

It is a Java EE Security Implementation mechanism that characterizes because is not an end-to-end solution, simply point-to-point.

Application-Layer Security

**Transport-Layer Security**

Message-layer Security

Declarative Security

Programmatic Security

This type of EJB perform a task for a client, optionally may implement a web service

Stateful Session

**Session**

Message-Driven

NAC

None of the above.

## EVALUACIÓN FINAL SEGUNDO SEMESTRE 2006

---

Is an object that represents a group of objects (such as the familiar Vector class)?

**a collection**

an ArrayList

a Map

all answers are correct

none answer is correct

Is a unified Java architecture for representing and manipulating collections, allowing them to be manipulated independently of the details of their representation?

a collection

data structures

**a collections framework**

all answers are correct

none answer is correct

What are two possible HTTP request types, and what are the names of the servlet methods that handle them?

**GET and POST, the doGet and doPost methods**

GET and POST, the service and init methods

SEND and RECEIVE, the service and init methods

REQUEST and RESPONSE, the getRequest and setResponse methods

all answers are correct

Can a single session object be accessed by different servlets in the same Web application?

No

**Yes**

What is the best way to prevent multiple servlet instances from simultaneously using the same resource?

**Implement the SingleThreadModel interface.**

Allow the Web container to prevent multiple servlet instances from simultaneously using the same resource automatically.

Synchronize those method calls that use that resource.

Construct a separate monitor object for each shared resource to coordinate the flow of threads into the resource.

none answer is correct

Is a technology that is based on a set of standards for building interoperable distributed applications, performs self-describing business functions, is service-oriented and is component-based?

web application

web site

**web service**

all answers are correct

none answer is correct

Is a description language to define Web service interfaces and how to invoke them?

**wSDL**

uddi

ajax

all answers are correct

none answer is correct

Is an online electronic registry for registering businesses and Web services and is a specification for description and discovery?

wSDL

**uddi**

ajax

all answers are correct

none answer is correct

Is an enterprise bean that allows Java EE applications to process messages asynchronously and it normally acts as a JMS message listener, which is similar to an event listener except that it receives JMS messages instead of events?

session bean

entity bean

**message driven bean**

all answers are correct

none answer is correct

It represents a single client inside the Application Server.

**session bean**

- entity bean
- message driven bean
- all answers are correct
- none answer is correct

Which language provides the rules for how JSP tags are used on JSP pages?

- XML
- The Java programming language
- SQL
- HTML
- none answer is correct**

A \_\_\_\_\_ defines a set of all entity classes that are managed by Entity-Manager instances in an application. This set of entity classes represents the data contained within a single data store.

- Entity Instance
- Entities
- Enterprise Beans
- Persistence Units**
- None answer is correct

Java Persistence consists of:

- The Java Persistence API
- The query language
- Object/relational mapping metadata
- A, B and C**

none answer is correct

Typically an \_\_\_\_\_ represents a table in a relational database, and each instance corresponds to a row in that table.

Web Container

Servlet

Enterprise Bean

**Entity**

none answer is correct

A \_\_\_\_\_ is a messaging system that implements the JMS interfaces and provides administrative and control features.

**JMS provider**

JMS clients

Administered objects

Messaging Domains

none answer is correct

The means by which communicating entities (for example, client and server) prove to one another that they are acting on behalf of specific identities that are authorized for access. This ensures that users are who they say they are.

**Authentication**

Authorization, or access control

Data integrity

Confidentiality or Data Privacy

Non-repudiation

Auditing

In the MVC design pattern, which part of the design verifies request data, maps client data to the model, and selects which view to show?

the model

the view

**the controller**

all answers are correct

none answer is correct

In the MVC design pattern, which part of the design renders the HTML response, requests updates from models, and provides HTML forms to the user?

the model

**the view**

the controller

all answers are correct

none answer is correct

What is Ant?

---

---

What is Hibernate?

---

---

## ENUNCIADO PRIMER PROYECTO SEGUNDO SEMESTRE 2006

---

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA – ESCUELA INGENIERÍA EN CIENCIAS Y  
SISTEMAS  
SEGUNDO SEMESTRE DE 2006  
PROYECTO 1

### PRIMER PROYECTO

Realizar un portal Web que cumpla con las siguientes características.

La página inicial del portal debe permitir que el usuario ingrese su *username* y *password* para tener acceso a las aplicaciones del portal. La autenticación se debe verificar en un Servlet, si el acceso es autorizado se debe de mostrar una página que contenga el menú de las aplicaciones del portal, en caso contrario una página en la cual se deniegue el acceso a las aplicaciones.

Deben maquillar la página inicial del portal con GIFs animados y animaciones de flash para hacer promociones.

El menú de las aplicaciones del portal Web debe contener las siguientes opciones.

#### **1. Carga de productos.**

Esta opción debe permitir cargar un archivo de texto con formato XML conteniendo las descripciones de los productos. El archivo de XML se debe de subir al Web Site (upload) mediante una JSP. Un Servlet debe de recibir el contenido del archivo XML y deberá ingresar los productos a la base de datos.

## 2. Listado de productos.

Ejemplo:

<b>PRODUCTO</b>	<b>NOMBRE</b>	<b>EXISTENCIA</b>	<b>DETALLE</b>
<b>1</b>	<b>CDs en blanco</b>	<b>500</b>	<b>Link (pagina de detalle)</b>
<b>2</b>	<b>DVDs en blanco</b>	<b>1500</b>	<b>Link (pagina de detalle)</b>
<b>...</b>			

## 3. Lista de precios. (Formato de moneda y fecha)

Ejemplo:

<b>PRODUCTO</b>	<b>NOMBRE</b>	<b>PRECIO</b>	<b>DETALLE</b>
<b>1</b>	<b>CDs en blanco</b>	<b>Q 1.62</b>	<b>Link (pagina de detalle)</b>
<b>2</b>	<b>DVDs en blanco</b>	<b>Q 3.50</b>	<b>Link (pagina de detalle)</b>
<b>...</b>			

## 4. Eliminar inventario

Esta opción permitirá eliminar el contenido del inventario. (debe eliminar todo el contenido de la base de datos)

En la página de detalle se debe mostrar toda la información del producto. (código, nombre, costo, fecha de ingreso, utilidad, precio). Todos los valores de moneda y fecha deben ser formateados utilizando JSTL.

La lista de los productos para las consultas se debe generar haciendo la consulta a la base de datos. Mediante un JavaBean se deberá realizar la consulta a la base de datos utilizando JDBC (no utilizar ODBC), el JavaBean debe de tener métodos que devuelvan

colecciones de objetos (productos de la base de datos). Estos JavaBeans deben ser utilizados en las JSPs y mediante JSTL se deberán de recorrer las colecciones para generar dinámicamente una tabla (código de html) con el listado de los productos.

Se debe de seleccionar en que moneda se quieren visualizar los precios. (Dólares, quetzales, euros, pesos). El formato de los precios debe darse con el símbolo para cada una de estas monedas.

El tipo de cambio para cada una de estas monedas lo dará un Web Service que recibirá como parámetro el tipo de moneda. Los valores que devolverá el Web Service los deberá leer de un archivo XML en el cual contendrá los tipos de cambio.

Ejemplo:

```
<monedas>
  <moneda>
    <nombre>dolar</nombre>
    <cambio>7.62</cambio>
  </moneda>
  <moneda>
    <nombre>euro</nombre>
    <cambio>8.5</cambio>
  </moneda>
  <moneda>
    <nombre>peso</nombre>
    <cambio>0.89</cambio>
  </moneda>
  <moneda>
    <nombre>quetzal</nombre>
    <cambio>1.00</cambio>
  </moneda>
</monedas>
```

Seguridad del portal:

La aplicación debe ser monitoreada mediante un Servlet Filter, ningún usuario que no este autenticado podrá acceder a ningún recurso Web del portal excepto la página de inicio y los recursos que esta utilice. El Servlet Filter debe de guardar una bitácora de accesos a los recursos Web en un archivo de texto con la siguiente información: fecha y hora de acceso, recurso accedido, usuario que realizo el acceso

Cuando un usuario intenta acceder a un recurso del portal que no es público el Servlet Filter debe denegar el acceso y redireccionar hacia una página de acceso denegado.

Para saber si un usuario se autenticó al ingresar al portal, el Servlet Filter deberá verificar que exista un usuario cargado a la Session (HttpSession), si este no existe en la session se tomará como no autenticado.

## Apéndice

Estructura de la tabla de productos

<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
<b>PRODUCTO</b>	<b>Llave primaria. Código del producto.</b>
<b>NOMBRE</b>	<b>Nombre del producto.</b>
<b>COSTO</b>	<b>Costo del producto en quetzales.</b>
<b>INGRESO</b>	<b>Fecha de ingreso del producto al inventario.</b>
<b>EXISTENCIA</b>	<b>Cantidad de productos existentes en el inventario.</b>
<b>UTILIDAD</b>	<b>Porcentaje de utilidad que se sumara al costo del producto para poder generar el precio del mismo.</b>

**Base de Datos:**

Todos los accesos a la base de datos se deben realizar mediante JDBC.

**TIEMPO PARA DESARROLLAR EL PROYECTO:**

3 semanas después de la fecha de entrega.



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA – ESCUELA INGENIERÍA EN CIENCIAS Y  
SISTEMAS  
SEGUNDO SEMESTRE DE 2006  
PROYECTO 2

### **SEGUNDO PROYECTO**

Este consistirá en agregar funcionalidad al proyecto anterior, los cambios o agregados del primer proyecto son:

- Internacionalización
- Persistencia
- Carretilla de compras
- Registro de usuarios
- Buzón de mensajes

#### **Internacionalización:**

El proyecto debe de tener soporte multi-lenguaje, para ello se deben de utilizar las Tag Libraries de internacionalización. Los idiomas que deben de contemplarse para el proyecto son: inglés y español. El usuario del portal no debe en ningún momento seleccionar el idioma con el cual quiera consultar el portal, sino que este automáticamente se lo deberá de mostrar en el idioma que tenga predeterminado el sistema operativo en donde se encuentra consultando el portal. Si no existiera el soporte para el idioma del SO el default será el idioma ingles.

#### **Persistencia:**

Toda la información que es manipulada en la base de datos (inserciones, consultas, actualizaciones, eliminaciones) deberán de realizarse utilizando Entity beans y deberá de ser manejada por el Contenedor.

**Carretilla de compras:**

Se debe de implementar un sistema de compras que permita agregar a una carretilla los productos que anteriormente fueron cargados a la base de datos. Al finalizar el proceso de compra se debe enviar un vaucher al usuario (mensaje a su buzón).

**Registro de usuarios:**

Se debe de realizar un registro de usuarios para que puedan loguearse y consultar el buzón de mensajes el cual deberá ser implementado usando Message Driven Beans.

**Buzón de mensajes:**

Interfaz web que permita a los usuarios registrados leer sus mensajes. Únicamente se realizará la operación de lectura de mensajes, mas no de envío.

Universidad De San Carlos De Guatemala  
Facultad De Ingeniería  
Estructura De Datos



## Hoja de Calificación

### Segundo Proyecto

Nombre	Carnet	Firma

Detalle	Ponderación	
Manual Usuario	/	5
Manual Técnico	/	5
Internacionalización	/	15
Persistencia	/	20
Carretilla de compras	/	15
Registro de usuarios	/	15
Buzón de mensajes	/	15
Reporte Bitácora	/	10
<b>Total</b>		100



## NOTAS FINALES SEGUNDO SEMESTRE 2006

#	Carnet	Nombre	15 1er Parcial	15 2do Parcial	10 Tareas	40 Laboratorio	20 Examen	Final	100 Nota Final
1	0093-13118	Cesar Calderón	0	4.2	2	0	0	0	<b>6.2</b>
2	0096-15573	Erick Gerardo Morales Rivas	12.75	7.8	6	1.2	13	13	<b>40.75</b>
3	1999-19582	Gustavo Adolfo Siguantay	6.6	6.6	10	24.4	17	17	<b>64.6</b>
4	2000-10633	Jorge Giovanni Obando Ortiz	7.8	10.2	8	26.8	10	10	<b>62.8</b>
5	2000-11024	Nancy Castillo	7.2	9.6	8	24.4	17	17	<b>66.2</b>
6	2000-11522	Carlos O. Natareno Yanes	8.4	9	10	24.4	13	13	<b>64.8</b>
7	2000-18212	Leonardo Donis	6	6.6	10	27.6	17.5	17.5	<b>67.7</b>
8	2001-10203	Mynor Peralta Herrera	6.6	6	6	5.6	13	13	<b>37.2</b>
9	2001-12874	Nelson Hernandez Foronda	6.6	7.8	2	0	10	10	<b>26.4</b>
10	2001-13042	Mynor Adalberto Agustin Martínez	6	4.2	6	2	8.5	8.5	<b>26.7</b>
11	2001-13611	Manuel Matul Palma	7.8	9	4	0.8	13	13	<b>34.6</b>
12	2001-17357	Sonia L. Sagastume Miranda	4.2	4.8	10	0	9	9	<b>28</b>
13	2001-17375	Hernández Siguantay Juan José	0	0	0	1.2	0	0	<b>1.2</b>
14	2001-17542	Edwin Enrique Burrion X.	7.8	0	2	0.8	0	0	<b>10.6</b>
15	2001-30016	Victor Josue Solís Urizar	0	6	8	0	0	0	<b>14</b>
16	2002-12907	Julio César Morales Jerónimo	9	10.2	10	28.8	16	16	<b>74</b>
17	2002-13196	Byron Rodolfo Zepeda Arévalo	8.4	7.8	10	30.8	15.5	15.5	<b>72.5</b>
18	2003-12383	Héctor Samuel Alvarado Diaz	9	10.2	10	26	11.5	11.5	<b>66.7</b>
19	2003-12385	Leonel Estuardo Mejía Avalos	6.6	10.2	4	30	12	12	<b>62.8</b>
20	2003-12422	B. Alexander Bolaños Lima	7.8	0	2	7.6	0	0	<b>17.4</b>

21	2003-12442	-	0	0	0	1.2	0	<b>1.2</b>
22	2003-13015	Edwar Fernando Barrios Estrada	7.8	4.8	10	25.6	14	<b>62.2</b>
23	2003-13053	Roberto Rubén Coloma Pineda	7.8	9.6	10	29.6	14	<b>71</b>
24	2003-13153	Pedro Baldemar Ramos Juarez	8.4	0	2	0	0	<b>10.4</b>
25	2003-13198	Fernando Abraham Avendaño Madrid	9.6	5.4	6	4.8	7	<b>32.8</b>
26	2003-13212	Ana Patricia Rodríguez Fernández	8.4	10.8	8	27.2	11	<b>65.4</b>
27	2003-13239	William Estuardo Escobar Argueta	9.6	9	10	26.4	12	<b>67</b>
28	2003-13390	Marylin Beatriz Paxtor García	9.6	0	2	0	0	<b>11.6</b>
29	2003-13470	Elder Anibal Chamalá	8.4	0	2	0	0	<b>10.4</b>
30	2003-20452	Melvin Rolando Marcos Mijangos	6	9	8	28	13	<b>64</b>
31	2003-30475	Pedro Luis Domingo Vasquez	8.4	3.6	10	34.8	11	<b>67.8</b>
32	2003-31154	Edwin Guillermo Ramírez Morales	9	4.8	10	37.6	13	<b>74.4</b>
33	2004-12360	Eliezer Abdiel Palacios Marroquín	6.6	0	6	6.8	0	<b>19.4</b>
34	2004-12389	David Estuardo Morales	9.6	11.4	10	31.2	11	<b>73.2</b>
35	2004-12458	Edgar Rubén Sabán Raxón	8.4	8.4	10	32.4	13.5	<b>72.7</b>
36	2004-12577	Elsa Cristina Calderon Flores	10.35	10.8	10	30.4	17	<b>78.55</b>
37	2004-12709	Byron José Ramírez Monterroso	9	9.6	8	30.4	18	<b>75</b>
38	2004-12831	Gustavo Arnoldo Ayapan Vargas	7.8	6.6	10	32.8	11	<b>68.2</b>
39	2004-12905	Ervin Gerardo Ramírez Tacejo	7.2	6.6	10	25.2	15	<b>64</b>
40	2004-12978	Audie Rene Juárez	6.6	12.75	8	28.8	17	<b>73.15</b>
41	2004-12987	Julio Roberto Chicas Polanco	9	9.6	8	27.6	14	<b>68.2</b>
42	2004-13003	Max Alejandro Antonio Cerna Flores	7.8	10.2	10	32	14.5	<b>74.5</b>
43	2004-13023	Yuri Asucena Castro	8.4	7.2	10	25.2	14	<b>64.8</b>
44	2004-13044	Henry Giovanni Barrientos García	9.6	4.2	6	8	7	<b>34.8</b>

45	2004-13054	Jorge Andrés Yass Coy	8.4	7.8	10	10.4	16.5	<b>53.1</b>
46	2004-13148	César Salvador Rojas	8.4	9.6	10	24.4	14	<b>66.4</b>



## ANEXO 4: MATERIAL PRIMER SEMESTRE 2007

---

### PRIMERA EVALUACIÓN PARCIAL PRIMER SEMESTRE 2007

---

1. \_\_\_\_\_ is a Java programming language class that is used to extend the capabilities of servers that host applications access via a request-response programming model.
  - A.  Class
  - B.  Method
  - C.  Implementation
  - D.  Custom Tag
  - E.  **Servlet**
  
2. \_\_\_\_\_ is a text document that contains two types of text: static data, which can be expressed in any text-based format (such as HTML, SVG, WML, and XML), and JSP elements, which construct dynamic content?
  - A.  Schema
  - B.  **XML Document**
  - C.  JSP page
  - D.  Java Server Faces Application
  - E.  Custom Tag
  
3. Syntax for comments in java server page.
  - A.  `<%= ...%>`
  - B.  **<%! ...%>**

- C.  `<% ...%>`
- D.  None answer is correct
- E.  All answer are correct

4. \_\_\_\_\_ is an object that represents a group of objects.

- A.  A map
- B.  A collection**
- C.  A array list
- D.  A class definition
- E.  A class group

5. What are two possible HTTP request types, and what are the names of the servlet methods that handle them?

- A.  Get and post, the service and int methods
- B.  Get and post, the doGet and doPost methods**
- C.  Get and post, the init and doPost methods
- D.  Get and post, the service and doPost methods
- E.  Send and Receive, the service and int methods

6. Allows page authors to use simple expression to dynamically read data from JavaBeans components:

- A.  Java Server Page**
- B.  HttpSession
- C.  HttpResponse
- D.  HttpRequest
- E.  HttpPage

7. The familiar set abstraction. No duplicate elements permitted. May or may not be ordered. Extends the Collection interface.

A.  Collection

**B.  Set**

C.  List

D.  Map

E.  SortedMap

8. It's a kind of servlet that can transform the header and content (or both) of a request or response

**A.  Http servlet**

B.  Servlet listener

C.  Servlet filter

D.  None answer is correct

E.  All answer are correct

9. \_\_\_\_\_ involves creating in-memory objects representing an entire document tree and the complete infoset state for an XML document.

**A.  Document Object Model**

B.  Streaming Model

C.  Rendering Model

D.  Conversion Model

E.  Parsing Model

10. \_\_\_\_\_ provides a fast and convenient way to bind between XML schemas and Java representations, making it easy for Java developers to incorporate XML data and processing functions in Java applications.

- A.  Java API for XML Web Services (JAX-WS)
- B.  Java Architecture for XML Binding (JAXB)**
- C.  Streaming API for XML (StAX)
- D.  SOAP with Attachments API for Java (SAAJ)
- E.  Java API for XML Registries (JAXR)

11. What is a JSF application?

- A.  An application configuration resource file, which defines page navigation rules and configures beans and other custom objects**
- B.  A set of JSP pages
- C.  A set of backing beans
- D.  None answer is correct
- E.  All answer are correct

12. The \_\_\_\_\_ platform uses a distributed multitiered application model for enterprise applications.

- A.  Java Beans
- B.  Java EE**
- C.  Java EE component
- D.  Application Client
- E.  Web Client

13. A \_\_\_\_\_ is a self-contained functional software unit that is assembled into a Java EE application with its related classes and files and that communicates with other components

- A.  Java Beans
- B.  Java EE
- C.  **Java EE component**
- D.  Application Client
- E.  Web Client

14. The server and client tiers might also include components based on the \_\_\_\_\_ component architecture to manage the data flow between an application client or applet and components running on the Java EE server, or between server components and a database.

- A.  **Java Beans**
- B.  Java EE
- C.  Java EE component
- D.  Application Client
- E.  Web Client

15. Ordered collection, also known as a *sequence*. Duplicates are generally permitted. Allows positional access. Extends the Collection interface.

- A.  Collection
- B.  Set
- C.  **List**
- D.  Map
- E.  SortedSet



**Instrucciones:** Responda 15 preguntas. Si responde más de 15, se descartará una respuesta correcta. Marcar solo una opción en medio de los corchetes. Si marca dos opciones, se anulará la pregunta Use lapicero o marcador, no use lápiz.. No use corrector.

1. In the Streaming API for XML, Binds a source schema to a set of schema derived program elements. The binding is described by an XML-based binding language.
  - A.  **Schema compiler**
  - B.  Schema generator
  - C.  Binding runtime framework:
  - D.  Container.
  - E.  None of the above.
  
2. This type of EJB perform a task for a client, optionally may implement a web service
  - A.  Stateful Session
  - B.  **Session**
  - C.  Message-Driven
  - D.  NAC
  - E.  None of the above
  
3. Technology for building web services and clients that communicate using XML, and allows developers to write message-oriented as well as RPC-oriented web services.

- A.  **JAX-WS**
- B.  JAXB
- C.  SOAP
- D.  STAX
- E.  None of the above.

4. Provides a client application the ability to convert a JAXB-derived Java object tree back into XML data.

- A.  **Marshalling.**
- B.  Unmarshalling.
- C.  Compilation.
- D.  Push Parsing.
- E.  Pull Parsing.

5. This model involves creating in-memory objects representing an entire document tree and the complete infoset state for an XML document. Once in memory, the trees can be navigated freely and parsed arbitrarily, and as such provide maximum flexibility for developers.

- A.  Marshalling
- B.  Streaming
- C.  **Document object model**
- D.  Push Parsing.
- E.  Pull Parsing.

6. Refers to a programming model in which XML infosets are transmitted and parsed serially at application runtime, often in real time, and often from dynamic sources

whose contents are not precisely known beforehand. Moreover, its parsers can start generating output immediately, and infoset elements can be discarded and garbage collected immediately after they are used.

- A.  Marshalling
- B.  Streaming**
- C.  Document object model
- D.  Push Parsing.
- E.  Pull Parsing.

7. Refers to a programming model in which a client application calls methods on an XML parsing library when it needs to interact with an XML infoset—that is, the client only gets XML data when it explicitly asks for it.

- A.  Marshalling
- B.  Streaming
- C.  Document object model
- D.  Streaming Push Parsing.
- E.  Streaming Pull Parsing.**

8. Refers to a programming model in which an XML parser sends XML data to the client as the parser encounters elements in an XML infoset—that is, the parser sends the data whether or not the client is ready to use it at that time.

- A.  Marshalling
- B.  Streaming
- C.  Document object model
- D.  Streaming Push Parsing.**
- E.  Streaming Pull Parsing.

9. The characteristics of a local client are?

- A.  **It must run in the same JVM as the enterprise bean it accesses**
- B.  It cannot be a web component of another enterprise bean
- C.  The location of the enterprise bean it accesses is transparent
- D.  A, B y C.
- E.  None correct

10. Which is the component that condenses the logic of the business of an application.

- A.  Session Bean
- B.  **Enterprise Bean**
- C.  State Management Modes
- D.  Stateful Session Beans
- E.  none answer is correct

11. Represents a single client inside the Application Server:

- A.  Enterprise Bean
- B.  Entity Bean
- C.  **Session Bean**
- D.  Interactive Session
- E.  Web Service Client

12. In JAX-WS, a web service operation invocation is represented by an XML-based protocol such as \_\_\_\_\_, its specification defines the envelope structure, encoding rules, and conventions for representing web service invocations and responses.

- A.  Http
- B.  Html
- C.  JXB
- D.  SOAP**
- E.  None of the above

13. In JAX-WS, a web service operation invocation is represented by an XML-based protocol such as \_\_\_\_\_, its specification defines the envelope structure, encoding rules, and conventions for representing web service invocations and responses.

- A.  Deployment models
- B.  Security Models
- C.  Streaming models**
- D.  Document object models
- E.  None of the Above

14. Is an infrastructure that enables the building, deployment, and discovery of web services. It is a neutral third party that facilitates dynamic and loosely coupled business-to-business (B2B) interactions. It is available to organizations as a shared resource, often in the form of a web-based service.

- A.  Entity.
- B.  Entiy Instance.
- C.  Enterprise Bean.
- D.  XML registry**
- E.  Library.

15. Is an enterprise bean that allows Java EE applications to process messages asynchronously.

- A.  Entity Beans
- B.  Session Beans
- C.  **message-driven bean**
- D.  Stateless Session Beans
- E.  none of the above

16. With JAX-WS clients and webservices have a big advantage: the platform\_\_\_\_\_of the java programming language.. In addition JAX-WS is not restrictive. A JAX-WS client can access a webservice that is not running on the java platform and vice versa.

- A.  Marshalling
- B.  Totally Dependent
- C.  **Independent**
- D.  Relative Independent
- E.  all are corrects

17. \_\_\_\_\_-provides unMarshalling and Marshalling operations for accessing, manipulating and vlidating XML content.

- A.  Schema generator
- B.  **Binding runtime FrameWork**
- C.  Schema Compiler
- D.  all are corrects

18. Currently there are a variety of XML specifications Registries. These included:

- A.  ebXML and ODI

B.  **UDDI and ebXML**

C.  XML Registry

D.  JAXR

19. Streaming API for XML is based on:

A.  event-driven Java Technology

B.  pull-parsing Java Technology

C.  A is correct nor B

D.  **Both A and B are corrects**

20. There are 3 types of beans:

A.  **Entity, Message Driven, Session**

B.  Session, Managed, Message Driven

C.  Enterprise, Businesslike, Commercial.

D.  Stateless, Clueless, Message Driven.

E.  None are correct



## TERCERA EVALUACIÓN PARCIAL PRIMER SEMESTRE 2007

---

1. Java provides an \_\_\_\_\_ mapping facility for java developers for managing relational data in java applications.
  - A.  JNDI
  - B.  JAX-WS
  - C.  JDBC and JNDI
  - D.  **none all above.**
  
2. Typically \_\_\_\_\_ represents a table in \_\_\_\_\_ database , and each entity instance corresponds to a row in that table.
  - A.  Session bean , EJB Container
  - B.  **an object , relational**
  - C.  Entity, hierarchic
  
3. Type of Multiplicity Relationship where one Entity Instance can be related with multiple instances of other entities.
  - A.  Many to One
  - B.  One to One
  - C.  Many to Many
  - D.  **none all above.**
  
4. Is a type of Security Service applied into JEE applications where is embedded in an application and is used to make security decisions. Its useful when declarative security alone is not sufficient to express the security model of an application.

- A.  Declarative Security
  - B.  Metadata annotations
  - C.  **Programmatic Security**
  - D.  a and c are correct.
5. Java EE Security Services provide a robust and easily configured mechanism for \_\_\_\_\_ and \_\_\_\_\_ to application functions.
- A.  authenticating users , validate users
  - B.  **authorizing access , authenticating users**
  - C.  login, logout
  - D.  none all above.
6. Two requirements for build an Entity class are:
- A.  A public or protected non-argumented constructor
  - B.  Persistence Instances variables must be declarative private, protected or package-private.
  - C.  No methods or persistence Instance variables must declared final.
  - D.  **a and c are corrects nor b.**
  - E.  none all above.
7. Two of the followings are areas of of Java Persistence.
- A.  **Object/Relational mapping Metadata**
  - B.  JNDI Persistence Methods
  - C.  **Java Persistence API**

- D.  none all above
8. In matter of Services can help a developer to make secure the following elements.
- A.  Enterprise Beans
- B.  **Web Services**
- C.  **Web Applications**
- D.  b and c are not corrects.
9. If you are Securing WebServices , components such as \_\_\_\_\_ are protected.
- A.  **transport layer, message layer and application layer.**
- B.  client layer and database layer.
- C.  client layer, database layer, component layer
- D.  none all above.
10. JCE is part of Java Security Services and provide a framework and implementations for encryption, key generation and key agreement and Message Authentication code (MAC) algorithms.
- A.  **true**
- B.  false
- C.  none all above.



## EVALUACIÓN FINAL PRIMER SEMESTRE 2007

---

1. The means by which communicating entities (for example, client and server) prove to one another that they are acting on behalf of specific identities that are authorized for access. This ensures that users are who they say they are.
  - A.  Access
  - B.  Authentication**
  - C.  Authorization
  - D.  Auditing
  - E.  Affirmation
  
2. It is a Java EE Security Implementation mechanism that characterizes because is not an end-to-end solution, simply point-to-point.
  - A.  Application-Layer Security
  - B.  Transport-Layer Security**
  - C.  Message-layer Security
  - D.  Declarative Security
  - E.  Programmatic Security
  
3. Java EE security services are provided by \_\_\_\_\_ and can be implemented using declarative or programmatic techniques.
  - A.  The application security
  - B.  The Annotations
  - C.  The component container**
  - D.  The deployment descriptors

- E.  Application server
4. For a Web application, \_\_\_\_\_ is a complete database of users and groups that identify valid users of a Web application (or a set of Web applications) and are controlled by the same authentication policy
- A.  A domain
- B.  A profile
- C.  A database
- D.  A role
- E.  **A realm**
5. An abstract name for the permission to access a particular set of resources in an application. It can be compared to a key that can open a lock. Many people might have a copy of the key. The lock doesn't care who you are, only that you have the right key.
- A.  A Password
- B.  An Authorization
- C.  An Authentication
- D.  A profile
- E.  **A role**
6. Is an enterprise bean that allows Java EE applications to process messages asynchronously.
- A.  Entity Beans
- B.  Session Beans
- C.  **message-driven bean**

- D.  Stateless Session Beans
  - E.  none of the above
7. The characteristics of a local client are?
- A.  **It must run in the same JVM as the enterprise bean it accesses**
  - B.  It cannot be a web component of another enterprise bean
  - C.  The location of the enterprise bean it accesses is transparent
  - D.  A, B y C.
  - E.  None correct
8. Represents a single client inside the Application Server:
- A.  Enterprise Bean
  - B.  Entity Bean
  - C.  **Session Bean**
  - D.  Interactive Session
  - E.  Web Service Client
9. Is an object that represents a group of objects (such as the familiar Vector class)?
- A.  **A collection**
  - B.  An ArrayList
  - C.  A Map
  - D.  All above answers are correct
  - E.  None answer is correct
10. Is a unified Java architecture for representing and manipulating collections, allowing them to be manipulated independently of the details of their representation?

- A.  A collection
- B.  Data structures
- C.  **A collections framework**
- D.  All answers are correct
- E.  None answer is correct

11. What are two possible HTTP request types, and what are the names of the servlet methods that handle them?

- A.  **GET and POST, the doGet and doPost methods**
- B.  GET and POST, the service and init methods
- C.  SEND and RECEIVE, the service and init methods
- D.  REQUEST and RESPONSE, the getRequest and setResponse methods
- E.  All answers are correct

12. Is an enterprise bean that allows Java EE applications to process messages asynchronously and it normally acts as a JMS message listener, which is similar to an event listener except that it receives JMS messages instead of events?

- A.  Session bean
- B.  Entity bean
- C.  **Message driven bean**
- D.  All answers are correct
- E.  None answer is correct

13. It represents a single client inside the Application Server.

- A.  **Session bean**

- B.  Entity bean
- C.  Message driven bean
- D.  All answers are correct
- E.  None answer is correct

14. Java Persistence consists of:

- A.  The Java Persistence API
- B.  The query language
- C.  Object/relational mapping metadata
- D.  **A, B and C**
- E.  None answer is correct

15. A \_\_\_\_\_ is a messaging system that implements the JMS interfaces and provides administrative and control features.

- A.  **JMS provider**
- B.  JMS clients
- C.  Administered objects
- D.  Messaging Domains
- E.  none answer is correct

16. The means by which communicating entities (for example, client and server) prove to one another that they are acting on behalf of specific identities that are authorized for access. This ensures that users are who they say they are.

- A.  **Authentication**
- B.  Authorization, or access control
- C.  Data integrity

- D.  Confidentiality or Data Privacy
- E.  Non-repudiation
- F.  Auditing

17. In the MVC design pattern, which part of the design verifies request data, maps client data to the model, and selects which view to show?

- A.  The model
- B.  The view
- C.  **The controller**
- D.  All answers are correct
- E.  None answer is correct

18. In the MVC design pattern, which part of the design renders the HTML response, requests updates from models, and provides HTML forms to the user?

- A.  The model
- B.  **The view**
- C.  The controller
- D.  All answers are correct
- E.  None answer is correct

19) What is Ant?

20) What is Hibernate?



## ENUNCIADO PRIMER PROYECTO PRIMER SEMESTRE 2007

---

Universidad de San Carlos

Facultad de Ingeniería

Escuela Ingeniería en Ciencias y Sistemas

Curso. Estructura de Datos

Catedráticos. Ing. Guillermo López , Ing. Carlos Alonso, Ing. Marlon Orellana

Primer Semestre 2007

### **ENUNCIADO DE PROYECTO UNICO DE LABORATORIO**

Una empresa de televisión a nivel nacional impulsa una nueva forma de presentar noticias a través de Internet, para efecto de dichos requerimientos se le ha contratado a usted para el desarrollo de dicho sistema en base a los requerimientos específicos que se le plantean de forma detallada.

#### **Requerimientos Generales**

El sistema coordinado para el registro y distribución de noticias a nivel nacional requiere de las siguientes operaciones:

1. Registro de Noticias (Altas, Bajas y Cambios)
2. Registro de Usuarios (Altas, Bajas y Cambios)
  - a. Usuarios Administración
  - b. Usuarios Suscriptores
3. Publicación de Noticias
4. Envío de Noticias a suscriptores
5. Sincronización de Noticias Departamentos-Central

#### **Requerimientos Específicos**

##### **Registro de Noticias:**

Para el registro de noticias se requiere que un usuario con privilegios de administración de noticias pueda ejecutar operaciones como: Altas, Bajas y Cambios, Búsquedas y la

visualización de los listados de noticias que puedan ser manipuladas a través del sistema.

Dicho registro involucra el uso de imágenes, video y sonido con el objeto de que la noticia sea descrita por archivos multimedia que denoten la veracidad de la misma.

### **Registro de Usuarios:**

Como en todo sistema la seguridad del mismo permitirá el control del registro de usuarios de la siguiente forma:

#### **1. Desde el WebSite**

Para el registro de este tipo de usuarios la persona puede acceder desde la pagina principal a registrarse para poder afiliarse a los servicios que presta la Empresa para el envío de Noticias a través de correo electrónico. Esta opción permite el registro de usuarios suscriptores del servicio de noticias.

#### **2. Desde la Intranet de la empresa**

Internamente la aplicación funciona de igual forma para el registro de usuarios de alto nivel como usuarios administradores de noticias, publicadores, y redactores de noticias definiendo el funcionamiento de los roles de usuarios de la siguiente forma:

**Usuarios Administradores:** Usuarios con privilegios absolutos a la utilización de todas las opciones en el sistema.

**Usuarios Publicadores:** Usuarios que tienen como objeto la redacción y o Publicación de noticias que son escritas por los redactores o ellos mismos para que puedan visualizarse en el website.

**Usuarios Redactores:** Usuarios que tiene como función única la redacción de noticias es decir tienen a su cargo el manejo de noticias que le corresponden únicamente a el, pudiendo crear, modificar y eliminar noticias a excepción de las noticias publicadas que no podrán ser eliminadas a excepción de que el Usuario Publicador cambie el estado de

la (s) noticias a eliminarse.

**Usuarios Suscriptores:** Este tipo de usuarios sólo serán registrados como usuarios que harán uso del sistema para poder adquirir el servicio de noticias en línea, para el cual deberán registrar datos personales y de contacto vía electrónica para que las noticias lleguen a su cuenta de correo personal y de esta manera gozar de un servicio personalizado de noticias.

### **Publicación de Noticias:**

El módulo de publicación de Noticias ejecutará las siguientes operaciones:

- 1. Publicación de noticias creadas por los redactores o los mismos publicadores para poder ser visualizadas desde el portal de noticias diariamente.**
- 2. Activación de las Publicaciones del día creadas por los redactores.**
- 3. Desactivación de las Publicaciones no deseadas.**
- 4. Listar y Buscar de Publicaciones**

### **Envío de noticias a suscriptores:**

Este módulo dispondrá a los usuarios Administradores/Publicadores el manejo del envío de noticias a los usuarios registrados del sistema, comúnmente presentando un listado de las noticias del día listándolas para activar aquellas noticias de interés según el Administrador/Publicador para que estas sean enviadas a los usuarios Suscriptores directamente a su cuenta de correo personal.

### **Sincronización de Noticias Departamentos-Central:**

Dada la oportunidad de acortar las distancias, se denota para este módulo en el sistema que toda aquella noticia que no pueda registrarse en la oficina Central pueda existir una sincronización directa de la misma los datos correspondiente a la noticia que serán enviados a la oficina central de forma encriptada a través de un servicio Web que registrará en un formato Xml todas las noticias enviadas desde los departamentos y posteriormente las descriptará para ser almacenadas en la base de datos del sistema de noticias.

### **Entrega**

El proyecto se dividirá en Fases de entrega progresiva:

- 1er Entrega proyecto, 17-marzo, Análisis del proyecto.
- 2do Entrega proyecto, 21-abril, Módulo principal y de seguridad.
- 3er Entrega proyecto, 28-mayo, Módulo de noticias y sincronización.

El proyecto será entregado en grupos de 4 estudiantes máximo.

### **Temas de evaluación**

#### **Tecnologías**

- JSF(Java Server Faces)
- Servlets
- Struts
- Hibernate

#### **Implementaciones**

- Manejo de Estructuras Dinámicas usando la API Collection
- Conexiones a Bases de Datos con JNDI y JDBC
- Envío de Mails

Manejo de Multimedia Audio/Video

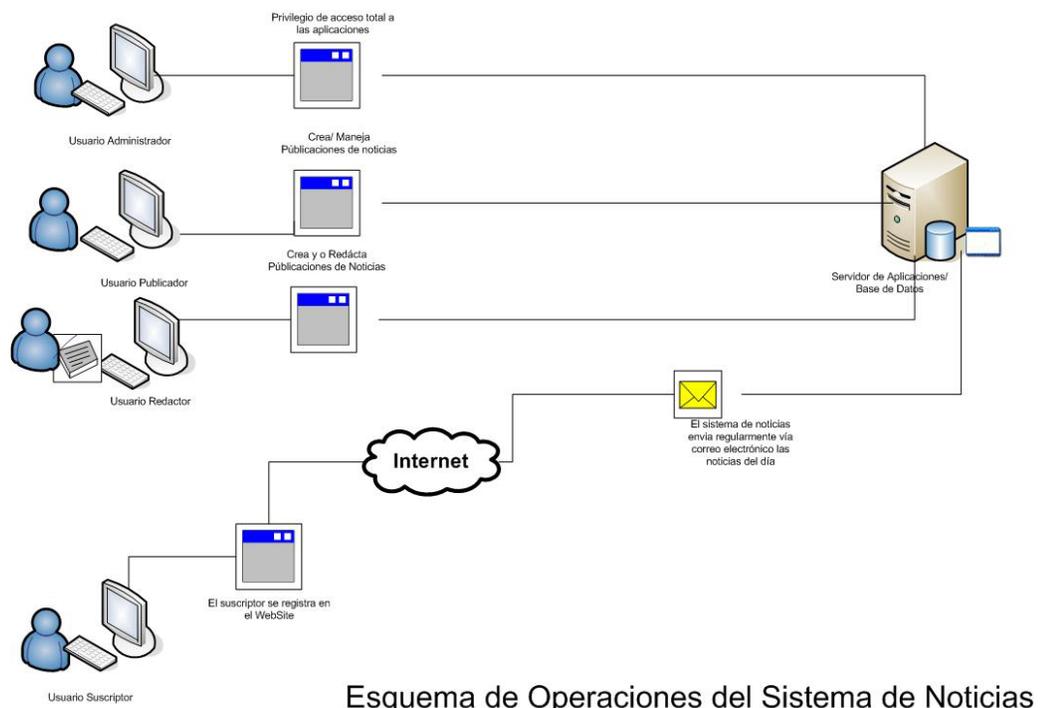
Manejo de Imágenes

WebServices con Seguridad

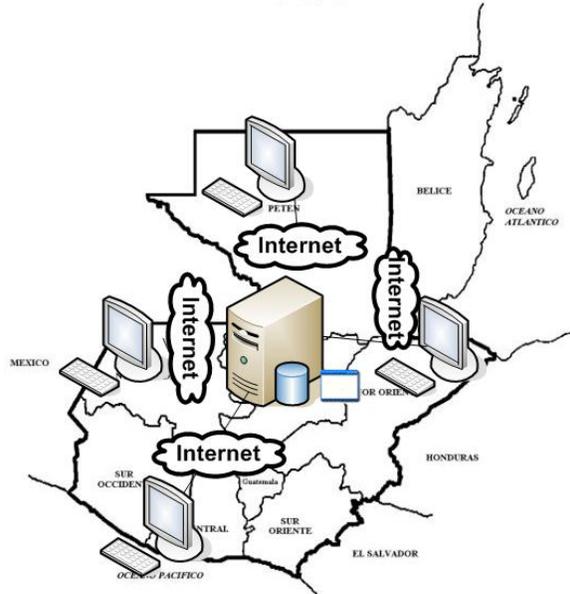
XML

Transacciones

## Esquematación del Funcionamiento del Proyecto



## ESQUEMA DE SINCRONIZACION DE NOTICIAS DEPARTAMENTOS/CENTRAL DE NOTICIAS



### DERCAS SISTEMA DE NOTICIAS

#### Descripción:

Sistema que permite expandir noticias a través de un sitio web, dicho sitio debe constar de varios módulos los cuales se listan a continuación:

- Módulo Principal
- Módulo de Seguridad
- Módulo de Noticias
- Módulo de Sincronización.

### **Módulo Principal:**

Como su nombre lo indica este módulo es el principal y debe unir el resto de módulos, en este módulo debe implementarse la presentación inicial del sitio, tiene que tener: vista previa de noticias, publicidad, menú principal, desde donde se podrá ingresar al resto de módulos, etc.

Cuando un usuario ingresa al sitio es un usuario invitado, no tiene que registrarse, pero también debe tratarse como un usuario, el sistema tiene que obtener los permisos/menús que le corresponde al usuario y luego presentarlos.

Un invitado puede registrarse con un usuario (previamente ingresado en el sistema), desde la página principal, la página que aparece cuando está como invitado, y al registrarse, el sistema tiene que agregar o quitar menús y permisos que el sistema le había fijado cuando ingresó como invitado.

La página principal tiene que tener una opción de registro como nuevo suscriptor, donde el usuario actualmente invitado, tiene que ingresar sus datos personales, ciertos datos son obligatorios (nombre, correo electrónico, etc.), estos datos sirven para que el sistema le envíe los links de las nuevas noticias.

### **Módulo de Seguridad:**

*Permisos:* los permisos son ingresados cuando se instala el sitio, es decir pueden ir incluidos en el script de creación de base de datos, etc.

*Menús:* igual que los permisos estos son ingresados cuando se instala el sitio.

*Roles:* Los roles son grupos de permisos y menús, que facilitan el trabajo a los administradores, cuando crea un usuario no tiene que asignarle permiso a permiso y menú a menú si no puede solo asignarle un rol y este ya incluye varios permisos y menús, el sistema tiene que permitir ingresar nuevos roles y definirlos, definirlos no es mas que indicarle que permisos y menús componen dicho rol, dentro de los roles principales deben existir: Administradores, Publicadores, Redactores, Suscriptores, Invitados.

*Usuarios:* Los usuarios los puede crear otro usuario que tenga el privilegio, y tienen que tener un rol, un usuario también lo puede crear un invitado, al registrarse como un nuevo suscriptor, pero esta forma de crear usuarios es especial ya que automáticamente se le tiene que asignar el rol de suscriptor, que ya tiene que estar creado y definido.

Un usuario con privilegios puede crear, modificar y eliminar cada uno de los objetos anteriores, siempre y cuando tenga el privilegio.

### **Módulo de Noticias:**

Este módulo tiene que tener las siguientes opciones:

*Secciones:* Son grupos de noticias, en la página principal las noticias no se muestran directamente si no por secciones.

*Noticias:* con esta opción, un usuario puede ingresar nuevas noticias, una noticia consta de un nombre, una descripción, fotos, videos, y un resumen, el resumen a la vez consta también de un texto pero resumido de la noticia y fotos pero de menor calidad, este resumen es para incluirlo en la página principal, el sistema tiene que permitir modificación y eliminación de las noticias o de sus componentes.

*Publicación de Noticias:* El sistema no publica todas las noticias que están actualmente almacenadas, la opción de publicación de noticias, permite indicarle que noticias se publican y durante que fechas se van a mantener activas.

*Desactivación de Noticias:* El sistema tiene que incluir un proceso que se ejecuta independientemente del servidor web, pero puede ser controlado desde una página del sitio, dicho proceso se encarga de verificar que noticias se tienen que desactivar a diario y cuando una noticia se desactiva, al ingresar al sitio ya no tiene que aparecer publicada la noticia.

*Envío de Noticias:* El envío de noticias al igual que la desactivación automática de noticias, lo tiene que hacer un proceso que también se tiene que controlar desde el sitio pero que es independiente del servidor web, este proceso selecciona todas las noticias a diario, y a cierta hora, fijada por un usuario con privilegios, las tiene que enviar a los correos de los suscriptores.

### **Módulo de Sincronización:**

Se desea acortar distancias y tiempo, por medio de un módulo independiente del sitio web, dicho módulo es un programa que permite todas las opciones del módulo de noticias, que en cierto tiempo se sincronizan hacia el servidor.

Se asume que la pc donde se almacena este módulo no consta con un servidor de base de datos, por lo que el sistema tiene que almacenar toda la información en archivos xml, y en carpetas, que luego para sincronizar hacia el servidor tiene que encriptar.

El servidor debe desencriptar esta información y luego almacenarla en la base de datos, también se desea que toda esta información sea correcta cuando llegue al servidor por lo que el sistema tiene que asegurarse que esta información es la misma que está en la máquina cliente que sincroniza la información.

**Nota:** De forma intrínseca todas las opciones constan de búsqueda, tomando en cuenta algunos campos como filtros.

El siguiente diagrama de despliegue le puede servir como guía de lo que se necesita del sistema.

### **Datos Técnicos:**

- Los procesos pueden ser web-services, de esta forma podría publicar un web service y al mismo tiempo heredar de un Servlet para manipularlo desde el sitio web.
- Para Encriptar los archivos desde el manipulador de noticias, y desencriptarlos en el servidor puede utilizar un algoritmo simétrico del JCE.
- Para comprobar que los archivos sincronizados llegan correctamente puede utilizar los algoritmos de encriptación MD5, SHA, etc.
- Para facilitar y optimizar el uso de archivos XML utilice componentes basados en tecnología DOM y lenguajes de consulta como el xpath.
- Reutilice todas las clases del módulo de noticias en el Manipulador de noticias.

### **Tecnologías a Utilizar:**

Entre las tecnologías que puede utilizar para la construcción del sistema están:

- Servlets.
- Jsp.
- JDBC.
- JCE.
- JavaMail.

- Web Services.
- EJBs. (Enterprise Java Beans)
- JB. (Java Beans)
- JSTL.
- XML, xpath.
- JSF (Java Server Faces).

Servidores WEB Recomendados: Apache Tomcat, WebSphere.

Manejadores de Base De Datos Recomendados: MySQL, Postgresql, MaxDB.

IDEs Recomendados: Eclipse, NetBeans.



## NOTAS FINALES PRIMER SEMESTRE 2007

#	CARNET	NOMBRE	10 pts	10 pts	10 pts	10 pts	40 pts	20 pts	100/pts
			1er	2do	3er	Tareas	Laboratorio	Final	Nota final
			Parcial	Parcial	Parcial				
1	9313118	CESAR CALDERON DE LEON	2.67	2.00	0.00	0.00	0.00	0.00	<b>4.67</b>
2	199515907	MANOLO GRAMAJO RODAS	7.33	7.33	5.50	1.60	0.00	10.00	<b>31.77</b>
3	200010645	FERNANDO JOSE BOIZAN TELLO	6.00	0.00	0.00	6.40	0.00	0.00	<b>12.40</b>
4	200010689	ADOLFO JAVIER ESPINOZA PAZ	7.33	9.33	6.50	8.40	40.00	11.00	<b>82.57</b>
5	200010961	EFRAIN SARAVIA PACHECO	8.00	6.00	7.00	8.40	40.00	12.60	<b>82.00</b>
6	200011186	MARCO TULIO ANDRADE	4.44	3.33	2.50	1.60	17.00	8.80	<b>37.68</b>
7	200011212	CARLOS JOAQUIN MARROQUIN	7.33	6.00	5.00	1.60	35.00	7.60	<b>62.53</b>
8	200011743	JOSMAN DANIEL FLOREZ JUAREZ	6.67	7.33	5.00	6.80	15.00	10.40	<b>51.20</b>
9	200018480	MANUEL ELIAS MUHUN ESQUIT	5.33	0.00	0.00	6.40	0.00	0.00	<b>11.73</b>
10	200112813	JUANLUIS ESTRADA MONTERROSO	4.67	7.33	7.00	8.40	18.00	11.40	<b>56.80</b>
11	200112857	EDNA DONINEY CARDONA MUÑOZ	5.33	0.00	0.00	1.60	0.00	4.80	<b>11.73</b>
12	200112975	BAYRI GONZALEZ	6.67	6.67	6.00	3.10	28.80	10.00	<b>61.23</b>
		MYNOR ADALBERTOAGUSTIN							
13	200113042	MARTINEZ	4.00	3.33	5.00	2.00	0.00	12.60	<b>26.93</b>
		SONIA LUCRECIA SAGASTUME							
14	200117357	MIRANDA	6.67	3.33	6.00	6.50	0.00	6.00	<b>28.50</b>
15	200117517	EDVING DAVID MORALES ARANA	6.67	8.00	7.00	9.50	40.00	16.00	<b>87.17</b>
16	200130016	VICTOR JACOB SOLIS URIZAR	5.33	0.00	0.00	0.00	0.00	8.60	<b>13.93</b>
17	200130024	LUIS FERNANDO COTI ITZEP	8.00	4.67	5.50	9.50	0.00	0.00	<b>27.67</b>

18	200212527	RUDY ANIBAL CAJBON RUCAL	6.00	4.67	5.00	8.40	0.00	10.20	<b>34.27</b>
19	200212588	EDWIN ALFREDO SANTALEZ LOPEZ	5.33	0.00	0.00	8.40	0.00	0.00	<b>13.73</b>
20	200216457	AURA LILY NORIEGA FIGUEROA	6.67	4.00	3.33	8.00	5.00	9.20	<b>36.20</b>
21	200217819	CARLOS EDUARDO ESCOBAR R. JAVIER ESTUARDO HERNANDEZ	7.33	7.33	8.00	6.80	30.00	7.20	<b>66.67</b>
22	200312415	CASTELLANOS	4.67	8.00	7.00	8.40	25.00	4.20	<b>57.27</b>
23	200312466	EDGAR JOSUE POLANCO SIGUERE	6.67	8.67	7.50	8.00	25.00	11.60	<b>67.43</b>
24	200312514	JOSE LUIS CHITIC	5.33	6.67	6.00	6.40	22.00	8.20	<b>54.60</b>
25	200312785	LUIS CARLOS HIDALGO	6.00	8.00	3.00	10.00	25.00	12.60	<b>64.60</b>
26	200312890	LUIS EMILIO OROZCO FUENTES	6.00	7.33	8.00	10.00	28.00	14.00	<b>73.33</b>
27	200313153	PEDRO BALDEMAR RAMOS JUAREZ	7.33	6.00	3.00	2.00	0.00	6.00	<b>24.33</b>
28	200313198	FERNANDO AVENDAÑO	4.67	7.33	4.00	4.00	18.00	11.40	<b>49.40</b>
29	200313261	JASON ARTURO GALVEZ E. MARYLIN BEATRIZ PAXTOR	8.00	6.67	8.67	10.00	35.00	16.40	<b>84.73</b>
30	200313390	GARCIA	6.00	6.00	0.00	1.60	0.00	0.00	<b>13.60</b>
31	200318963	JOSE FERNANDO PEREZ SIPAQUE AUGUSTO ELIAS MAZARIEGOS	8.67	7.33	7.00	8.00	0.00	8.20	<b>39.20</b>
32	200412406	CANTORAL	5.33	4.00	0.00	8.00	0.00	0.00	<b>17.33</b>
33	200412874	JULIO ENRIQUE VIVAS CASTILLO	4.67	7.33	6.00	8.40	25.00	11.00	<b>62.40</b>
34	200413190	GERSON GAMALIEL PEREZ ORTEGA	6.00	7.33	6.50	10.00	25.00	16.00	<b>70.83</b>
35	200413700	JOSE ESTUARDO DIAZ CAMEY JERSON ALEXANDER MONTERROSO	7.33	6.00	5.50	6.80	12.00	8.60	<b>46.23</b>
36	200413740	GARCIA	5.33	8.00	1.50	8.00	34.00	8.00	<b>64.83</b>
37	200511704	JORGE ALEJANDRO DIAZ	8.67	8.67	8.00	6.80	31.00	6.20	<b>69.33</b>

		CARLOS ROBERTO SANDOVAL							
38	200511795	REYES	8.00	8.67	8.00	10.00	26.00	12.40	<b>73.07</b>
		JULIO EDUARDO MORALES							
39	200511905	TOLEDO	8.00	6.67	7.50	10.00	26.00	12.20	<b>70.37</b>
40	200512059	PABLO SERGIO ROMEO VELIZ	6.00	9.33	7.00	8.40	25.00	0.00	<b>55.73</b>
41	200512079	VICTOR LEONEL OROZCO LOPEZ	8.00	9.33	7.50	8.40	33.00	17.20	<b>83.43</b>
42	200512080	JOSUE BENJAMIN AVILA MARTINEZ	6.00	4.00	3.50	8.00	0.00	7.80	<b>29.30</b>
43	200512085	MARIA ISABEL JUAREZ PACHECO	8.00	6.67	5.50	10.00	5.00	12.00	<b>47.17</b>
44	200512086	MARIO RAMON CASTELLANOS	8.00	7.33	4.50	4.80	29.00	9.00	<b>62.63</b>
		IVINA DEL CARMEN ACUÑA							
45	200512138	MORALES	6.00	5.33	7.50	8.40	25.00	13.60	<b>65.83</b>
		FRANCISCO ALFONSO PINEDA							
46	200512141	JIMENEZ	6.67	10.00	8.00	8.00	28.60	0.00	<b>61.27</b>
47	200516280	EDVIN OMAR NAJERA ESPINO	5.33	7.33	6.50	8.40	30.00	9.40	<b>66.97</b>
		JENNIFER ILEANA MARTINEZ							
48	200516341	DONIS	6.00	6.67	6.00	8.40	25.00	16.60	<b>68.67</b>



## EXAMEN DE PRIMERA RETRASADA PRIMER SEMESTRE 2007

---

*I. SERIE: INSTRUCCIONES: Desarrolle en base a los conceptos de persistencia y manejo de estructuras dinámicas el siguiente problema planteado.*

En una empresa de gran flujo de operaciones requieren optimizar el tiempo de procesamiento y búsqueda de las mismas dentro de su repositorio de información para ello se deben de recuperar los datos desde dicho repositorio de datos( base de datos en Oracle 10g) y cuya aplicación esta basada en tecnología JSF(Java Server Faces). La extracción de dicha información se ejecuta mediante una invocación a la base de datos a través de una consulta del usuario. Plantee la forma de extracción, captura, y presentación de información al usuario, tomando en cuenta los aspectos de manejo de seguridad en la consulta y procesamiento de datos en la aplicación.

*Respuesta. //*

- 1. Para el manejo de la extracción de la información utilizo alguna estructura dinámica del api collection para capturar la data que deseo procesar y realizo un tipo de fetch de data para presentar poco a poco los datos al usuario.*
- 2. Respecto a la parte de seguridad puedo utilizar*
  - a. Encriptación de los datos pero se descuida la parte de optimización del procesamiento de las transacciones*
  - b. La persistencia resolvería la parte de seguridad generando o agregando un layer de lógica del negocio a nuestra aplicación.*
  - c. Para el procesamiento de datos establezco el manejo de transacciones directamente sobre la base de datos conjuntamente con la aplicación.*



## EXAMEN DE SEGUNDA RETRASADA PRIMER SEMESTRE 2007

---

*I. SERIE: INSTRUCCIONES: Desarrolle en base a los conceptos de persistencia y manejo de estructuras dinámicas el siguiente problema planteado.*

En una empresa de gran flujo de operaciones requieren optimizar el tiempo de procesamiento y búsqueda de las mismas dentro de su repositorio de información para ello se deben de recuperar los datos desde dicho repositorio de datos( base de datos en Oracle 10g) y cuya aplicación esta basada en tecnología JSF(Java Server Faces). La extracción de dicha información se ejecuta mediante una invocación a la base de datos a través de una consulta del usuario. Plantee la forma de extracción, captura, y presentación de información al usuario, tomando en cuenta los aspectos de manejo de seguridad en la consulta y procesamiento de datos en la aplicación.

**II. SERIE: INSTRUCCIONES: Proporcione el concepto de los siguientes elementos (en español). No hay un mínimo de líneas, siempre y cuando se describan las características que los identifican y caracterizan.**

1. Enterprise Bean, Session Bean, Message Driven Bean.
2. Persistence.
3. Securing Web Applications.
4. Securing the Application Server.
5. Custom Tags.
6. Java Server Faces Application.
7. JAX-WS.
8. JAXB Architecture.
9. StAX
10. Java EE Containers.



## NOTAS PRIMERA RETRASADA PRIMER SEMESTRE 2007

---

			40 pts	40 pts	20 pts	100/pts
					1er	
#	CARNET	NOMBRE	Laboratorio	Zona	Retrasada	Nota final
1	200512059	PABLO SERGIO ROMEO VELIZ	25	30.73	11.33	<b>67.07</b>
		FRANCISCO ALFONSO PINEDA				
2	200512141	JIMENEZ	28.60	32.67	12.67	<b>73.93</b>

## NOTAS SEGUNDA RETRASADA PRIMER SEMESTRE 2007

---

			40 pts	40 pts	20 pts	100/pts
					2da	
#	CARNET	NOMBRE	Laboratorio	Zona	Retrasada	Nota final
		JAVIER ESTUARDO HERNANDEZ				
1	200312415	CASTELLANOS	25	28.07	15.6	<b>68.67</b>
2	200312785	LUIS CARLOS HIDALGO	25	27.00	12.6	<b>64.6</b>

---