



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

**READECUACIÓN CURRICULAR EN EL ÁREA DE TECNOLOGÍAS DE LA  
INFORMACIÓN EN EL ITCOE DE LA FACULTAD DE INGENIERÍA,  
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**Yuri Asucena Castro Estrada**

Asesorado por el Ing. César Salvador Rojas Argueta

Guatemala, septiembre de 2012

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**READECUACIÓN CURRICULAR EN EL ÁREA DE TECNOLOGÍAS DE LA  
INFORMACIÓN EN EL ITCOE DE LA FACULTAD DE INGENIERÍA,  
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA LA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**YURI ASUCENA CASTRO ESTRADA**

ASESORADO POR EL ING. CÉSAR SALVADOR ROJAS ARGUETA

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERA EN CIENCIAS Y SISTEMAS**

GUATEMALA, SEPTIEMBRE DE 2012

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Juan Carlos Molina Jiménez
VOCAL V	Br. Mario Maldonado Muralles
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**


DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADORA	Inga. Mirna Ivonne Aldana Larrazábal
EXAMINADOR	Ing. Edgar Estuardo Santos
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **READECUACIÓN CURRICULAR EN EL ÁREA DE TECNOLOGÍAS DE LA INFORMACIÓN EN EL ITCOE DE LA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 1 de septiembre de 2011.



**Yuri Asucena Castro Estrada**

Guatemala 11 de abril de 2012

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

Estimada Ingeniera Sarmiento Zeceña de Serrano.

Por este medio atentamente le informo que como Asesor de la Práctica del Ejercicio Profesional Supervisado (E.P.S), de la estudiante universitaria **Yuri Asucena Castro Estrada** de la carrera de Ingeniería en Ciencias y Sistemas, con carné No **200413023**, procedí a revisar el informe final, cuyo título es **"READECUACIÓN CURRICULAR EN EL ÁREA DE TECNOLOGÍAS DE LA INFORMACIÓN EN EL ITCOE DE LA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**.

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

Sin otro particular, me es grato suscribirme.

Atentamente,



**César Salvador Rojas Argueta**  
Ingeniero en Ciencias y Sistemas  
Colegiado No. 10375

Ing. Cesar Salvador Rojas Argueta  
Colegiado. No. 10375  
Asesor EPS



Guatemala, 27 de abril de 2012.  
REF.EPS.DOC.645.04.2012.

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

Estimada Ingeniera Zeceña de Serrano.


Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) de la estudiante universitaria de la Carrera de Ingeniería en Ciencias y Sistemas, **Yuri Asucena Castro Estrada** Carné No. **200413023** procedí a revisar el informe final, cuyo título es **“READECUACIÓN CURRICULAR EN EL ÁREA DE TECNOLOGÍAS DE LA INFORMACIÓN EN EL ITCOE DE LA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**.

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

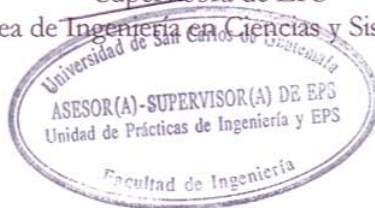
Sin otro particular, me es grato suscribirme.

Atentamente,

“Id y Enseñad a Todos”

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

FFAPdM/RA





Guatemala, 27 de abril de 2012.  
REF.EPS.D.464.04.2012.

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

Estimado Ingeniero Perez Turk.

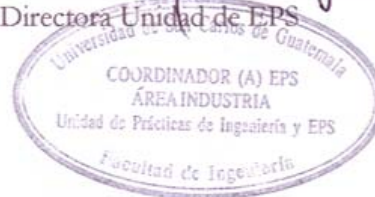
Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **“READECUACIÓN CURRICULAR EN EL ÁREA DE TECNOLOGÍAS DE LA INFORMACIÓN EN EL ITCOE DE LA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, que fue desarrollado por la estudiante universitaria **Yuri Asucena Castro Estrada** carné No. **200413023** quien fue debidamente asesorada por el Ing. César Salvador Rojas Argueta y supervisada por la Inga. Floriza Felipa Ávila Pesquera de Medinilla.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor y 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 Zeceña de Serrano  
Directora Unidad de EPS



NISZds/ra



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

Guatemala, 16 de Mayo de 2012

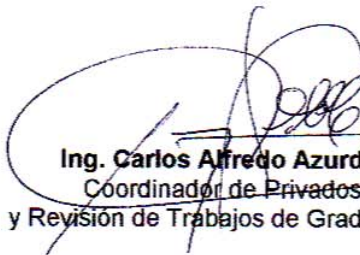
Ingeniero  
**Marlon Antonio Pérez Turk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas


Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación-EPS de la estudiante **YURI ASUCENA CASTRO ESTRADA**, carné **2004-13023**, titulado: **"READECUACIÓN CURRICULAR EN EL ÁREA DE TECNOLOGÍAS DE LA INFORMACIÓN EN EL ITCOE DE LA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

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

Atentamente,

  
**Ing. Carlos Alfredo Azurdía**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación





E  
S  
C  
U  
E  
L  
A  
  
D  
E  
  
C  
I  
E  
N  
C  
I  
A  
S  
  
Y  
  
S  
I  
S  
T  
E  
M  
A  
S

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación titulado **“READECUACIÓN CURRICULAR EN EL ÁREA DE TECNOLOGÍAS DE LA INFORMACIÓN EN EL ITCOE DE LA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, presentado la estudiante YURI ASUCENA CASTRO ESTRADA, aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**

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



*Guatemala, 21 de septiembre 2012*



DTG. 459.012

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: **READECUACIÓN CURRICULAR EN EL ÁREA DE TECNOLOGÍAS DE LA INFORMACIÓN EN EL ITCOE DE LA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentada por la estudiante universitaria **Yuri Asucena Castro Estrada**, autoriza la impresión del mismo.

IMPRÍMASE:

  
Ing. Murphy Olimpo Paiz Recinos  
Decano



Guatemala, 24 de septiembre de 2012.

/gdech

## **ACTO QUE DEDICO A:**

<b>Dios</b>	Por darme la oportunidad de culminar esta meta.
<b>Mis padres</b>	Juan José Castro y Vilma Consuelo Estrada, por todo el apoyo que me han brindado. Por la confianza que pusieron en mí cuando decidí empezar este viaje sola.
<b>Universidad de San Carlos de Guatemala</b>	Por brindarme la ayuda, mediante su programa de becas préstamo, en el momento en que lo necesité.
<b>Mis hermanos</b>	Juan Antonio y Nora Yesenia, por ser mis dos pilares y estar allí para apoyarme o regañarme cuando fue necesario.
<b>Ing. César Rojas</b>	Por ser mi hermano de batalla y apoyarme tanto en lo académico como en lo personal.
<b>Mis amigos</b>	Por ser mi segunda familia y brindarme momentos de felicidad inolvidables.

# ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	III
GLOSARIO .....	V
RESUMEN.....	XI
OBJETIVOS.....	XIII
INTRODUCCIÓN .....	XV
1. FASE DE INVESTIGACIÓN .....	1
1.1. Antecedentes de la empresa .....	1
1.2. Manejo de procesos basado en la metodología del TCS .....	3
1.2.1. Misión .....	3
1.2.2. Visión.....	3
1.2.3. Servicios que realiza.....	4
1.3. Planteamiento del problema y su solución .....	5
1.4. Cursos impartidos.....	6
2. FASE TÉCNICO-PROFESIONAL .....	9
2.1. Readequación curricular .....	9
2.1.1. Estudio de los contenidos de cursos .....	10
2.1.2. Entrevistas a <i>trainers</i> .....	10
2.1.3. Comparativa de cursos .....	11
2.1.4. Elaboración de programa .....	11
2.2. Cursos actualizados .....	12
2.2.1. Área de Ingeniería de software.....	13
2.2.1.1. Introduction to software engineering –ISE-.....	13

2.2.1.2.	Software Quality Management -SQM-.....	13
2.2.1.3.	Solution architecture –SA-.....	14
2.2.2.	Desarrollo de software.....	15
2.2.2.1.	Técnicas básicas de programación.....	15
2.2.2.2.	Active Server Page -ASP-.....	16
2.2.2.3.	OoP c++.....	16
2.2.2.4.	Introducción a la programación de computadoras 1-IPC 1-.....	17
2.2.2.5.	Introducción a la programación de computadoras 2 -IPC 2-.....	17
2.2.3.	Programación en Java.....	18
2.2.3.1.	Java básico.....	18
2.2.3.2.	Java intermedio.....	19
2.2.3.3.	Java avanzado.....	19
2.3.	Plantilla general de curso.....	20
3.	FASE ENSEÑANZA-APRENDIZAJE.....	23
	CONCLUSIONES.....	27
	RECOMENDACIONES.....	29
	BIBLIOGRAFÍA.....	31
	APÉNDICES.....	33

## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Página principal de cursos .....	24
2.	Página de área específica .....	24
3.	Información de curso .....	25



## GLOSARIO

<b>Book</b>	Traducción de libro.
<b>CONCYT</b>	Consejo Nacional de Ciencia y Tecnología.
<b>DBMS</b>	Sistemas de gestión de bases de datos (SGBD). Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos.
<b>DCL</b>	Data Control Language, es un lenguaje proporcionado por el sistema de gestión de base de datos que incluye una serie de comandos SQL que permiten al administrador controlar el acceso a los datos contenidos en la base de datos.
<b>DDL</b>	Data Definition Language, es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma, llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.



<b>DML</b>	Data Manipulation Language, es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.
<b>Encapsulamiento</b>	Acción de ocultar el estado de los miembros de un objeto de manera que sólo se puede cambiar mediante las operaciones definidas para ese objeto.
<b>Framework</b>	Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.
<b>Hardware</b>	Conjunto de los componentes que integran la parte material de una computadora.
<b>INDO</b>	Gobierno de la India.
<b><i>Instructor Guide</i></b>	Documento que utiliza un instructor para impartir un curso. El documento contiene en detalle los objetivos, prerequisites, módulos, contenidos, planificación, ponderación, criterio de evaluación e instrucciones de ejecución de las actividades a realizar para un curso específico.

<b>IT</b>	Tecnologías de la información. Es el estudio, diseño, desarrollo, innovación puesta en práctica, ayuda o gerencia de los sistemas informáticos computarizados, particularmente usos del software y hardware.
<b>ITCoE</b>	Information Technology Education Centre of Excellence. En español, Centro de Excelencia en Educación de Tecnologías de la Información.
<b>JDBC</b>	Java Database Connectivity, es una interfaz de programación que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del Sistema Operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.
<b><i>Kaisen</i></b>	Mejoramiento. Estrategia de calidad en la empresa y en el trabajo, individual y colectivo.
<b>LinQ</b>	Language Integrated Query, es un proyecto de Microsoft que agrega consultas nativas semejantes a las de SQL a los lenguajes de la plataforma .NET.
<b>Microsoft</b>	Empresa multinacional de origen estadounidense dedicada al sector de la informática.

<b>Normalización</b>	En informática consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.
<b>ODBC</b>	Open DataBase Connectivity, es un estándar de acceso a bases de datos que hace posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos almacene los datos.
<b>OOP</b>	Programación orientada a objetos. Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos.
<b><i>Outsourcing</i></b>	Subcontratación. Es el proceso económico en el cual una empresa mueve o destina los recursos orientados a cumplir ciertas tareas hacia una empresa externa por medio de un contrato. Esto se da especialmente en el caso de la subcontratación de empresas especializadas.
<b>Polimorfismo</b>	En informática, se refiere a la posibilidad de enviar un mensaje a un grupo de objetos cuya naturaleza puede ser heterogénea.
<b><i>Presentation</i></b>	Traducción de presentación.

<b>Programación</b>	Es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales.
<b>SICA</b>	Sistema de Integración Centroamericana.
<b>Software</b>	Equipamiento lógico o soporte lógico de un sistema informático; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.
<b>SQL</b>	Structured Query Language, es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas.
<b>SubVersion</b>	Sistema de control de versiones, software libre creado bajo la licencia Apache/BSD.
<b>TIC</b>	Tecnologías de la información y la comunicación. Agrupan los elementos y las técnicas usadas en el tratamiento y la transmisión de las informaciones, principalmente de informática, Internet y telecomunicaciones.

<b><i>Trainer</i></b>	Entrenador. Persona a cargo de ejecutar los cursos que se imparten en el ITCoE.
<b>Tupla</b>	También conocido como registro. Representa un objeto único de datos implícitamente estructurados en una tabla. Es un conjunto de campos que contienen los datos que pertenecen a una misma repetición de entidad.
<b>UML</b>	Lenguaje Unificado de Modelado, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.
<b>URL</b>	Uniform resource locator, es una secuencia de caracteres, de acuerdo con un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones, presentaciones digitales, etcétera.
<b>Word Wide Web</b>	World Wide Web (WWW) o Red informática mundial, es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.

## **RESUMEN**

La tecnología es cambiante, por lo que se necesita mantener al día la información que se posee de cada herramienta informática que sea utilizada.

El mismo principio aplica para materias que se imparten, principalmente en el área de la tecnología, es necesario que el contenido esté actualizado para poder brindar educación de calidad.

El presente trabajo expone el proceso de readecuación llevado a cabo en el ITCoE, sobre los cursos del área de Lenguajes de Programación e Ingeniería de Software. Dicha readecuación consistió en la elaboración de programas actualizados por cada curso de las áreas mencionadas, siendo un total de 11 los cursos que fueron evaluados y readecuados.

De la misma manera, se desarrolló una plantilla para que los cursos impartidos en ITCoE que no pertenecen a las áreas mencionadas puedan adaptarse a esta metodología de forma fácil y rápida.



# OBJETIVOS

## General

Administrar el ITCoE mediante la planificación, apertura y desarrollo de cursos de IT, a través de la metodología y los procesos existentes desde su creación.

## Específicos

1. Readeclarar y actualizar los planes de estudio de once cursos impartidos en ITCoE en el área de IT.
2. Estandarizar el programa de cada curso impartido en el ITCoE para facilitar el desarrollo de los mismos.
3. Crear una plantilla que sirva de documento base para definición de los contenidos de los cursos actuales y próximos a impartir en ITCoE.
4. Definir un sistema de evaluación con base en el contenido y metodología desarrollada en cada uno de los once cursos objetivos de este estudio.
5. Fortalecer la metodología de enseñanza que se basa en la filosofía *I do, We do, You Do*.



6. Crear un fondo documental que proporcione la información de los cursos impartidos por ITCoE a estudiantes y población en general.
7. Documentar y definir los contenidos y sistemas de evaluación de once cursos impartidos en ITCoE.

## INTRODUCCIÓN

Para formar profesionales en Tecnologías de la Información (IT) de alta calidad y rendimiento se debe contar con personal capacitado y con el mejor pénsum de estudio posible dentro de sus recursos. La Universidad de San Carlos de Guatemala, a través de la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería, juega un papel fundamental como la principal casa de estudios para la formación de personal con alta calidad técnico/profesional en TICs y ciencias de la computación.

La Facultad de Ingeniería y la Escuela de Ciencias y Sistemas deben responder a esos cambios y avances tecnológicos haciendo ajustes en la metodología de enseñanza y actualizaciones en los pénsum de estudios para generar profesionales competitivos.

Es así como en el 2006 se funda, gracias a una donación realizada por el Gobierno de la India, el IT Education Centre of Excellence India-Guatemala (ITCoE), contando con dos laboratorios equipados con computadoras e infraestructura necesaria para capacitar al personal local en el área de tecnologías de la información (IT).

La metodología de enseñanza-aprendizaje del ITCoE fue transmitida por el personal de TATA Consultancy Services (TCS) en su estancia en Guatemala hacia el personal local, pero como es de dominio público, la tecnología es una de las ciencias que cambia con mayor rapidez en la actualidad, por lo que la enseñanza debe estar actualizándose continuamente.

La documentación en cualquier institución es de vital importancia, ya que provee la información necesaria para controlar las actividades que derivan de sus procesos. En el caso de instituciones educativas y académicas, llevar el control de los cursos y materias impartidas es fundamental, así como la buena adecuación de los contenidos programáticos de los cursos que imparten.

Documentar correctamente los contenidos de los cursos que se imparten provee calidad a los mismos y les asegura una existencia independiente del tiempo y las personas. Una programación debe indicar de manera precisa qué es lo que se debe impartir, cuándo impartirlo y cómo impartirlo, buscando la mejor manera de mantener orden y organizar los recursos con los que cuenta la institución, con el fin de utilizarlos eficientemente.

El presente trabajo de graduación trata sobre la readecuación curricular y documentación de los cursos impartidos en el India-Guatemala IT Centre of Excellence (ITCoE) en el área de tecnologías de la información.

# **1. FASE DE INVESTIGACIÓN**

## **1.1. Antecedentes de la empresa**

Debido a la necesidad de educación tecnológica de la región y como parte de las iniciativas para el desarrollo centroamericano, en febrero de 2005, el Gobierno de la India (INDO) junto con el Sistema de Integración Centroamericana (SICA), llevaron a cabo una reunión en donde se propuso la creación e instalación de los IT Education Centres en los países miembros del SICA, el cual busca la reducción de la brecha digital que existe entre estos países.

El Gobierno de Guatemala junto con el de la India, por medio de su representante para el país, H. E. Mrs. Primrose Sharma, acordaron los términos y detalles para instalar en el país un centro para la capacitación tecnológica con soporte y apoyo de la India, país dominante en el área de IT en el mundo.

Además, designó al Consejo Nacional de Ciencia y Tecnología (CONCYT) la tarea de buscar una ubicación adecuada para la instalación del centro. La Universidad de San Carlos de Guatemala (USAC) fue elegida para albergar el centro, por ser la más grande del país y su reputación de universidad estatal, siendo la cuarta universidad fundada en América.

El Gobierno de la India designó a TATA Consultancy Services (TCS) de India para la implementación del IT Centre of Excellence (ITCoE).

TATA Consultancy Services es una empresa de servicios de IT, que ofrece soluciones de negocio y *outsourcing* que entrega resultados reales, asegurando un grado de calidad y certeza que ninguna otra empresa puede dar. TCS ofrece un portafolio integral de soluciones y servicios IT a través de su única red global de entrega, reconocida por su excelencia y calidad de desarrollo.

El centro fue ubicado en la Facultad de Ingeniería, en donde el decano de la misma gestionó el acondicionamiento del espacio necesario para su instalación, colocándolo como una de sus prioridades. Los ingenieros Luis Vettorazzi y Armín Mazariegos, fueron designados para trabajar en conjunto con TCS y así llevar a cabo este proyecto.

El ITCoE fue creado con dos laboratorios de 26 computadoras cada uno, así como una oficina central con cuatro computadoras junto con dos servidores para manejar y gestionar los laboratorios.

Como parte del apoyo acordado, TCS envió a tres instructores para que capacitaran al personal nacional y de esta manera dar continuidad al proyecto. Ellos fueron: Mr. Mrutunjaya Panda, como *Centre Manager*, y Mr. Aashish Bajaj y Mrs. BimalJeetKaur (quien fue reemplazada por Mr. VikashVardhan) para realizar las labores de entrenamiento.

Siendo el ITCoE un centro de capacitación tecnológica dirigido a diferentes sectores de la sociedad, como educativo, industrial, académico, etcétera, desde su creación se han impartido capacitaciones a distintas unidades facultativas de la USAC, entidades de gobierno, instituciones educativas, etcétera

Una de las fortalezas del ITCoE es su metodología, tanto en la enseñanza como en lo administrativo. La primera se basa en la filosofía de *I do, we do, you do*, en donde el instructor durante su enseñanza incluye ejemplos, luego tanto el instructor como los alumnos realizan otros similares y, por último, los estudiantes realizan ejercicios ellos solos, de tal forma que sean capaces de resolverlos por ellos mismos.

## **1.2. Manejo de procesos basado en la metodología del TCS**

TCS considera como elementos importantes en la metodología aplicada a sus procesos, los siguientes puntos.

### **1.2.1. Misión**

“Ser líder en brindar a estudiantes y población en general capacitación y formación relacionadas con tecnología de punta que induzcan el mejoramiento de la calidad y nivel académico de estudiantes y población del país en general, teniendo como apoyo metodologías con estándares internacionales de calidad en servicios educativos”.

### **1.2.2. Visión**

“Para el 2012 brindar servicios sustentables de educación de alto nivel tecnológico a estudiantes y a la población en general apoyando en la tecnología de la información que se obtiene a través del beneficio que representa las donaciones de organizaciones internacionales a esta entidad”.

### **1.2.3. Servicios que realiza**

El ITCoE es una institución creada con el objetivo de ser un centro de capacitación en el área de IT. Entre los servicios que ofrece se pueden contar:

- Capacitación a estudiantes de la carrera de Ingeniería en Sistemas en temas afines a su carrera.
- Laboratorios equivalentes para los cursos de la carrera de Ingeniería en Sistemas, para Introducción a la programación y computación 1 e Introducción a la programación y computación 2.
- Capacitaciones para el personal administrativo de las diferentes unidades académicas de la Universidad de San Carlos de Guatemala.
- Capacitaciones para el personal técnico de los Ministerios del Gobierno en cursos avanzados de Tecnologías de la Información.
- Capacitaciones abiertas en diferentes niveles para estudiantes de colegios e institutos, así como para otras universidades del país.

Los cursos que se imparten en ITCoE están enfocados a diferentes niveles y sectores de la población, por lo que pueden ser cursados por cualquier persona que esté interesada. La única restricción que se tiene es que, por lo menos, tenga conocimientos básicos de computación y del área al que pertenezca el curso.

De igual manera, se cuenta con una pequeña biblioteca y préstamo de libros, que no es frecuentado. Se poseen dos hipótesis de la razón de dicha situación. La primera indica que este servicio no es muy conocido ni difundido; la segunda, es debido a que el contenido de los libros no es del todo actual y en algunos casos, hasta podrían considerarse obsoletos.

### **1.3. Planteamiento del problema y su solución**

El ITCoE es una institución creada con el objetivo de brindar conocimientos en el área de tecnologías de la información. En sus inicios fue impartida por personas capacitadas específicamente en cada área, estableciendo contenidos que en ese momento eran los indicados. Con el pasar de los años y el cambio de administración, este principio se ha ido desvaneciendo, siendo, en algunas ocasiones, una dificultad conocer la programación de cada curso.

Uno de los principales problemas detectados fue la poca continuidad al momento de realizar la documentación e informes de cada curso impartido, los contenidos indicados por TCS no fueron actualizados y en algunos casos los cursos no se impartían basándose en ellos debido al poco orden existente en anteriores administraciones.

El proyecto de EPS consistió en realizar una readecuación de los cursos existentes y la creación de un formato que permitió estandarizar el programa de cada uno. Se entiende por readecuación curricular, la revisión de los contenidos existentes de cada curso, verificación de la actualidad de dicha información, así como la comparativa de esta información respecto a la que están impartiendo los actuales *trainers*.



Así también, la readecuación incluyó la elaboración de un nuevo formato para establecer el contenido programático de los cursos impartidos, el cual cuenta con su versión resumida —disponible al público en general—, que contiene únicamente la cantidad de módulos y nombres de los mismos y la versión extendida, que contiene una especificación más detallada de cada módulo y temas a tratar en los mismos.

Dicho formato servirá de base para futuros cursos a impartirse. El formato mencionado contiene los siguientes puntos:

- Nombre del curso
- Descripción
- Objetivos
- Duración
- Contenido
- Prerrequisitos

Con esta información se podrá organizar de mejor manera, tanto el horario interno asignado para el desarrollo de clases como la distribución de salones y la capacidad del personal que impartirá las clases. Así también, permite brindarle un mejor producto a la persona que llega buscando información de un curso de su interés.

#### **1.4. Cursos impartidos**

Los cursos impartidos en ITCoE fueron creados por TCS con miras a cubrir las distintas áreas de las ciencias de la computación.

Las áreas con las cuales se inició el proyecto fueron:

- Lenguajes de programación
- Bases de datos
- Ingeniería del software

Debido a su importancia, actualmente cuenta con otra área de aplicación, en donde se imparten cursos a personas de otras facultades con el fin de motivarlos a involucrar la tecnología en sus profesiones. Esta área es directamente *Web*.

Este proyecto sólo abarca los cursos que pertenecen a las áreas de Lenguajes de Programación e Ingeniería de software, debido a que el tema de bases de datos, dado lo amplio que es, requiere conocimientos específicos en el tema.



## 2. FASE TÉCNICO-PROFESIONAL

### 2.1. Readecuación curricular

Se entiende como readecuación, la revisión de los contenidos existentes de cada curso impartido en ITCoE en el área de desarrollo de software. Con esta revisión se puede verificar la actualidad de la información y de esta manera saber si los contenidos impartidos son los adecuados o investigar nuevas tendencias tecnológicas que cubran las necesidades de cada curso.

La readecuación curricular a aplicar en ITCoE parte de la necesidad de tener a la mano, de manera fácil y rápida, los contenidos de los cursos que se imparten en esa institución.

Para realizar la readecuación se realizó una serie de pasos, los cuales eran repetitivos en cada curso actualizado. Estos pasos son los siguientes:

- Estudio de los contenidos del curso.
- Entrevista con *trainer* encargado del curso.
- Comparativa entre el contenido del curso y la información obtenida del *trainer*.
- Elaboración del programa del curso a partir de la información recapitulada entre el *trainer* y la información otorgada por TCS.

### **2.1.1. Estudio de los contenidos de cursos**

Cada curso desarrollado por TCS cuenta con la siguiente documentación:

- *Instructor Guide*
- *Presentations*
- *Book*
- Otros documentos

Dichos documentos fueron desarrollados en el 2005 por TCS con las tendencias y conceptos tecnológicos aplicados en el año mencionado. Después de su entrega no han sido actualizados a la fecha.

### **2.1.2. Entrevistas a *trainers***

Con el fin de obtener la información sobre los contenidos que se imparten y verificar que la metodología heredada de TCS se estuviera aplicando, se realizaron entrevistas a los diferentes *trainers* que imparten los cursos del área de desarrollo de software. De esta manera se logró determinar qué tipo de contenidos era el que recibían los estudiantes de estos cursos, de igual manera se determinó qué conceptos y temas no estaban siendo aplicados, cuáles habían sido renovados por los *trainers* y cuáles aún eran funcionales.

### **2.1.3. Comparativa de cursos**

Después de realizar las entrevistas a los *trainers*, se comparó la información obtenida de los mismos, con la impartida por TCS. Se hizo una investigación de campo a partir del esquema general y los objetivos de cada curso, buscando temas de actualidad que aplicarán, y de ser necesario cambiar algún tema. En algunos casos, el contenido del curso no sufrió cambio, ya que el que se imparte es aún eficiente y aplicable.

### **2.1.4. Elaboración de programa**

Debido a que no existía ningún documento que permitiera ver de manera resumida y concisa la información de cada curso, se elaboró uno por cada curso, el cual contenía desglosado el contenido del mismo, así como algunos datos importantes, tales como: el sistema de evaluación, el objetivo del curso, su alcance y sus límites, entre otros.

Para brindar mejor servicio tanto a las personas interesadas en recibir los cursos, como para los que lo imparten, cada documento está dividido en dos:

- Programa general: que estará disponible para todo público en donde muestra, de manera resumida la información de cada curso.
- Programa específico: contiene la misma información recopilada en el programa general, además de un desglose, sesión por sesión, de los contenidos impartidos en el curso.

## 2.2. Cursos actualizados

Debido a la importancia de estos cursos en relación a las materias y laboratorios impartidos en la Escuela de Sistemas, solamente se seleccionó el área de Ingeniería de software que comprenden los siguientes cursos separados en tres áreas:

- Ingeniería de software
  - ISE
  - SQM
  - SA
  
- Desarrollo de software
  - Técnicas básicas de programación
  - ASP
  - OOPs C++
  - IPC 1
  - IPC 2
  
- Programación en Java
  - Java básico
  - Java intermedio
  - Java avanzado

### **2.2.1. Área de Ingeniería de software**

Tiene como principal objetivo que el estudiante aprenda la aplicación de un enfoque sistemático, disciplinado y cuantificable en el desarrollo de software. Comprende las siguientes áreas.

#### **2.2.1.1. Introduction to software engineering –ISE-**

Este curso busca familiarizar a los estudiantes con los fundamentos de la Ingeniería de software y darles la capacidad de aplicar el análisis de sistemas y las técnicas de diseño para desarrollo de sistemas de software.

Entre sus objetivos se encuentran:

- Aprender los fundamentos de la ingeniería de software
- Introducir conceptos y técnicas de modelado de sistemas en la ingeniería de software.

Tiene una duración de 20 horas, siendo 10 clases de dos horas cada una.

#### **2.2.1.2. Software Quality Management -SQM-**

Este curso busca que el estudiante comprenda los conceptos de calidad y la importancia de la misma en el desarrollo de software, dándole a conocer las distintas herramientas o normas existentes para llevar el control y asegurar la calidad en todo el ciclo de vida de software.



Entre sus objetivos se encuentran:

- Aprender los conceptos de la gestión de proyectos
- Comprender los principios y técnicas de testeo
- Conocer el concepto de calidad y su aplicación

Este curso tiene una duración de 10 horas, comprendidas en 5 sesiones de 2 horas cada una.

### **2.2.1.3. Solution architecture –SA-**

Este curso busca que los alumnos puedan analizar los requerimientos del negocio en un escenario determinado y definir la solución técnica con las distintas arquitecturas para optimizar los resultados empresariales mediante el desarrollo de herramientas de Microsoft.

Entre sus objetivos se encuentran:

- Aprender técnicas de toma y análisis de requerimientos
- Conocer las herramientas de Microsoft para desarrollo
- Aprender a aplicar modelos de datos

Este curso tiene una duración de 16 horas, comprendidas en 8 sesiones de 2 horas cada una.

## **2.2.2. Desarrollo de software**

Tiene como objetivo principal orientar al estudiante en las diferentes técnicas, metodologías y lenguajes de programación. Comprende los siguientes cursos.

### **2.2.2.1. Técnicas básicas de programación**

Este curso busca familiarizar a los estudiantes con las técnicas necesarias para la construcción lógica de herramientas de programación.

Entre sus objetivos se encuentran:

- Familiarizar al estudiante con el enfoque de resolución de problemas utilizando la programación convencional.
- Enseñar al estudiante la relación entre datos, información y programas.
- Enseñar al estudiante diferentes herramientas para representar la lógica de la programación.
- Conocer los diferentes ciclos, instrucciones y estructura de la programación convencional.

Este curso tiene una duración de 12 horas, comprendidas en 6 sesiones de 2 horas cada una.

### **2.2.2.2. Active Server Page -ASP-**

Este curso busca que el estudiante comprenda el significado y relevancia de encriptación de lado del servidor, asimismo, que el estudiante tenga conocimientos de la herramienta Microsoft Transaction Server y de la tecnología Microsoft .Net

Entre sus objetivos se encuentran.

- Aprender a utilizar Microsoft .Net
- Conectar .Net con SQL Server
- Aprender a realizar aplicaciones Asp.Net

Este curso tiene una duración de 22 horas, comprendidas en 11 sesiones de 2 horas cada una.

### **2.2.2.3. OoP c++**

Este curso busca que el estudiante tenga una comprensión amplia de la programación orientada a objetos con C++.

Entre sus objetivos se encuentran:

- Aprender los principios de la programación orientada a objetos
- Aprender sobre expresiones y estructuras de control
- Conocer las funciones en C++
- Entender qué son clases y objetos

Este curso tiene una duración de 20 horas, comprendidas en 10 sesiones de 2 horas cada una.

#### **2.2.2.4. Introducción a la programación de computadoras 1-IPC 1-**

Que el estudiante salga con conceptos básicos de la programación orientada a objetos, los cuales sepa aplicar dentro del lenguaje de programación Java, además de tener los conceptos básicos de la programación para aprender cualquier lenguaje de programación.

Entre sus objetivos se encuentran:

- Familiarizar al estudiante con programación orientada a objetos
- Introducir al estudiante en la programación en Java
- Proveer al estudiante de conceptos básicos en la programación

Este curso tiene una duración de 5 meses, equivalente a la duración del laboratorio de Introducción a la Programación de la Escuela de Ciencias y Sistemas de la Universidad de San Carlos de Guatemala.

#### **2.2.2.5. Introducción a la programación de computadoras 2 -IPC 2-**

El curso busca familiarizar al estudiante con conceptos de sistemas de gestión de bases de datos. Así también, le enseña sobre el diseño y creación de modelos de bases de datos relacionales normalizadas y bien estructuradas. Por último, se muestra la forma de aplicar este conocimiento con el lenguaje de programación C#.

Entre sus objetivos se encuentran:

- Proveer al estudiante conocimientos sobre los sistemas de gestión de bases de datos.
- Enseñar a crear y diseñar modelos de bases de datos relacionales
- Aprender sobre sistemas de bases de datos normalizadas

Este curso tiene una duración de 5 meses, equivalente a la duración del laboratorio de Introducción a la Programación de la Escuela de Ciencias y Sistemas de la Universidad de San Carlos de Guatemala.

### **2.2.3. Programación en Java**

Enseña al estudiante conceptos y usos de los diferentes métodos del lenguaje de programación Java. Comprende los siguientes cursos:

#### **2.2.3.1. Java básico**

Este curso trata sobre los conceptos básicos de la programación orientada a objetos, que pueda aplicar dentro del lenguaje de programación Java.

Entre sus objetivos se encuentran:

- Familiarizar al estudiante con programación orientada a objetos
- Introducir al estudiante en la programación en Java
- Proveer al estudiante de conceptos básicos en la programación

Este curso tiene una duración de 20 horas, equivalente a 5 sesiones teóricas y 5 sesiones prácticas de 2 horas cada una.

### **2.2.3.2. Java intermedio**

Este curso busca familiarizar a los estudiantes con la programación orientada a objetos y las características básicas de Java para el desarrollo de aplicaciones de negocios en la plataforma Java 2.

Entre sus objetivos se encuentran:

- Entender qué es la programación orientada a objetos
- Aprender las tecnologías Java beans, servlets, JDBC y Swing

Este curso tiene una duración de 32 horas, equivalente a 8 sesiones teóricas y 8 sesiones prácticas de 2 horas cada una.

### **2.2.3.3. Java Avanzado**

Este curso busca orientar al estudiante en un nivel avanzado de programación en el lenguaje Java, abarcando temas sobre *RDBMS*, programación *web*, programación con *servlets*, entre otros.

Entre sus objetivos se encuentran:

- Aprender sobre el desarrollo de aplicaciones en la plataforma *J2EE*
- Conocer el concepto de *servlet*
- Conocer el concepto de Administración *Web*

Este curso tiene una duración de 34 horas, equivalentes a 8 sesiones teóricas y 9 sesiones prácticas de 2 horas cada uno.

### **2.3. Plantilla general de curso**

A manera de brindar información rápida y concisa de cada curso, se implementó el desarrollo de un documento que contuviera la información básica de cada uno. Para establecer un estándar entre los mismos, se creó una plantilla para la creación de futuros cursos.

Esta plantilla contiene las siguientes partes:

- Información general
- Objetivos
- Alcances
- Metodología de enseñanza
- Sistema de evaluación
- Consideraciones varias
- Bibliografía recomendada
- Contenido

La sección de información general incluye una subsección que contiene los siguientes campos:

- Área: indica a la que pertenece el curso.
- Duración: indica el tiempo en horas de cada curso.
- Prerrequisito: indica qué curso se necesita antes de tomar el actual.
- posrequisito: indica qué curso va después del curso actual.
- Vigencia: indica la vigencia del documento, definiendo el semestre y el año en curso.

La plantilla fue diseñada con el fin de ser intuitiva al momento de su aplicación para que de forma fácil y rápida pueda ser aplicada a las áreas no documentadas de ITCoE.





### 3. FASE ENSEÑANZA-APRENDIZAJE

Debido a la naturaleza del proyecto, no se llevó a cabo capacitación con cada uno de los *trainers* que imparten los cursos en el ITCoE. A partir de la plantilla general, se empezó a aplicar la misma en cursos existentes en otras áreas de ITCoE.

Cursos como: Tecnologías *Web 2.0*, entre otros, adoptó la plantilla general para iniciar a impartirse.

Adicional a eso, como parte del proyecto, se creó una base documental de cada curso, la cual incluye los siguientes ítems:

- Programa corto del curso
- Programa extendido

Esta base documental estará disponible tanto en versión digital como física en las oficinas de ITCoE para el acceso tanto de *trainers*, a través de subversion, como estudiantes interesados en recibir los cursos impartidos, a través de copias impresas.

De igual manera, según el principio de difusión de la información, esta misma base documental se encontrará disponible en la página *web* de ITCoE. A continuación se muestra un diseño propuesto para la presentación de los cursos.

Figura 1. **Página principal de cursos**



Fuente: elaboración propia.

La página principal daría fácil acceso a las tres áreas de cursos, como indica en la figura 1.

Figura 2. **Página de área específica**



Fuente: elaboración propia.

Al navegar por la página e ingresar a una sección en específico, se muestra el listado de cursos pertenecientes a la sección seleccionada, así como su duración.

Figura 3. Información de curso



The screenshot shows a web page with a dark red header. On the left, there is a vertical navigation menu with blue buttons labeled 'Página principal', 'Desarrollo de Software', 'ASP', 'IPC 1', 'IPC 2', and 'C++'. The main content area has the title 'Técnicas básicas de Programación' in a large, bold, italicized font. Below the title, there is a 'Descripcion' section with the text: 'Este curso busca familiarizar a los estudiantes con las técnicas necesarias para la construcción lógica de herramientas de programación.' Underneath that is an 'Objetivos' section with a bulleted list of four points: 'Familiarizar al estudiante con el enfoque de resolución de problemas utilizando la programación convencional.', 'Enseñar al estudiante la relación entre datos, información y programas.', 'Enseñar al estudiante diferentes herramientas para representar la lógica de la programación.', and 'Conocer los diferentes ciclos, instrucciones y estructura de la programación convencional.'

Fuente: elaboración propia.

Al ingresar a la información del curso, el menú del lado izquierdo cambiaría, mostrando el listado de cursos. Dentro del contenido de la página, muestra la descripción del curso y los objetivos del mismo, así también, tiene la opción de descargar el contenido completo del curso.



## CONCLUSIONES

1. A partir del contenido existente se revisó y actualizó el contenido de 11 cursos impartidos en el ITCoE.
2. Como parte del trabajo de EPS, se realizó un programa general que permitirá estandarizar la información de cada curso impartido, de esta manera se puede llevar mejor control sobre la calidad de información que se proporciona al público.
3. Con base en los requerimientos tomados, se creó un plantilla general del programa curricular, con la cual, futuros cursos a impartirse pueden acoplarse al sistema de estandarización de cursos sin mayor problema.
4. Definición de un sistema de evaluación, con la cual los encargados de impartir las clases puedan planificar las actividades a realizar con el fin de cubrir la metodología de evaluación propuesta.
5. Con los programas de cursos a disposición de los estudiantes y personal docente de ITCoE se logró reforzar la metodología de enseñanza heredada de TCS.
6. Se proporcionó al departamento de ITCoE una copia tanto digital como física, con la cual podrán tener a disposición la información de forma fácil y tener un respaldo físico de la misma, que formará parte del archivo general de ITCoE.



## RECOMENDACIONES

1. Continua revisión del contenido de los cursos que imparte con el fin de mantener actualizada la información de los mismos. De esta manera brindar educación de alto nivel y cumplir con los acuerdos establecidos entre la universidad y TCS.
2. Incluir paulatinamente los cursos de otras áreas al estándar de documentación realizada en el presente trabajo, ya que de esta manera se crea consistencia en la información que se ofrece a los estudiantes interesados en cursar las materias y se implementa control sobre los cursos impartidos.
3. Crear un sistema de control y actualización de contenidos, mediante el cual se pueda tener control sobre la calidad de educación que se imparte, para poder enfocar y utilizar de mejor manera el recurso humano en el ITCoE.





## BIBLIOGRAFÍA

1. *Fiusac –India-Guatemala IT Education Centre of Excellence* [en línea]  
<<http://itcoe.ingenieria-usac.edu.gt>>. [Consulta: 9 de abril de 2011].



## **APÉNDICES**



Apéndice 1. **Plantilla general**

**Programa general: <Nombre del curso>**

<b>Información general</b>			
Área	<Área a la que pertenece el curso>		
Duración	<Duración en horas, entre paréntesis número de clases>		
Prerrequisito	<Nombre del curso prerrequisito, si no tiene “Ninguno”>		
Posrequisito	<Nombre del curso posrequisito, si no tiene “Ninguno”>		
Vigencia	<Semestre>		
<b>Descripción del curso:</b>			
	<Agregar una descripción corta del curso>		
<b>Objetivos:</b>			
	➤ <Indicar los objetivos generales del curso>		
<b>Al finalizar el curso el estudiante estará capacitado para:</b>			
	➤ <Indicar de manera específica los alcances del curso>		
<b>Metodología de enseñanza:</b>			
	➤ <Indicar la metodología a utilizar>		
<b>Evaluaciones: &lt;Si aplica, caso contrario se omite el renglón&gt;</b>			
	Asistencia:	<n puntos>	
	Tareas/hojas de trabajo:	<n Puntos>	
	<N> Exámenes cortos:	<punteo individual>	<n puntos>
	Examen final:	<n puntos>	
	Total:	<b>100 Puntos</b>	
<b>Consideraciones:</b>			
	➤ <Indicar las consideraciones que se deban tomar en cuenta para poder optar a la nota final>		

Continuación del apéndice 1.

<b>Bibliografía recomendada</b>	
	➤ <Si la hubiera, indicar bibliografía recomendada>
<b>Contenido:</b>	
<ul style="list-style-type: none"> <li>• &lt;En términos generales, indicar el listado de temas que tratarán en el curso &gt;</li> </ul>	<ul style="list-style-type: none"> <li>• &lt;Sin detalle, solamente los títulos o nombres de los temas a impartir&gt;</li> </ul>

**Programa específico: <Nombre del curso>**

<b>Información general</b>	
Área	<Área a la que pertenece el curso>
Duración	<Duración en horas, entre paréntesis número de clases>
Prerrequisito	<Nombre del curso prerrequisito, si no tiene “ninguno”>
Posrequisito	<Nombre del curso posrequisito, si no tiene “ninguno”>
Vigencia	Primer semestre 2012
<b>Descripción del curso:</b>	
	<Agregar una descripción corta del curso>
<b>Objetivos:</b>	
	➤ <Indicar los objetivos generales del curso>
<b>Al finalizar el curso el estudiante estará capacitado para:</b>	
	➤ <Indicar de manera específica los alcances del curso>
<b>Metodología de enseñanza:</b>	
	➤ <Indicar la metodología a utilizar>
<b>Evaluaciones: &lt;Si aplica, caso contrario se elimina el renglón&gt;</b>	

Continuación del apéndice 1.

	Asistencia:	<n puntos>	
	Tareas/hojas de trabajo:	<n Puntos>	
	<N> Exámenes cortos:	<punteo individual>	<n puntos>
	Examen final:	<n puntos>	
	Total:	<b>100 Puntos</b>	
<b>Consideraciones:</b>			
	➤ <Indicar las consideraciones que se deban tomar en cuenta para poder optar a la nota final>		
<b>Bibliografía recomendada</b>			
	➤ <Si la hubiera indicar bibliografía recomendada>		
<b>Contenido:</b>			
	<ul style="list-style-type: none"> <li>• &lt;En términos generales, indicar el listado de temas que tratarán en el curso &gt;</li> </ul>	<ul style="list-style-type: none"> <li>• &lt;Sin detalle, solamente los títulos o nombres de los temas a impartir&gt;</li> </ul>	
<b>Programación:</b>			
	<ul style="list-style-type: none"> <li>• &lt;Describir por clases, el contenido de cada una, indicando los temas que tratarán en cada clase. &gt;</li> <li>• &lt;De igual manera, indicar en qué clase se realizarán exámenes cortos y examen final. &gt;</li> </ul>		

Fuente: elaboración propia.



Apéndice 2. **Programa de curso: Técnicas básicas de programación**

<b>Información general</b>	
Área	Técnicas básicas de programación.
Duración	12 horas (6 clases de 2 horas cada clase)
Prerrequisito	Ninguno
Posrequisito	IPC 1
Vigencia	Primer semestre 2012
<b>Descripción del curso:</b>	
	Este curso busca familiarizar a los estudiantes con las técnicas necesarias para la construcción lógica de herramientas de programación.
<b>Objetivos:</b>	
	<ul style="list-style-type: none"> <li>➤ Familiarizar al estudiante con el enfoque de resolución de problemas utilizando la programación convencional.</li> <li>➤ Enseñar al estudiante la relación entre datos, información y programas.</li> <li>➤ Enseñar al estudiante diferentes herramientas para representar la lógica de la programación.</li> <li>➤ Conocer los diferentes ciclos, instrucciones y estructura de la programación convencional.</li> </ul>
<b>Al finalizar el curso el estudiante estará capacitado para:</b>	
	<ul style="list-style-type: none"> <li>➤ Conocer la diferencia entre programa, información y datos.</li> <li>➤ Indicar herramientas de programación básica.</li> <li>➤ Realizar ciclos e iteraciones en un programa.</li> <li>➤ Saber la importancia de la programación modular.</li> <li>➤ Crear arreglos y saber manipularlos.</li> <li>➤ Conocer el concepto de archivos y como manipularlos.</li> <li>➤ Conocer y crear reportes.</li> </ul>

Continuación del apéndice 2.

<b>Metodología de enseñanza:</b>									
	<ul style="list-style-type: none"> <li>➤ Clases presenciales.</li> <li>➤ Presentaciones en Power Point.</li> </ul>								
<b>Evaluaciones:</b>									
	<table border="1"> <tr> <td>Asistencia:</td> <td>15 puntos</td> </tr> <tr> <td>1 Examen corto:</td> <td>25 puntos</td> </tr> <tr> <td>Examen final:</td> <td>60 puntos</td> </tr> <tr> <td>Total:</td> <td><b>100 Puntos</b></td> </tr> </table>	Asistencia:	15 puntos	1 Examen corto:	25 puntos	Examen final:	60 puntos	Total:	<b>100 Puntos</b>
Asistencia:	15 puntos								
1 Examen corto:	25 puntos								
Examen final:	60 puntos								
Total:	<b>100 Puntos</b>								
<b>Consideraciones:</b>									
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia.</li> <li>➤ La nota de aprobación es de 61 puntos.</li> </ul>								
<b>Bibliografía recomendada</b>									
	<ul style="list-style-type: none"> <li>➤ <i>How to solve it by computer</i> <ul style="list-style-type: none"> <li>○ R.G. Dromey</li> </ul> </li> </ul>								
<b>Contenido:</b>									
<ul style="list-style-type: none"> <li>• Desarrollo de software</li> <li>• Diseño de sistemas</li> <li>• Comprobación de sistemas</li> <li>• Datos e información.</li> <li>• Estructuras de control</li> <li>• Iteraciones</li> <li>• Estructura de programas</li> <li>• Paso de parámetros</li> <li>• Herramientas para ilustración de la programación</li> </ul>	<ul style="list-style-type: none"> <li>• Arreglos</li> <li>• Creación de arreglos</li> <li>• Extracción de datos de arreglos</li> <li>• Clasificación de arreglos</li> <li>• Búsqueda de arreglos</li> <li>• Introducción a archivos</li> <li>• Leer, escribir y modificar archivos</li> <li>• Reportes</li> </ul>								

Continuación del apéndice 2.

## **Programación:**

- **Clase 1: Introducción a la programación**
  - Desarrollo de software
  - Diseño de sistemas
  - Comprobación de sistemas
  - Datos e información
    - Procesamiento de datos
    - Procesamiento de datos usando computadoras
    - Características de un buen programa
    - Declaraciones en la ejecución de un programa
    - Organización de un programa
    - Tipos de datos
  - Herramientas para la ilustración de un programa
    - Algoritmos
    - Diagramas de flujo
    - Pseudo código
- **Clase 2: Introducción a las estructuras de control**
  - Selección de estructuras de control
    - Selección de una vía
    - Selección de dos vías
    - Selección multi-vias
    - *If—End if*
    - *If— else— endif*
    - *Do—case—endcase*
  - Iteraciones
    - *For — next*
    - *Do— While*

Continuación del apéndice 2.

- *Repeat— until*
- **Clase 3: Introducción a la programación modular**
  - Estructura de un programa
    - Ventajas de un enfoque estructurado
    - Subrutinas utilizando diagramas de flujo
    - Subrutinas utilizando pseudo código
  - Paso de parámetros
- **Examen parcial**
- **Clase 4: Manejo de arreglos**
  - Arreglos
    - Definición
    - Tipos de arreglos
  - Creación de matrices
    - Declaración de matrices
    - Definición de matrices
    - Inicialización de matrices
    - Referencia de elementos en un arreglo
    - Almacenamiento de datos en arreglos
  - Extracción de datos de un arreglo
  - Clasificación de matrices
  - Búsqueda en arreglos
    - Tipos de búsqueda
- **Clase 5: Introducción al manejo de archivos**
  - Introducción a archivos
    - ¿Qué es un archivo?
    - Tipos de archivos
  - Escritura de archivos

Continuación del apéndice 2.



Fuente: elaboración propia.

### Apéndice 3. Programa de curso: ASP

<b>Información general</b>	
Área	Desarrollo de software
Duración	22 horas (11 clases de dos horas cada clase)
Prerrequisito	Ninguno
Posrequisito	Ninguno
Vigencia	Primer semestre 2012
<b>Descripción del curso:</b>	
	Este curso busca que el estudiante comprenda el significado y relevancia de encriptación de lado del servidor, asimismo, busca que el estudiante tenga conocimientos de la herramienta Microsoft Transaction Server y de la tecnología Microsoft .Net

Continuación del apéndice 3.

<b>Objetivos:</b>			
	<ul style="list-style-type: none"> <li>➤ Aprender a utilizar Microsoft .Net</li> <li>➤ Conectar .Net con SQL Server</li> <li>➤ Aprender a realizar aplicaciones Asp.Net</li> </ul>		
<b>Al finalizar el curso el estudiante estará capacitado para:</b>			
	<ul style="list-style-type: none"> <li>➤ Entender el significado y relevancia de <i>server-side encrypting</i></li> <li>➤ Desarrollar aplicaciones con conexión a base de datos</li> <li>➤ Conocer las diferentes sentencias cíclicas de Visual Basic</li> <li>➤ Explicar el concepto de programación orientada a objetos</li> <li>➤ Realizar aplicaciones desarrolladas en Asp.Net</li> </ul>		
<b>Metodología de enseñanza:</b>			
	<ul style="list-style-type: none"> <li>➤ Clases presenciales</li> <li>➤ Presentaciones en Power Point</li> </ul>		
<b>Evaluaciones:</b>			
	Asistencia:	10 puntos	
	Tareas/hojas de trabajo:	20 puntos	
	2 Exámenes cortos:	15 puntos c/u	30 puntos
	Examen final:	40 puntos	
	Total:	<b>100 Puntos</b>	
<b>Consideraciones:</b>			
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia.</li> <li>➤ La nota de aprobación es de 61 puntos.</li> </ul>		

Continuación del apéndice 3.

<b>Contenido:</b>	
<ul style="list-style-type: none"> <li>• ¿Qué es .Net?</li> <li>• Componentes del <i>Framework</i> de .Net</li> <li>• Sentencias de control en Visual Basic</li> <li>• Orientación a objetos</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Namespaces y properties</i> en .Net</li> <li>• Componentes gráficos de .Net</li> <li>• Conexión de .Net con SQL server</li> <li>• Aplicaciones <i>Web</i></li> <li>• <i>Linq</i></li> <li>• <i>Windows Form</i></li> </ul>
<b>Programación:</b>	
<ul style="list-style-type: none"> <li>• <b>Clase 1:</b> <ul style="list-style-type: none"> <li>○ Introducción al <i>Framework</i> de .Net           <ul style="list-style-type: none"> <li>▪ ¿Qué es el <i>Framework</i> de .Net?</li> <li>▪ Componentes del <i>Framework</i> <ul style="list-style-type: none"> <li>• Conjunto de lenguajes</li> <li>• Biblioteca de clases base</li> <li>• Entorno común de ejecución</li> </ul> </li> </ul> </li> </ul> </li> <li>• <b>Clase 2:</b> <ul style="list-style-type: none"> <li>○ Sentencias de control en sintaxis Visual Basic           <ul style="list-style-type: none"> <li>▪ Sentencias condicionales</li> <li>▪ Sentencias cíclicas</li> <li>▪ Operadores asignación</li> <li>▪ Operadores relacionales</li> <li>▪ Operadores lógicos</li> </ul> </li> <li>○ Características específicas del lenguaje</li> </ul> </li> <li>• <b>Clase 3:</b> <ul style="list-style-type: none"> <li>○ Programación orientada a objetos           <ul style="list-style-type: none"> <li>▪ Clase</li> </ul> </li> </ul> </li> </ul>	

Continuación del apéndice 3.

- Objeto
  - Declaración de atributos
  - Métodos
  - Constructores
- **Clase 4:**
  - *Namespaces* y propiedades
  - Que es un *namespace*
  - Propiedades
    - *Read only*
    - *Write only*
    - *Write y read*
- **Primer examen corto**
- **Clase 5:**
  - Introducción a los componentes interfaz gráfica.
    - Botones
    - Etiquetas
    - *Textbox*
    - *Label*
    - *Forms*
    - Listas
- **Clase 6:**
  - *Windows Forms Applications*
  - Características
  - Propiedades
  - Ejemplos prácticos
- **Clase 7**
  - Conexión a base de datos



Continuación del apéndice 3.

- Instancia a base de datos desde .Net
- *String Connection*
- Conexión *ODBC*
- Consultas
- Ejecución de comandos
- Paso de parámetros y *store procedures*
- **Segundo examen corto**
- **Clase 8**
  - *Web applications*
    - Variables de sesión
    - Objeto *request*
    - Objeto *response*
    - *PostBack*
    - Ejemplos de clase
- **Clase 9**
  - ¿Qué es *Linq*?
  - Variable *Var*
  - Conexión a base de datos
  - *Linq for Sql*
  - Ejemplos prácticos
- **Clase 10**
  - *C# vs VB*
  - Ventajas y características
- **Clase 11 Examen final**

Fuente: elaboración propia.

Apéndice 4. **Programa de curso: IPC 1**

<b>Información general</b>	
Área	Desarrollo de software
Duración	5 meses (equivalente al curso de IPC1)
Prerrequisito	Ninguno
Posrequisito	IPC 2
Vigencia	Primer semestre 2012
<b>Descripción del curso:</b>	
	Que el estudiante obtenga conceptos básicos de la programación orientada a objetos para poder aplicarlos dentro del lenguaje de programación Java, además de tener los conceptos básicos de la programación, para aprender cualquier lenguaje de programación.
<b>Objetivos:</b>	
	<ul style="list-style-type: none"> <li>➤ Familiarizar al estudiante con programación orientada a objetos.</li> <li>➤ Introducir al estudiante en la programación en Java.</li> <li>➤ Proveer al estudiante de conceptos básicos en la programación.</li> </ul>
<b>Al finalizar el curso el estudiante estará capacitado para:</b>	
	<ul style="list-style-type: none"> <li>➤ Indicar conceptos de programación orientada a objetos.</li> <li>➤ Tener nociones de las instrucciones de control y ciclos en programación.</li> <li>➤ Manejar los conceptos básicos de programación en Java.</li> <li>➤ Realizar aplicaciones en Java.</li> <li>➤ Manejar excepciones y carga de archivos en Java.</li> </ul>
<b>Metodología de enseñanza:</b>	
	<ul style="list-style-type: none"> <li>➤ Clases presenciales</li> <li>➤ Presentaciones en Power Point</li> <li>➤ Clases prácticas.</li> </ul>

Continuación del apéndice 4.

<b>Evaluaciones:</b>			
	Tareas:		10 puntos
	3 Exámenes cortos:	5 puntos c/u	15 puntos
	2 Prácticas:	5 puntos c/u	10 Puntos
	Proyecto 1:		15 puntos
	Proyecto 2:		25 puntos
	Examen final		25 puntos
	Total:		<b>100 Puntos</b>
<b>Consideraciones:</b>			
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia.</li> <li>➤ La nota de aprobación es de 61 puntos.</li> <li>➤ Debe entregar ambas prácticas y proyectos para optar a examen final</li> </ul>		
<b>Bibliografía sugerida:</b>			
	<ul style="list-style-type: none"> <li>➤ Programación en Java 2               <ul style="list-style-type: none"> <li>○ Luis Joyanes Aguilar e Ignacio Zahonero Martínez                   <ul style="list-style-type: none"> <li>▪ McGraw-Hill</li> </ul> </li> </ul> </li> <li>➤ Java, Cómo Programar.               <ul style="list-style-type: none"> <li>○ Harvey M. Deitel, Paul J. Deitel. Quinta edición 2004                   <ul style="list-style-type: none"> <li>▪ Pearson, Prentice-Hall</li> </ul> </li> </ul> </li> </ul>		
<b>Contenido:</b>			
<ul style="list-style-type: none"> <li>• Definición en Java</li> <li>• Componentes del lenguaje</li> <li>• Sentencias de control</li> <li>• Estructura de datos</li> <li>• Conceptos de <i>OOP</i></li> </ul>		<ul style="list-style-type: none"> <li>• Herencia y polimorfismo</li> <li>• Modificadores y control de acceso</li> <li>• Clases y métodos abstractos</li> <li>• Excepciones y manejo de errores</li> <li>• Memoria dinámica</li> </ul>	

Continuación del apéndice 4.

<ul style="list-style-type: none"><li>• Clases, objetos</li><li>• Paquetes</li><li>• Constructores</li></ul>	<ul style="list-style-type: none"><li>• Lectura de archivos</li><li>• Recursividad</li><li>• <i>Api Swing</i></li></ul>
<b>Programación:</b>	
<ul style="list-style-type: none"><li>• <b>Sesión 1:</b><ul style="list-style-type: none"><li>○ Introducción<ul style="list-style-type: none"><li>▪ Historia de Java</li><li>▪ Introducción a aplicaciones Java</li><li>▪ Versiones y plataformas</li><li>▪ Características de Java</li></ul></li></ul></li><li>• <b>Sesión 2:</b><ul style="list-style-type: none"><li>○ Componentes del lenguaje<ul style="list-style-type: none"><li>▪ Operadores</li><li>▪ Tipos de datos</li><li>▪ Variables y constantes</li><li>▪ Sentencias y expresiones</li><li>▪ Comentarios</li><li>▪ Ámbitos de las variables</li></ul></li></ul></li><li>• <b>Sesión 3:</b><ul style="list-style-type: none"><li>○ Sentencias de control<ul style="list-style-type: none"><li>▪ <i>If</i></li><li>▪ <i>Switch</i></li><li>▪ <i>While</i></li><li>▪ <i>Do-while</i></li><li>▪ <i>For</i></li><li>▪ <i>Break</i></li><li>▪ <i>Continue</i></li></ul></li></ul></li></ul>	

Continuación del apéndice 4.

- Estructura de datos
  - Métodos
  - Arreglos
- **Sesión 4:**
  - *UML* y el paradigma orientado a objetos
    - Conceptos básicos
    - Notación *UML*
- **Sesión 5 y 6:**
  - Programación orientada a Objetos
    - Clase abstracta
    - Alcance de las clases
    - Control de acceso a los miembros
    - *This*
    - Constructores
    - Constructores sobrecargados
    - Métodos *set* y *get*
    - Recolección de basura
    - Composición
    - Miembros estáticos
    - Variables de instancia finales
    - Paquetes
      - Concepto
      - Creación de nuestros propios paquetes
      - Paquetes de Java
    - Reutilización de software
    - Abstracción de datos y encapsulamiento
- **Sesión 7:**

Continuación del apéndice 4.

- Herencia
  - Superclases y subclases
  - Miembros *protected*
  - Relación entre las superclases y las subclases
  - Jerarquía de herencia de tres niveles
  - Constructores y finalizadores en las subclases
- Polimorfismo
  - Relación entre los objetos en una jerarquía de herencia
  - Clases y métodos abstractos
  - Métodos y clases final
  - Clases anidadas

Clases de tipo de envoltura para los tipos primitivos

- **Sesión 8:**

- Modificadores y control de acceso
  - *Default*
  - *Public*
  - *Private*
  - *Protected*
  - *Static*
  - *Final*

- Interfaces

- **Sesión 9:**

- Excepciones y errores
  - Como manejar las excepciones
  - *Try, throw y catch*
  - *Throws*
  - *Finally*

Continuación del apéndice 4.

- Múltiples sentencias *catch*
- Sentencias *try* anidadas
- Creación de nuestras propias excepciones
- **Sesión 10:**
  - Archivos
    - Lectura y escritura de archivos de texto
    - Lectura y escritura de archivos secuenciales (binarios)
- **Sesión 11:**
  - Memoria dinámica
    - Listas
    - Listas doblemente enlazadas
    - Pilas
  - Árboles binarios
  - Recursividad
    - ¿Qué es recursividad?
    - Recursividad o iteración
    - Funciones recursivas y ejemplos
- **Sesión 12:**
  - *Swing*
    - *JLabel*
    - *JButton*
    - *JCheckBox* y *JRadioButton*
    - *JComboBox*
    - *JList*
    - *Layouts*
    - *JTextArea*
    - *JPanel*

Continuación del apéndice 4.

<ul style="list-style-type: none"> <li>▪ <i>JSlider</i></li> <li>▪ <i>JPopupMenu</i></li> <li>▪ <i>JDesktopPane</i> y <i>JInternalFrame</i></li> <li>○ <i>JtabbedPane</i></li> </ul>
--

Fuente: elaboración propia.

#### Apéndice 5. Programa de curso: IPC 2

<b>Información general</b>	
Área	Desarrollo de software
Duración	5 meses (Equivalente al curso de IPC2)
Prerrequisito	IPC 1
Posrequisito	Ninguno
Vigencia	Primer semestre 2012
<b>Descripción del curso:</b>	
	El curso busca familiarizar al estudiante con conceptos de sistemas de gestión de bases de datos, así también, enseñar al estudiante sobre el diseño y creación de modelos de bases de datos relacionales normalizados y bien estructurados, por último, mostrar la forma de aplicar este conocimiento con el lenguaje de programación C#.
<b>Objetivos:</b>	
	<ul style="list-style-type: none"> <li>➤ Proveer al estudiante del conocimiento sobre los sistemas de gestión de bases de datos.</li> <li>➤ Enseñar a crear y diseñar modelos de bases de datos relacionales.</li> </ul>



Continuación del apéndice 5.

	➤ Aprender sobre sistemas de bases de datos normalizados.		
<b>Al finalizar el curso el estudiante estará capacitado para:</b>			
	<ul style="list-style-type: none"> <li>➤ Diferenciar entre los sistemas de gestión de bases de datos</li> <li>➤ Diseñar y crear sistemas de bases de datos relacionales</li> <li>➤ Aplicar normalización</li> <li>➤ Realizar diagramas de entidad relación</li> <li>➤ Realizar conexiones de base de datos</li> <li>➤ Manejar conceptos de relación, tuplas y registros</li> </ul>		
<b>Metodología de enseñanza:</b>			
	<ul style="list-style-type: none"> <li>➤ Clases presenciales</li> <li>➤ Presentaciones en Power Point</li> <li>➤ Clases prácticas</li> </ul>		
<b>Evaluaciones:</b>			
	3 prácticas:	10, 10 y 20 puntos	40 puntos
	Tareas:		10 puntos
	2 Exámenes cortos:		5 puntos
	Proyecto:		30 puntos
	Examen final		15 puntos
	Total:		<b>100 Puntos</b>
<b>Consideraciones:</b>			
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia.</li> <li>➤ La nota de aprobación es de 61 puntos.</li> <li>➤ Debe entregar proyectos para optar a examen final.</li> </ul>		
<b>Contenido:</b>			

Continuación del apéndice 5.

<ul style="list-style-type: none"><li>• Introducción a las bases de datos</li><li>• Tipos de bases de datos</li><li>• Conceptos básicos de bases de datos</li><li>• Diagrama entidad relación</li></ul>	<ul style="list-style-type: none"><li>• Lenguaje SQL</li><li>• <i>DML</i></li><li>• Transacciones</li><li>• Lenguaje C#</li><li>• ASP.Net</li><li>• Normalización</li></ul>
<b>Programación:</b>	
<ul style="list-style-type: none"><li>• <b>Sesión 1:</b><ul style="list-style-type: none"><li>○ Introducción a bases de datos<ul style="list-style-type: none"><li>▪ Descripción</li><li>▪ Características</li><li>▪ Ventajas</li><li>▪ Desventajas</li></ul></li></ul></li><li>• <b>Sesión 2:</b><ul style="list-style-type: none"><li>○ Definición de <i>DBMS</i></li><li>○ Tipos de bases de datos</li></ul></li><li>• <b>Sesión 3:</b><ul style="list-style-type: none"><li>○ Conceptos básicos de bases de datos<ul style="list-style-type: none"><li>▪ Dato</li><li>▪ Campo</li><li>▪ Registro</li><li>▪ Archivo</li><li>▪ Tabla</li><li>▪ Tupla</li><li>▪ Llave primaria</li><li>▪ Llave foránea</li></ul></li></ul></li></ul>	

Continuación del apéndice 5.

- **Sesión 4:**
  - Tipos de relaciones entre tablas
    - Uno a uno
    - Uno a muchos
    - Muchos a muchos
- **Sesión 5:**
  - Diagrama entidad relación
    - Definición de diagrama
    - Conceptos generales del diagrama entidad relación
    - Ejemplos
    - Diversas notaciones del diagrama
- **Sesión 6 y 7:**
  - Normalización
    - Definición
    - Conceptos básicos
    - Formas normales
- **Sesión 8:**
  - Lenguaje SQL
    - Introducción
    - Definición
    - Características
    - Historia
    - Partes o categorías de las sentencias de SQL
- **Sesión 9:**
  - *DML*
    - *Insert*
    - *Delete*

Continuación del apéndice 5.

- *Update*
- *Select*
- **Sesión 10:**
  - *DDL*
    - *Create*
    - *Alter*
    - *Drop*
  - *DCL*
    - *Grant*
    - *Revoke*
- **Sesión 11:**
  - *TCL*
    - *Commit*
    - *Rollback*
    - *Begin transaction*
- **Sesión 12:**
  - Índices
  - Vistas
  - Tipos de datos en SQL
  - Operadores existentes
  - *Join* y tipos de *join*
  - Consultas
- **Sesión 13:**
  - Funciones agregadas
  - *Triggers*
  - Procedimientos almacenados

Continuación del apéndice 5.

- **Sesión 14:**
  - Subconsultas
  - Vistas
- **Sesión 15 y 16:**
  - Lenguaje C#
    - Historia
    - Conceptos básicos
    - Características
    - Operadores de asignación
    - Operadores relacionales
    - Operadores lógicos
    - Sentencias de control
    - Sentencias cíclicas
    - Métodos y atributos
    - Manejo de clases
- **Sesión 17 y 18:**
  - *Web Forms* con ASP .NET
    - Componentes básicos
    - Formularios
    - Objetos principales
    - *Session*
    - *Request*
    - *Response*
    - Conexión a bases de datos
    - Inserciones
    - Reportes

Continuación del apéndice 5.

<ul style="list-style-type: none"> <li>▪ Actualizaciones</li> <li>▪ Eliminaciones</li> </ul>
--

Fuente: elaboración propia.

Apéndice 6. **Programa de curso: C++**

<b>Información general</b>	
Área	Desarrollo de software
Duración	20 horas (10 clases de dos horas cada clase)
Prerrequisito	Ninguno
Posrequisito	Ninguno
Vigencia	Primer semestre 2012
<b>Descripción del curso:</b>	
	Este curso busca que los estudiantes comprendan el concepto de programación orientada a objetos, usando el lenguaje de programación C++.
<b>Objetivos:</b>	
	<ul style="list-style-type: none"> <li>➤ Entender qué es programación orientada a objetos.</li> <li>➤ Manejar sentencias de control e instrucciones de programación en C++.</li> <li>➤ Conocer el concepto de funciones, arreglos de cadenas, clases y objetos en C++.</li> </ul>
<b>Al finalizar el curso el estudiante estará capacitado para:</b>	
	<ul style="list-style-type: none"> <li>➤ Conocer los principios de programación orientada a objetos.</li> <li>➤ Realizar programas en donde se apliquen sentencias de control.</li> </ul>

Continuación del apéndice 6.

	<ul style="list-style-type: none"> <li>➤ Crear funciones.</li> <li>➤ Realizar diferentes tipos de arreglos.</li> <li>➤ Diferenciar entre clases y objetos.</li> <li>➤ Sobre escribir operaciones.</li> </ul>										
<b>Metodología de enseñanza:</b>											
	<ul style="list-style-type: none"> <li>➤ Clases presenciales</li> <li>➤ Presentaciones en Power Point</li> </ul>										
<b>Evaluaciones:</b>											
	<table border="1"> <tr> <td>Asistencia:</td> <td>10 puntos</td> </tr> <tr> <td>Tareas/hojas de trabajo:</td> <td>20 puntos</td> </tr> <tr> <td>1 examen Corto:</td> <td>20 puntos</td> </tr> <tr> <td>Examen final:</td> <td>50 puntos</td> </tr> <tr> <td><b>Total:</b></td> <td><b>100 Puntos</b></td> </tr> </table>	Asistencia:	10 puntos	Tareas/hojas de trabajo:	20 puntos	1 examen Corto:	20 puntos	Examen final:	50 puntos	<b>Total:</b>	<b>100 Puntos</b>
Asistencia:	10 puntos										
Tareas/hojas de trabajo:	20 puntos										
1 examen Corto:	20 puntos										
Examen final:	50 puntos										
<b>Total:</b>	<b>100 Puntos</b>										
<b>Consideraciones:</b>											
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia.</li> <li>➤ La nota de aprobación es de 61 puntos.</li> </ul>										
<b>Bibliografía sugerida:</b>											
	<ul style="list-style-type: none"> <li>➤ <i>Object – oriented programming with C++</i> <ul style="list-style-type: none"> <li>○ Robert Lafore</li> </ul> </li> <li>➤ <i>Object – Oriented Programming with C++</i> <ul style="list-style-type: none"> <li>○ E. Balagurusamy</li> </ul> </li> <li>➤ <i>Learning with C++</i> <ul style="list-style-type: none"> <li>○ Yeshwant Kanitkar</li> </ul> </li> </ul>										
<b>Contenido:</b>											

Continuación del apéndice 6.

<ul style="list-style-type: none"><li>• Principios de programación orientada a objetos.</li><li>• Ventajas de la orientación a objetos.</li><li>• Introducción a C++</li><li>• Estructuras de control y expresiones.</li><li>• Casteo de variables</li><li>• Funciones en C++</li></ul>	<ul style="list-style-type: none"><li>• Introducción a clases y objetos</li><li>• destructores</li><li>• Sobreescritura de operaciones</li><li>• Herencia en C++</li><li>• Conceptos avanzados en C++</li><li>• Lectura y escritura de archivos</li><li>• Operaciones con archivos</li><li>• Plantillas y excepciones</li><li>• Arreglos y cadenas</li></ul>
<b>Programación:</b>	
<ul style="list-style-type: none"><li>• <b>Clase 1:</b><ul style="list-style-type: none"><li>○ La necesidad de la programación orientada a objetos.<ul style="list-style-type: none"><li>▪ Repaso sobre los lenguajes procedurales.</li><li>▪ Paradigma de la programación orientada a objetos.</li></ul></li><li>○ Concepto de objetos<ul style="list-style-type: none"><li>▪ Objetos</li><li>▪ Clases</li><li>▪ Abstracción de datos</li><li>▪ Encapsulamiento</li><li>▪ Herencia</li><li>▪ Polimorfismo</li></ul></li><li>○ Introducción a C++<ul style="list-style-type: none"><li>▪ Plantillas de C++</li><li>▪ Estructura de C++</li><li>▪ Creación de código fuente</li><li>▪ Compilación y enlace</li></ul></li></ul></li></ul>	



Continuación del apéndice 6.

- **Clase 2:**

- Introducción a identificadores y contantes en C++
  - *KeyWords*
  - *Identifiers*
  - *Constants*
- Tipos de datos
  - Tipos de datos construidos
  - Tipos de datos definidos por usuarios
  - Tipos de datos derivados
- Constantes simbólicas
- Tipos de compatibilidad
- Variables y su alcance
  - Inicialización de variables
  - Variables de referencia
- Conversiones implícitas y conversiones de tipo
- Operadores y expresiones
- Estructuras de control
  - Estructuras de decisión
  - Constructores de ciclos

- **Clase 3:**

- Introducción a funciones
- Paso de parámetros
  - Constantes
  - Variables
  - Referencia
  - Parámetros por default
  - Retorno de variable desde una función

Continuación del apéndice 6.

- Funciones sobre escritas
- **Clase 4:**
  - Fundamentos de arreglos
    - Definición de arreglo
    - Inicializando arreglos
    - Arreglos multidimensionales
    - Paso de arreglos en funciones
  - Usos de los arreglos
    - Arreglos como estructura
    - Arreglos como clases miembro
    - Arreglo como objeto
  - Introducción a cadenas
  - Arreglos como cadenas
- **Clase 5:**
  - Introducción a clases
    - Declaración de clases
    - Definición de funciones miembro
    - Funciones privadas
    - Clases locales
  - Objetos
    - Colocación de objetos en memoria
    - Objeto como función
    - Retorno de objetos
  - Miembros estáticos
  - Constructores
    - Introducción a constructores
    - Múltiples constructores en una clase

Continuación del apéndice 6.

- Constructores dinámicos.
- Destruyores
- **Primer examen corto**
- **Clase 6**
  - Sobrecarga de operadores unitarios
  - Sobrecarga de operadores binarios
    - Operadores aritméticos
    - Concatenar cadenas
    - Operadores de comparación
    - Operadores aritméticos de operación
  - Conversión de datos
  - Consejos para sobrecarga de operadores y conversión de datos
    - Usar similares significados
    - Usar similar sintaxis
- **Clase 7**
  - Herencia en C++
    - Razones para aplicar la herencia
    - Clases padre
    - Clases derivadas
  - Tipos de Herencia
    - Herencia simple
    - Especificadores de acceso
    - Herencia multinivel
    - Herencia múltiple
    - Herencia jerárquica
    - Herencia híbrida
  - Implementación de herencia

Continuación del apéndice 6.

- Abstracción de clases
- Clases anidadas
- **Clase 8**
  - Introducción a punteros
  - Sobreescritura de funciones
    - Enlace anticipado
    - Enlace demorado
    - Funciones virtuales miembro
    - Clases virtuales base
  - *Friend functions*
  - Pre procesamiento de C++
- **Clase 9**
  - Introducción a operaciones de entrada y salida
    - Clases de flujos
    - Ventajas sobre flujos
    - Herencia en clases de flujo
  - Errores de flujos
  - Entrada y salida de archivos
  - Trabajando con archivos
- **Clase 10**
  - Plantillas de funciones
  - Clases de plantillas
  - Librería estándar de plantillas
  - Manejo de excepciones
    - Necesidad de una excepción
    - Sintaxis de excepciones
    - Excepciones múltiples

Continuación del apéndice 6.

<ul style="list-style-type: none"> <li>▪ Excepciones con parámetros.</li> <li>• <b>Examen final</b></li> </ul>
--

Fuente: elaboración propia.

Apéndice 7. **Programa de curso: ISE**

<b>Información general</b>	
Área	<i>Software Engineering and OOAD</i>
Duración	20 horas (10 clases de dos horas cada clase)
Prerrequisito	Ninguno
Posrequisito	<i>SQM</i>
Vigencia	Primer semestre 2012
<b>Descripción del curso:</b>	
	Este curso busca familiarizar a los estudiantes con los fundamentos de la ingeniería de software y darles la capacidad de aplicar el análisis de sistemas y las técnicas de diseño para desarrollo de sistemas de software.
<b>Objetivos:</b>	
	<ul style="list-style-type: none"> <li>➤ Aprender los fundamentos de la ingeniería de software</li> <li>➤ Introducir conceptos y técnicas de modelado de sistemas en la Ingeniería de software.</li> </ul>
<b>Al finalizar el curso el estudiante estará capacitado para:</b>	
	<ul style="list-style-type: none"> <li>➤ Definir conceptos de sistemas y los tipos de sistemas de información.</li> <li>➤ Elaborar sistemas con base a su ciclo de vida.</li> <li>➤ Realizar análisis preliminares de software.</li> </ul>

Continuación del apéndice 7.

	<ul style="list-style-type: none"> <li>➤ Tomar decisiones con base a estudios de factibilidad.</li> <li>➤ Tomar requerimientos y realizar un modelado funcional en base a ellos.</li> </ul>		
<b>Metodología de enseñanza:</b>			
	<ul style="list-style-type: none"> <li>➤ Clases presenciales</li> <li>➤ Presentaciones en Power Point</li> </ul>		
<b>Evaluaciones:</b>			
	Asistencia:	10 puntos	
	Tareas/hojas de trabajo:	20 puntos	
	2 Exámenes cortos:	15 puntos c/u	30 puntos
	Examen final:	40 puntos	
	Total:	<b>100 Puntos</b>	
<b>Consideraciones:</b>			
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia.</li> <li>➤ La nota de aprobación es de 61 puntos.</li> </ul>		
<b>Bibliografía Recomendada</b>			
	<ul style="list-style-type: none"> <li>➤ <i>Software Engineering – A Practitioner’s Approach</i> <ul style="list-style-type: none"> <li>○ Roger S. Pressman (McGraw-Hill)</li> </ul> </li> <li>➤ <i>Object Oriented Modeling and Design</i> <ul style="list-style-type: none"> <li>○ Rumbaugh Blaha PremerlaniEddy Lorensen (Prentice-Hall)</li> </ul> </li> </ul>		
<b>Contenido:</b>			
<ul style="list-style-type: none"> <li>• Sistemas: conceptos</li> <li>• Introducción al análisis y diseño de sistemas.</li> </ul>		<ul style="list-style-type: none"> <li>• Requerimientos funcionales</li> <li>• Técnicas de modelación de sistemas.</li> </ul>	

Continuación del apéndice 7.

<ul style="list-style-type: none"> <li>• Introducción al ciclo de vida de software.</li> <li>• Modelos de ciclo de vida</li> <li>• Modelos alternativos de desarrollo</li> <li>• Análisis preliminar de sistemas</li> </ul>	<ul style="list-style-type: none"> <li>• Toma de requerimientos</li> <li>• Modelos de entidad-relación</li> <li>• Diagramas de entidad relación</li> <li>• Testeo y pruebas de calidad</li> <li>• Estimaciones</li> <li>• Técnicas de toma de decisión</li> </ul>
<p><b>Programación:</b></p>	
<ul style="list-style-type: none"> <li>• <b>Clase 1: Capítulo 1 libro de ayuda. “Introduction to Systems”</b> <ul style="list-style-type: none"> <li>○ Conceptos de sistemas</li> <li>○ Clasificación de sistemas</li> <li>○ Sistemas de información</li> <li>○ Introducción al análisis y diseño</li> <li>○ Función del analista de sistemas</li> <li>○ Reconocer quiénes son los usuarios del sistema</li> </ul> </li> <li>• <b>Clase 2: Capítulo 2 libro de ayuda “System Life Cycle Models”</b> <ul style="list-style-type: none"> <li>○ Introducción a los modelos de ciclo de vida de sistemas</li> <li>○ Ciclos de vida de sistemas           <ul style="list-style-type: none"> <li>▪ Sistema tradicional</li> <li>▪ Sistema de cascada</li> </ul> </li> <li>○ Sistemas alternativos de desarrollo           <ul style="list-style-type: none"> <li>▪ Prototipo</li> <li>▪ Metodología orientada a objetos</li> <li>▪ Análisis de sistemas dinámicos</li> <li>▪ <i>Rup</i> y <i>XP</i></li> </ul> </li> </ul> </li> <li>• <b>Clase 3: Capítulo 3 de libro de ayuda “Preliminary Analysis”</b> <ul style="list-style-type: none"> <li>○ Análisis preliminar y requerimientos iniciales</li> </ul> </li> </ul>	

Continuación del apéndice 7.

- Estudios de factibilidad
- Estimaciones
- **Clase 4: Capítulo 4 de libro de ayuda “*FactFinding and DecisionMakingTechniques*”**
  - Técnicas de toma de requerimientos
  - Toma de decisión y documentación
- **Primer examen corto**
  - **Clase 5: Capítulo 5 del libro de ayuda “*FunctionalModelling*”**
  - Requerimientos funcionales
  - Técnicas de modelaje funcional
- **Clase 6: Capítulo 6 del libro de ayuda “*FunctionalModelling - II*”**
  - Proceso de especificaciones
  - Modelo de control de flujo
  - Especificaciones de control
  - Estructura de gráficos
  - Codificación
  - Diccionario de datos
- **Clase 7: Capítulo 7 del libro de ayuda “*Data Modelling*”**
  - Requerimiento de datos
  - Técnicas de modelado de datos E-R
  - Conceptos de modelo E-R
  - Diagramas E-R
- **Clase 8: Capítulo 8 del libro de ayuda “*Other Data ModellingTechniques*”**
  - Modelo relacional
  - Modelo orientado a objetos
- **Segundo examen corto**



Continuación del apéndice 7.

- **Clase 9: Capítulo 9 del libro de ayuda “Testing And QualityAssurance”**
  - Fundamentos de pruebas de software
  - Pruebas de caja blanca
  - Pruebas de unidad
  - Pruebas de integración
  - Pruebas de validación
  - Pruebas de sistemas
  - Papel de las pruebas de calidad de software
  - Actividades en las pruebas de calidad
- **Clase 10:**
- **Examen final**

Fuente: elaboración propia.

Apéndice 8. **Programa de curso: SQM**

<b>Información general</b>	
Área	<i>Software Engineering and OOAD</i>
Duración	10 horas (5 clases de dos horas cada clase)
Prerrequisito	<i>ISE</i>
Posrequisito	SA
Vigencia	Primer semestre 2012

Continuación del apéndice 8.

<b>Descripción del curso:</b>			
	Este curso busca que el estudiante comprenda los conceptos de calidad y la importancia de la misma en el desarrollo de software, dándole a conocer las distintas herramientas o normas existentes para llevar el control y asegurar la calidad en todo el ciclo de vida de software.		
<b>Objetivos:</b>			
	<ul style="list-style-type: none"> <li>➤ Aprender los conceptos de la gestión de proyectos</li> <li>➤ Comprender los principios y técnicas de testeo</li> <li>➤ Conocer el concepto de calidad y su aplicación</li> </ul>		
<b>Al finalizar el curso el estudiante estará capacitado para:</b>			
	<ul style="list-style-type: none"> <li>➤ Definir conceptos de ingeniería de software y testeo</li> <li>➤ Entender la gestión de proyectos</li> <li>➤ Conocer los conceptos de calidad y saberlos implementar en el proceso de desarrollo de software</li> <li>➤ Medir la calidad en el desarrollo de software mediante herramientas de gestión de calidad</li> <li>➤ Conocer y aplicar normas estándar de calidad</li> </ul>		
<b>Metodología de enseñanza:</b>			
	<ul style="list-style-type: none"> <li>➤ Clases presenciales</li> <li>➤ Presentaciones en <i>Power Point</i></li> </ul>		
<b>Evaluaciones:</b>			
	Asistencia:	10 puntos	
	Tareas/hojas de trabajo:	20 puntos	
	2 Exámenes cortos:	15 puntos c/u	30 puntos
	Examen final:	40 puntos	
	Total:	<b>100 Puntos</b>	

Continuación del apéndice 8.

<b>Consideraciones:</b>	
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia</li> <li>➤ La nota de aprobación es de 61 puntos</li> </ul>
<b>Contenido:</b>	
<ul style="list-style-type: none"> <li>• Modelos de ciclo de vida de software.</li> <li>• Verificación y validación <ul style="list-style-type: none"> <li>• Enfoque de la ingeniería en el desarrollo de software.</li> </ul> </li> <li>• Características de gestión de proyectos</li> <li>• Áreas de gestión de proyectos.</li> <li>• Conceptos de calidad de software.</li> <li>• Aseguramiento de la calidad y sus ventajas.</li> </ul>	<ul style="list-style-type: none"> <li>• Gestión de calidad total</li> <li>• Costo de la calidad</li> <li>• Gestión de la calidad mediante mediciones.</li> <li>• Herramientas de gestión de calidad.</li> <li>• Control estadístico de procesos</li> <li>• Normas de calidad</li> <li>• Aseguramiento de calidad vs control de calidad.</li> <li>• Requisitos legales de las normas de calidad.</li> </ul>
<b>Programación:</b>	
<ul style="list-style-type: none"> <li>• <b>Clase 1: Software <i>Engineering and Testing</i></b> <ul style="list-style-type: none"> <li>○ Modelos de ciclo de vida de software <ul style="list-style-type: none"> <li>▪ Cascada</li> <li>▪ Modelo en V</li> <li>▪ Prototipado</li> <li>▪ Incremental</li> <li>▪ Espiral</li> </ul> </li> <li>○ Verificación y validación</li> </ul> </li> </ul>	

Continuación del apéndice 8.

- Principios de testeo
- Técnicas de testeo
- Estrategias de testeo
- Herramientas de testeo
- Enfoque de la ingeniería en el desarrollo de software
  - Tecnología CASE
  - Modelo de análisis
- **Clase 2: *Project Management***
  - Características de la gestión de proyectos
  - Ciclo de vida de proyectos
  - Descripción general de las áreas de gestión de proyectos
    - Gestión de integración
    - Gestión de alcance
    - Gestión de tiempos
    - Gestión de costos
    - Gestión de la calidad
    - Gestión de riesgos
    - Gestión de recurso humano
    - Gestión de comunicaciones
- **Primer examen corto**
- **Clase 3: *Quality Concepts***
  - ¿Qué es calidad?
  - Evolución de la calidad
    - Calidad por estándar
    - Calidad por uso
    - Calidad por costo
    - Calidad por necesidades latentes

Continuación del apéndice 8.

- Filosofías y definiciones de calidad
- Aseguramiento de la calidad Vs control de calidad
- Gestión de calidad total
  - Enfoque del cliente
  - Mejora continua
  - Participación total
- Costo de la calidad
- Aseguramiento de la calidad de software
- Ventajas del aseguramiento de la calidad
- **Clase 4: *Quantitative methods for Quality Management***
  - Medición de la gestión de calidad
  - Herramientas para la mejora de la calidad
  - 7 herramientas para la mejora de calidad
    - Herramientas para la identificación de problemas
    - Herramientas para el análisis de problemas
  - Control estadístico de procesos
    - Medidas de calidad de software
    - Medidas de productividad
    - Medidas de complejidad
  - Línea base
- **Segundo examen corto**
- **Clase 5: *Quality Standards***
  - Normas y su cumplimiento
  - Normas de calidad
    - Kaisen y las 5s
    - ISO 9000
    - ISO 14000

Continuación del apéndice 8.

<ul style="list-style-type: none"> <li>▪ ¿ISO o TQM?</li> <li>▪ Normas BSI</li> <li>▪ Modelo capacidad de madurez</li> <li>▪ Six—Sigma</li> <li>○ Requerimientos legales</li> <li>○ Cuestiones profesionales</li> <li>• <b>Examen final</b></li> </ul>
--

Fuente: elaboración propia.

Apéndice 9. **Programa de curso: SA**

<b>Información general</b>	
Área	Software <i>Engineering and OOAD</i>
Duración	16 horas (8 clases de dos horas cada clase)
Prerrequisito	<i>SQM</i>
Posrequisito	Ninguno
Vigencia	Primer semestre 2012
<b>Descripción del curso:</b>	
	Este curso busca que los alumnos puedan analizar los requerimientos del negocio en un escenario determinado y definir la solución técnica con las distintas arquitecturas para optimizar los resultados empresariales mediante el desarrollo de herramientas de Microsoft.

Continuación del apéndice 9.

<b>Objetivos:</b>	
	<ul style="list-style-type: none"> <li>➤ Aprender técnicas de toma y análisis de requerimientos</li> <li>➤ Conocer las herramientas de Microsoft para desarrollo</li> <li>➤ Aprender a aplicar modelos de datos</li> </ul>
<b>Al finalizar el curso el estudiante estará capacitado para:</b>	
	<ul style="list-style-type: none"> <li>➤ Manejar el <i>Framework</i> de soluciones de Microsoft</li> <li>➤ Analizar requerimientos y realizar planificaciones de proyectos en base a ellos.</li> <li>➤ Diseñar soluciones de software</li> <li>➤ Discernir entre las diferentes arquitecturas de tecnologías</li> <li>➤ Estandarizar sus aplicaciones y diseñar interfaces de usuario.</li> <li>➤ Aplicar modelos de datos</li> </ul>
<b>Metodología de enseñanza:</b>	
	<ul style="list-style-type: none"> <li>➤ Clases presenciales</li> <li>➤ Presentaciones en <i>Power Point</i></li> </ul>
<b>Evaluaciones:</b>	
	Asistencia: 10 puntos
	Tareas/hojas de trabajo: 20 puntos
	1 examen corto: 20 puntos
	Examen final: 50 puntos
	<b>Total: 100 Puntos</b>
<b>Consideraciones:</b>	
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia.</li> <li>➤ La nota de aprobación es de 61 puntos.</li> </ul>

Continuación del apéndice 9.

<b>Bibliografía sugerida:</b>	
	<ul style="list-style-type: none"> <li>➤ MCSD Guía de Estudio - <i>SOLUTION ARCHTECTURE</i> <ul style="list-style-type: none"> <li>○ Ben Ezzel —BPB—</li> </ul> </li> <li>➤ Guía de preparación de exámenes - <i>SOLUTION ARCHTECTURE</i> <ul style="list-style-type: none"> <li>○ Keith Morneau —IDG—</li> </ul> </li> </ul>
<b>Contenido:</b>	
<ul style="list-style-type: none"> <li>• Desarrollo de soluciones del ciclo de vida y MSF</li> <li>• <i>Team model</i></li> <li>• Modelo de procesos</li> <li>• Modelo de aplicación</li> <li>• Modelo de arquitectura empresarial.</li> <li>• Modelo de infraestructura</li> <li>• Modelo de costo total del propietario (TCO)</li> <li>• Fase de planeamiento</li> <li>• Modelo de diseño de soluciones.</li> <li>• Fundamentos de <i>Networking</i></li> </ul>	<ul style="list-style-type: none"> <li>• Elección de Hardware, sistema operativo</li> <li>• Elección de tecnologías de aplicación Microsoft</li> <li>• Principios de diseño</li> <li>• Estandarización de vistas</li> <li>• Asistente de usuario</li> <li>• Gestión de configuración de software</li> <li>• Proceso de testeo</li> <li>• Desarrollo de sistemas</li> <li>• Aplicación de modelo de datos</li> <li>• Análisis entidad relación</li> <li>• Normalización e integridad de datos.</li> </ul>
<b>Programación:</b>	
<ul style="list-style-type: none"> <li>• <b>Clase 1: Microsoft <i>Solutions Framework</i></b> <ul style="list-style-type: none"> <li>○ Desarrollo de soluciones del ciclo de vida y MSF</li> <li>○ <i>Team model</i></li> </ul> </li> </ul>	



Continuación del apéndice 9.

- Roles en el equipo
- Beneficios del *team model*
- Modelo de procesos
  - Fase de previsión
  - Fase de planificación
  - Fase de desarrollo
  - Fase de estabilización
  - Beneficios del modelo de proceso
- Modelo de diseño de soluciones
  - Diseño conceptual
  - Diseño lógico
  - Diseño físico
  - La flexibilidad en las perspectivas
  - Beneficios del modelo de diseño de soluciones
- Modelo de aplicación
- Modelo de arquitectura empresarial
  - Arquitectura empresarial
  - Arquitectura de aplicaciones
  - Arquitectura de la información
  - Tecnología de la arquitectura
  - Beneficios del modelo de arquitectura empresarial
- Modelo de infraestructura
- Modelo costo total de propietario (TCO)
- **Clase 2: *Project Planning and Analysing Business Requirements***
  - Fase de previsión
  - Fase de planeación

Continuación del apéndice 9.

- Análisis de los requerimientos del negocio
- Gestión de riesgos
- Casos de uso
- Análisis funcional de requerimientos
- Calendarización de proyectos
- **Clase 3: *Designing the Solution***
  - Modelo de diseño de la solución
    - Vista del proceso de diseño
    - Diseño lógico
    - Factores de riesgo
    - Diseño físico
  - Modelo de aplicación
    - Modelo de servicio
    - Arquitectura por niveles
    - Microsoft *Transaction Server*
- **Clase 4: *Technology Architecture***
  - Fundamentos de *Networking*
    - Roles en equipos de la red
    - Tipos de redes
    - Arquitectura de red
    - Protocolos
    - Modelo OSI
  - Elección de hardware
  - Elección de sistema operativo
  - Elección de aplicaciones +
  - Elección de tipo de solución
  - Elección de arquitectura de almacenamiento de datos

Continuación del apéndice 9.

- **Primer examen corto**
- **Clase 5: *User Interface Design and Application Development***
  - Principios de diseño
  - Estandarización de vistas
  - Problemas de interfaz
  - Consideraciones de desarrollo de plataformas
  - Asistencia al usuario
- **Clase 6 *Configuration Management and Deployment Strategies***
  - Gestión de la configuración de software
  - Fase de estabilización
  - Proceso de testeo
    - Tipo de testeo
    - Gestión de entregas
    - Reporte de errores en entregas
  - Implementación del sistema
  - Problemas de implementación y los desafíos
    - Despliegue media
    - Métodos de implementación
    - Pruebas de laboratorio de una implementación
    - Piloto de implementación
    - Finalización de la implementación
- **Clase 7 *Applying Data Models***
  - Asociación de datos
    - *DBMS* de archivos planos
    - *DBMS* jerárquica
    - *DBMS* relacional
  - Análisis de entidad relación

Continuación del apéndice 9.

<ul style="list-style-type: none"><li>▪ Entidades y atributos</li><li>▪ Relaciones</li><li>○ Normalización<ul style="list-style-type: none"><li>▪ Primera forma normal</li><li>▪ Segunda forma normal</li><li>▪ Tercera forma normal</li><li>▪ Impacto de la normalización</li><li>▪ Desnormalización</li></ul></li><li>○ Integridad de datos<ul style="list-style-type: none"><li>▪ Integridad de la entidad</li><li>▪ Integridad referencial</li></ul></li><li>○ Reglas del negocio y limitaciones</li><li>○ Arquitectura de almacenamiento de datos</li><li>○ Replicación de base de datos</li><li>• <b>Clase 8: examen final</b></li></ul>
--

Fuente: elaboración propia.

Apéndice 10. **Programa de curso: Java básico**

<b>Información general</b>	
Área	<i>Web Programming using Java</i>
Duración	40 horas (10 clases teóricas y 10 prácticas de 2 horas cada clase)
Prerrequisito	Ninguno
Posrequisito	Java intermedio
Vigencia	Primer semestre 2012

Continuación del apéndice 10.

<b>Descripción del curso:</b>			
	Que el estudiante salga con conceptos básicos de la programación orientada a objetos, los cuales pueda aplicar dentro del lenguaje de programación Java.		
<b>Objetivos:</b>			
	<ul style="list-style-type: none"> <li>➤ Familiarizar al estudiante con programación orientada a objetos</li> <li>➤ Introducir al estudiante en la programación en Java</li> <li>➤ Proveer al estudiante de conceptos básicos en la programación</li> </ul>		
<b>Al finalizar el curso el estudiante estará capacitado para:</b>			
	<ul style="list-style-type: none"> <li>➤ Indicar conceptos de programación orientada a objetos</li> <li>➤ Tener nociones de las instrucciones de control y ciclos en programación.</li> <li>➤ Realizar llamadas a métodos y funciones en Java</li> <li>➤ Manejar los conceptos básicos de programación en Java</li> </ul>		
<b>Metodología de enseñanza:</b>			
	<ul style="list-style-type: none"> <li>➤ Clases presenciales</li> <li>➤ Presentaciones en <i>Power Point</i></li> <li>➤ Clases prácticas</li> </ul>		
<b>Evaluaciones:</b>			
	Tareas:		15 puntos
	3 Exámenes cortos:	10 puntos c/u	30 puntos
	Proyecto:		25 puntos
	Examen final		30 puntos
	Total:		<b>100 Puntos</b>

Continuación del apéndice 10.

<b>Consideraciones:</b>	
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia</li> <li>➤ La nota de aprobación es de 61 puntos</li> <li>➤ Debe entregar ambas prácticas y proyectos para optar a examen final.</li> </ul>
<b>Bibliografía sugerida:</b>	
	<ul style="list-style-type: none"> <li>➤ Programación en Java 2 <ul style="list-style-type: none"> <li>○ Luis Joyanes Aguilar e Ignacio Zahonero Martínez <ul style="list-style-type: none"> <li>▪ McGraw-Hill</li> </ul> </li> </ul> </li> <li>➤ Java, Cómo Programar <ul style="list-style-type: none"> <li>○ Harvey M. Deitel, Paul J. Deitel. 5ª ed 2004 <ul style="list-style-type: none"> <li>▪ Pearson, Prentice-Hall</li> </ul> </li> </ul> </li> </ul>
<b>Contenido:</b>	
<ul style="list-style-type: none"> <li>• Definición en Java</li> <li>• Versiones de Java y su entorno.</li> <li>• Características de Java</li> <li>• Tipos de programa</li> <li>• Operadores</li> <li>• Conceptos de <i>OOP</i></li> <li>• Administración de memoria</li> <li>• Constructores</li> </ul>	<ul style="list-style-type: none"> <li>• Herencia y polimorfismo</li> <li>• Modificadores y control de acceso</li> <li>• Clases y métodos abstractos</li> <li>• Excepciones y manejo de errores</li> <li>• Memoria dinámica</li> <li>• Lectura de archivos</li> <li>• Hilos</li> <li>• <i>Api Swing</i></li> <li>• Clases, objetos</li> </ul>
<b>Programación:</b>	
<ul style="list-style-type: none"> <li>• <b>Clase 1:</b> <ul style="list-style-type: none"> <li>○ ¿Qué es Java?</li> <li>○ Versiones de Java y su entorno.</li> </ul> </li> </ul>	

Continuación del apéndice 10.

- **Clase 2:**
  - Tipos de programas
    - *Applets*
    - Aplicaciones
  - Componentes del lenguaje Java
    - Operadores
- **Clase 3:**
  - Concepto de programación orientada a objetos
    - Clase
    - Abstracción
    - Encapsulamiento
    - Herencia
    - Polimorfismo
- **Examen corto 1**
- **Clase 4**
  - Clases y objetos
  - Métodos *get* y *set*
  - Administración de memoria
- **Clase 5**
  - Control de Acceso y modificadores
    - *default*
    - *public*
    - *private*
    - *protected*
    - *static*
    - *final*
- **Examen corto 2**

Continuación del apéndice 10.

- **Clase 6**
  - Clases y métodos abstractos
  - Interfaces
  - ¿Qué es un paquete?
- **Clase 7:**
  - Excepciones.
    - *Try, throw and catch*
    - *Throws*
    - *Finally*
  - Declaraciones anidadas
  - ¿Por qué no usar excepciones?
- **Clase 8**
  - Hilos
  - Sincronización
- **Examen corto 3**
- **Clase 9**
  - *AWT*
    - *Label*
    - *Button*
    - *Checkbox*
    - *Radio Buttons*
    - *Choice Control*
    - *List Control*
    - *Text field and text area*
    - *Panel*
- **Clase 10**
  - *Layout managers*



Continuación del apéndice 10.

<ul style="list-style-type: none"> <li>▪ <i>Flow layout</i></li> <li>▪ <i>Border layout</i></li> <li>▪ <i>Card layout</i></li> <li>• <b>Clase 11-20</b> <ul style="list-style-type: none"> <li>○ <i>Laboratorio y desarrollo del proyecto</i></li> </ul> </li> <li>• <b>Examen final</b></li> </ul>
---

Fuente: elaboración propia.

Apéndice 11.      **Programa de curso: Java Intermedio**

<b>Información general</b>	
Área	<i>Web Programming using Java</i>
Duración	32 horas (8 clases teóricas y 8 prácticas de dos horas cada clase)
Prerrequisito	Java básico
Posrequisito	Java avanzado
Vigencia	Primer semestre 2012
<b>Descripción del curso:</b>	
	Este curso busca familiarizar a los estudiantes con la programación orientada a objetos y las características básicas de Java para el desarrollo de aplicaciones de negocios en la plataforma Java.
<b>Objetivos:</b>	
	<ul style="list-style-type: none"> <li>➤ Entender qué es la programación orientada a objetos</li> <li>➤ Aprender las tecnologías Java Beans, servlets, JDBC y Swing.</li> </ul>

Continuación del apéndice 11.

<b>Al finalizar el curso el estudiante estará capacitado para:</b>			
	<ul style="list-style-type: none"> <li>➤ Realizar aplicaciones Java Beans</li> <li>➤ Hacer <i>Servlets</i></li> <li>➤ Conectarse a una base de datos mediante <i>JDBC</i></li> <li>➤ Utilizar la librería <i>Swing</i></li> <li>➤ Conocer la <i>Collection Api</i> de Java</li> </ul>		
<b>Metodología de enseñanza:</b>			
	<ul style="list-style-type: none"> <li>➤ Clases presenciales</li> <li>➤ Presentaciones en <i>Power Point</i></li> </ul>		
<b>Evaluaciones:</b>			
	Asistencia:	10 puntos	
	1 proyecto:	30 puntos	
	2 Exámenes cortos:	10 puntos c/u	20 puntos
	Examen final:	40 puntos	
	Total:	<b>100 Puntos</b>	
<b>Consideraciones:</b>			
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia</li> <li>➤ La nota de aprobación es de 61 puntos</li> </ul>		
<b>Bibliografía Recomendada</b>			
	<ul style="list-style-type: none"> <li>➤ <i>Java 2 Platform Unleashed</i> <ul style="list-style-type: none"> <li>○ Jamie Jaworski (Techmedia SAMS)</li> </ul> </li> <li>➤ <i>Using Java 2 Platform</i> <ul style="list-style-type: none"> <li>○ Joseph L. Weber (PHI)</li> </ul> </li> <li>➤ <i>The Complete Reference Java 2 Platform</i> <ul style="list-style-type: none"> <li>○ Herbert Schildt (TMH)</li> </ul> </li> </ul>		

Continuación del apéndice 11.

<b>Contenido:</b>	
<ul style="list-style-type: none"><li>• <i>Collection framework</i></li><li>• <i>Collection Api</i></li><li>• <i>Swing vs AMT</i></li><li>• <i>Swing components</i></li><li>• <i>Secure applets</i></li><li>• <i>Jars</i></li><li>• <i>JDBC</i></li></ul>	<ul style="list-style-type: none"><li>• Clases URL</li><li>• <i>TCP sockets</i></li><li>• <i>UDP sockets</i></li><li>• Diseño distribuido</li><li>• <i>Servlets</i></li><li>• <i>Java beans</i></li></ul>
<b>Programación:</b>	
<ul style="list-style-type: none"><li>• <b>Clase 1:</b><ul style="list-style-type: none"><li>○ <i>Collection framework</i><ul style="list-style-type: none"><li>▪ Definición</li><li>▪ Componentes.</li></ul></li><li>○ Clases de la <i>collection</i> API<ul style="list-style-type: none"><li>▪ <i>Enumeration</i></li><li>▪ <i>ListIterator</i></li><li>▪ <i>Map</i></li><li>▪ <i>LinkedList</i></li><li>▪ <i>Vector</i></li><li>▪ <i>Stack</i></li><li>▪ <i>HashMap</i></li><li>▪ <i>HashSet</i></li></ul></li></ul></li><li>• <b>Clase 2:</b><ul style="list-style-type: none"><li>○ Comparación entre <i>Swing</i> y <i>AWT</i></li><li>○ Modelo vista controlador</li><li>○ Componentes de la clase <i>Swing</i></li></ul></li></ul>	

Continuación del apéndice 11.

- *Label*
- *JFC Applet*
- *Button*
- *TextField*
- *TextArea*
- *Borders*
- *Check boxes*
- *Radio buttons*
- *Layout managers*
  - *Box layout*
  - *Grid layout*
  - *GridBag layout*
- **Primer examen corto**
- **Clase 3:**
  - *List control*
  - *Combo box*
  - *List and combo box model*
  - *Menus*
    - *Menu items*
    - *Accelerator and mnemonic*
    - *Popup menus*
  - *Creating your icons*
  - *Tabbed panes*
  - *Toolbar*
  - *Progressbar*
  - *Slider control*
  - *Dialog boxes*

Continuación del apéndice 11.

- *Tables*
- *Table model*
- **Clase 4:**
  - Características de Java en la construcción de *Applets* seguros
  - Fases de la ejecución de una *Java Application* o *Applet*
    - *Java compiler*
    - *Verifier*
    - *Class loader*
  - Introducción a *Security Model*
    - *Security Model of J2EE*
  - *Some Security Permission Classes of J2EE*
    - *Java.security.Permission*
    - *Java.io.FilePermission*
    - *Java.net.SocketPermission*
    - *Java.util.PropertyPermission*
    - *Java.lang.RuntimePermission*
    - *Java.security.AllPermission*
  - JARS
    - Creación de JAR files
    - Como utilizar un archivo JAR con la etiqueta APPLET
  - *Signing Applets*
  - JDBC
    - *Remote Database Access*
    - Clasificación of JDBC *Drivers*
    - Mejoras en el JDBC 3.0 API
    - *Accessing Oracle database.*
  - *Database Security*

Continuación del apéndice 11.

- **Clase 5:**
  - Introducción a *URL*
  - *URL Class*
  - *URLConnection Class*
  - *Introduction to sockets*
  - *TCP Sockets*
    - *Server socket*
    - *Client socket*
  - *UDP Sockets*
    - *Server socket*
    - *Client socket*
  - *Multithreaded communication*
    - *Threaded server*
    - *Multithreaded server*
  - *Passing objects between client and server*
- **Segundo Examen Corto**
- **Clase 6:**
  - *Distributed application design*
    - *Distributed programming on Internet and Intranets*
    - *Distributed Computing Environment (DCE)*
    - *Distributed Component Object Model (DCOM)*
    - *Common Object Request Broker Architecture (CORBA)*
    - *Remote Method Invocation (RMI)*
  - *Java distributed object model*
    - *Advantages*
    - *Disadvantages*
  - *Remote method invocation*

Continuación del apéndice 11.

- *RMI API*
- *Implementing RMI*
- *Employee database*
- *Servlets*
  - *Historia.*
  - *Introducción a Java Servlets*
  - *Java Servlet API*
  - *Creating a servlet*
  - *Handling HEAD requests*
  - *init() method*
- *Communication between Applets and Servlets*
- **Clase 7:**
  - *Introduction to JavaBeans*
    - *Component technology*
    - *Components and containers*
    - *Introspection*
    - *Bean properties*
    - *Accessor methods*
    - *Interface methods*
    - *Persistence*
    - *Events*
  - *Using BDK*
    - *Using existing beans to develop a component*
    - *Converting a component into an Applet*
    - *The other Beans in the BeanBox*
    - *Other BeanDevelopment tools*
  - *JavaBeans API*

Continuación del apéndice 11.

- *JavaBeans Design*
  - *Creating a bean*
  - *Using this bean in an IDE*
  - *Creating a bean that has bound and vetoable property*
  - *Using the bean's bound and vetoable property*
- *Customizing Beans*
  - *Creating a Bean Customizer*
  - *Attaching a Bean Customizer with the Bean*
  - *Creating Property Editors*
  - *Text-Only Property Editors*
  - *Property Editors with tags*
  - *Custom Property Editors*
- **Clase 8-15**
  - Elaboración de proyecto
- **Clase 16: examen final**

Fuente: elaboración propia.

## Apéndice 12. Programa de curso: Java avanzado

<b>Información general</b>	
Área	<i>Web Programming using Java</i>
Duración	34 horas (8 clases teóricas y 9 prácticas de 2 horas cada clase)
Prerrequisito	Java intermedio
Posrequisito	Ninguno
Vigencia	Primer semestre 2012



Continuación del apéndice 12.

<b>Descripción del curso:</b>		
	Este curso busca orientar al estudiante en un nivel avanzado de programación en el lenguaje Java, abarcando temas sobre <i>RDBMS</i> , programación <i>web</i> , programación con <i>servlets</i> , entre otros.	
<b>Objetivos:</b>		
	<ul style="list-style-type: none"> <li>➤ Aprender sobre el desarrollo de aplicaciones en la plataforma <i>J2EE</i>.</li> <li>➤ Conocer el concepto de <i>servlet</i></li> <li>➤ Conocer el concepto de administración <i>Web</i></li> </ul>	
<b>Al finalizar el curso el estudiante estará capacitado para:</b>		
	<ul style="list-style-type: none"> <li>➤ Indicar conceptos de programación web</li> <li>➤ Crear y utilizar <i>Java servlets</i> y <i>Java server pages</i></li> <li>➤ Manipular datos desde un <i>servlet</i></li> <li>➤ Configurar parámetros de <i>servlets</i></li> <li>➤ Manejar <i>servlets</i> y bases de datos</li> </ul>	
<b>Metodología de enseñanza:</b>		
	<ul style="list-style-type: none"> <li>➤ Clases presenciales</li> <li>➤ Presentaciones <i>en Power Point</i>.</li> <li>➤ Clases prácticas</li> </ul>	
<b>Evaluaciones:</b>		
	Tareas:	15 puntos
	2 Exámenes cortos:	10 puntos c/u
	Proyecto:	35 puntos
	Examen final	30 puntos
	Total:	<b>100 Puntos</b>

Continuación del apéndice 12.

<b>Consideraciones:</b>	
	<ul style="list-style-type: none"> <li>➤ El estudiante debe cumplir con el 80% de asistencia</li> <li>➤ La nota de aprobación es de 61 puntos</li> <li>➤ Debe entregar ambas prácticas y proyectos para optar a examen final.</li> </ul>
<b>Bibliografía sugerida:</b>	
	<ul style="list-style-type: none"> <li>➤ <i>Core Exploring Java Servlets And JavaServer Pages</i> <ul style="list-style-type: none"> <li>○ MARTY HALL</li> <li>○ Prentice Hall PTR</li> </ul> </li> </ul>
<b>Contenido:</b>	
<ul style="list-style-type: none"> <li>• <i>Arquitectura Web</i></li> <li>• <i>Web Applications</i></li> <li>• Tecnologías de lado del servidor</li> <li>• <i>Servlets</i></li> <li>• <i>Java server pages</i></li> <li>• Contenedores</li> <li>• Ciclo de vida de un <i>servlet</i></li> <li>• Inicialización de parámetros</li> </ul>	<ul style="list-style-type: none"> <li>• Servidor</li> <li>• El cliente</li> <li>• La consulta</li> <li>• Paquetes <i>HTML</i></li> <li>• Cadenas de <i>servlet</i></li> <li>• Conexión <i>JDBC-ODBC</i></li> <li>• Implementaciones <i>JDBC</i></li> <li>• Transacciones</li> <li>• <i>HttpServlet Básico</i></li> </ul>
<b>Programación:</b>	
<ul style="list-style-type: none"> <li>• <b>Clase 1:</b> <ul style="list-style-type: none"> <li>○ <i>Arquitectura Web</i> <ul style="list-style-type: none"> <li>▪ <i>Navegador web</i></li> <li>▪ <i>Web Server</i></li> </ul> </li> <li>○ <i>Aplicaciones web</i> <ul style="list-style-type: none"> <li>▪ <i>Data collection</i> del lado del servidor</li> </ul> </li> </ul> </li> </ul>	

Continuación del apéndice 12.

- Solicitud
- Procesamiento de datos y ejecución de *script* en el lado del servidor.
- Respuesta
- Tecnología del lado del servidor
  - *Common Gateway interfaz*
  - *Active Server pages*
  - *Java script* del lado del servidor
- *Servlets y Java server pages*
  - ¿Qué es un *servlet*?
  - El poder de los *servlet*
  - Extendiendo *servlets* a través de la tecnología Java
  - El poder de *JSP*
- **Clase 2:**
  - El contenedor de un *servlet*
  - Ciclo de vida de un *servlet*
  - *Servlet API*
  - *HttpServlet* básico
  - Manejo de datos en formularios de *servlet*
  - Configuración *Servlet* e inicialización de parámetros
  - Manejo de excepciones
  - Manejo de errores en *servlets*
- **Clase 3:**
  - Inicialización de parámetros
    - Configurando un parámetro de inicio
    - Obteniendo un parámetro de inicio
    - Usando un parámetro de inicio

Continuación del apéndice 12.

- Recuperar nombres de parámetros
- Modelo *Single-Thread*
- El servidor
- El cliente
- La respuesta
- **Examen corto 1**
- **Clase 4**
  - Creación de paquetes *HTML*
  - Códigos de estado
  - *HTTP Headers*
- **Clase 5**
  - ¿Qué es un *servlet chain*?
  - ¿Cómo invocar un *servlet chain*?
    - Encadenar *servlet* usando *servlet alias*
    - Encadenar *servlet* usando *MIME types*.
  - *Server Side Include*
    - *Servlet Alias*
    - Convocar un *Servlet* desde un *HTML*
    - Convocar un *servlet* desde otro *servlet*
- **Clase 6**
  - Sesión y transacción
  - Rastreo de sesión
    - Campos ocultos
    - Rescritura de *URL*
    - Uso de *cookies* persistentes
    - Uso de rastreo de sesión *servlet API*
  - Comunicación interservlet

Continuación del apéndice 12.

- **Clase 7:**
  - Acceso a base de datos remoto
  - Clasificación de *drivers JDBC*
  - Conexión *JDBC-ODBC*
  - Implementación *JDBC*
  - Conexión a base de datos
  - Usando sentencias preparadas
  - Usando transacciones
  - *Access Oracle database*
  - Procedimientos almacenados con base de datos Oracle
  - Seguridad de base de datos
  - Paquete de conexiones
  - Metadata
- **Clase 8**
  - *Java Applets*
  - *RMI*
  - Comunicación *Servlet Applets*
  - Beneficios de usar *RMI* en *applets*
  - Limitaciones de *RMI* en *applets*
- **Clase 9-17**
  - Desarrollo de proyecto
- **Examen final**

Fuente: elaboración propia.