



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

**IMPLEMENTACIÓN DEL CURSO *IPE* BASADO EN LA METODOLOGÍA DEL *ITCoE* PARA  
DESARROLLAR UNA MEJOR ABSTRACCIÓN EN LOS CONCEPTOS DE PROGRAMACIÓN  
DESARROLLADO PARA EL CENTRO TECNOLÓGICO DE LA INDIA DE LA FACULTAD DE  
INGENIERÍA USAC**

**Alex Steven Campos Hernández**  
**Josué Rodolfo Obregón Velásquez**  
**Sergio Antonio Villalta Montenegro**  
Asesorados por la Inga. Gladys Aceituno

Guatemala, mayo de 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**IMPLEMENTACIÓN DEL CURSO *IFE* BASADO EN LA METODOLOGÍA DEL *ITCoE* PARA  
DESARROLLAR UNA MEJOR ABSTRACCIÓN EN LOS CONCEPTOS DE  
PROGRAMACIÓN, DESARROLLADO PARA EL CENTRO TECNOLÓGICO DE LA INDIA DE  
LA FACULTAD DE INGENIERÍA, USAC**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA  
DE LA FACULTAD DE INGENIERÍA  
POR

**ALEX STEVEN CAMPOS HERNÁNDEZ  
JOSUÉ RODOLFO OBREGÓN VELÁSQUEZ  
SERGIO ANTONIO VILLALTA MONTENEGRO**  
ASESORADOS POR LA INGA. GLADYS ACEITUNO

AL CONFERÍRSELES EL TÍTULO DE  
**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, MAYO DE 2011

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

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADORA	Inga. Sonia Yolanda Castañeda
EXAMINADORA	Inga. Floriza Felipa Ávila Pesquera de Medinilla
EXAMINADOR	Ing. Marlon Antonio Pérez Turk
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, presentamos a su consideración nuestro trabajo de graduación titulado

**IMPLEMENTACIÓN DEL CURSO *IFE* BASADO EN LA METODOLOGÍA DEL *ITCOE* PARA DESARROLLAR UNA MEJOR ABSTRACCIÓN EN LOS CONCEPTOS DE PROGRAMACIÓN DESARROLLADO PARA EL CENTRO TECNOLÓGICO DE LA INDIA DE LA FACULTAD DE INGENIERÍA USAC**

Tema que nos fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, en agosto de 2009.



Alex Steven Campos Hernández



Josué Rodolfo Obregón Velásquez



Sergio Antonio Villalta Montenegro



Guatemala, 05 de febrero de 2011

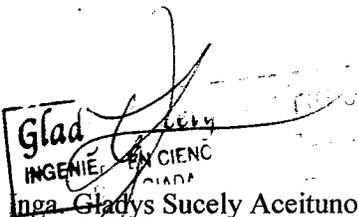
Ingeniera  
Norma Iliana Sarmiento  
Dirección Unidad de EPS  
Facultad de Ingeniería  
USAC

Ingeniera Sarmiento:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **ALEX STEVEN CAMPOS HERNÁNDEZ, JOSUÉ RODOLFO OBREGÓN VELÁSQUEZ y SERGIO ANTONIO VILLALTA MONTENEGRO**, titulado **“Implementación del curso IPE basado en la metodología del ITCoE para desarrollar una mejor abstracción en los conceptos de programación, desarrollado para el centro tecnológico de la India de la Facultad de Ingeniería, USAC”**, y a mi criterio, el mismo cumple con los objetivos propuestos para su desarrollo según el protocolo.

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

Atentamente,

  
INGENIERA EN CIENCIAS Y SISTEMAS  
Inga Gladys Sucely Aceituno  
Ingeniera en Ciencias y Sistemas  
Asesora del Proyecto



Guatemala, 17 de febrero de 2011.  
REF.EPS.DOC.263.02.2011.

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

Estimada Ingeniera Sarmiento Zeceña.

Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) de los estudiantes universitarios de la Carrera de Ingeniería en Ciencias y Sistemas, **Josué Rodolfo Obregón Velásquez** Carné No. **200412545**, **Sergio Antonio Villalta Montenegro** Carné No. **200412738** y **Alex Steven Campos Hernández** Carné No. **200412793** procedí a revisar el informe final, cuyo título es **“IMPLEMENTACIÓN DEL CURSO IPE BASADO EN LA METODOLOGÍA DEL ITCoE, PARA DESARROLLAR UNA MEJOR ABSTRACCIÓN DE LOS CONCEPTOS DE PROGRAMACIÓN, DESARROLLADO PARA EL CENTRO TECNOLÓGICO DE LA INDIA DE LA FACULTAD DE INGENIERÍA, USAC”**.

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

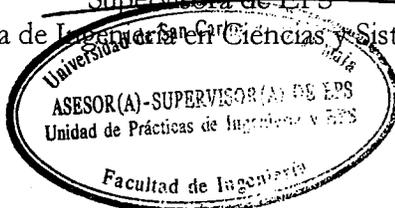
Sin otro particular, me es grato suscribirme.

Atentamente,

*“Id y Enseñad a Todos”*

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

FFAPdM/RA





Guatemala, 17 de febrero de 2011.  
REF.EPS.DOC.119.02.2011.

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

Estimado Ingeniero Perez Turck.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **“IMPLEMENTACIÓN DEL CURSO IPE BASADO EN LA METODOLOGÍA DEL ITCoE, PARA DESARROLLAR UNA MEJOR ABSTRACCIÓN DE LOS CONCEPTOS DE PROGRAMACIÓN, DESARROLLADO PARA EL CENTRO TECNOLÓGICO DE LA INDIA DE LA FACULTAD DE INGENIERÍA, USAC”**, que fue desarrollado por los estudiantes universitarios **Josué Rodolfo Obregón Velásquez** Carné No. **200412545**, **Sergio Antonio Villalta Montenegro** Carné No. **200412738** y **Alex Steven Campos Hernández** Carné No. **200412793** quienes fueron debidamente asesorados por la Inga. Gladys Aceituno y supervisados 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 de la Asesora 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 Zaccán de Serrano  
Directora Unidad de Prácticas de Ingeniería y EPS



NISZ/ra



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

Guatemala, 23 de Febrero de 2011

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 los estudiantes **ALEX STEVEN CAMPOS HERNÁNDEZ**, carné 2004-12793, **JOSUÉ RODOLFO OBREGÓN VELÁSQUEZ**, carné 2004-12545 y **SERGIO ANTONIO VILLALTA MONTENEGRO**, carné 2004-12738 titulado: "IMPLEMENTACIÓN DEL CURSO IPE BASADO EN LA METODOLOGÍA DEL ITCOE PARA DESARROLLAR UNA MEJOR ABSTRACCIÓN EN LOS CONCEPTOS DE PROGRAMACIÓN, DESARROLLADO PARA EL CENTRO TECNOLÓGICO DE LA INDIA DE LA FACULTAD DE INGENIERIA, USAC", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

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

Atentamente,

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



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

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado **“IMPLEMENTACIÓN DEL CURSO IPE BASADO EN LA METODOLOGÍA DEL ITCoE PARA DESARROLLAR UNA MEJOR ABSTRACCIÓN EN LOS CONCEPTOS DE PROGRAMACIÓN DESARROLLADO PARA EL CENTRO TECNOLÓGICO DE LA INDIA DE LA FACULTAD DE INGENIERÍA USAC”**, presentado por los estudiantes ALEX STEVEN CAMPOS HERNÁNDEZ, JOSUÉ RODOLFO OBREGÓN VELÁSQUEZ Y SERGIO ANTONIO VILLALTA MONTENEGRO, aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**

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



Guatemala, 26 de mayo 2011



DTG. 167.2011

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: **IMPLEMENTACIÓN DEL CURSO IPE BASADO EN LA METODOLOGÍA DEL ITCoE PARA DESARROLLAR UNA MEJOR ABSTRACCIÓN EN LOS CONCEPTOS DE PROGRAMACIÓN DESARROLLADO PARA EL CENTRO TECNOLÓGICO DE LA INDIA DE LA FACULTAD DE INGENIERÍA USAC**, presentado por los estudiantes universitarios **Alex Steven Campos Hernández, Josué Rodolfo Obregón Velásquez y Sergio Antonio Villalta Montenegro**, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Murphy Olimpo Paiz Recinos  
Decano

Guatemala, 31 de mayo de 2011.



/gdech



## **ACTO QUE DEDICO A:**

### **Alex Steven Campos Hernández**

- Dios**                    Que ha sido mi fiel amigo, mi escudo, mi fortaleza y mi salvador. Quien me da la felicidad, la alegría, el gozo, la paz en mi corazón para enfrentar mí diario caminar.
- Mis padres**            Erwin Campos y Fabiola Hernández, que han sido los pilares de mi vida, gracias por su apoyo incondicional en lo físico, económico y espiritual. Por sus consejos, correcciones y, principalmente, por el amor que me han dado a lo largo de toda mi vida.
- Mi hermano**            Jonnathan Campos, que ha sido compañero de batallas, aventuras, travesuras y un sinfín de emociones. Gracias por todos los consejos y disfrutar conmigo todos los momentos de mi vida, por ser mi hermano y amigo.
- Mis hermanas**           Jennifer y Sthefany Campos, que contagian de alegría y juventud mi caminar. Me hacen recordar la alegría del niño que todos llevamos dentro, el cual nos hace vivir con felicidad.
- Mi novia**                Nydia Balán, que con su amor y su cariño hizo que la culminación de esta etapa la gozara de una manera única y especial.

**Mis abuelos,  
tíos, primos  
y familia**

Que me apoyaron de distinta manera a lo largo de mi camino y, en especial, de esta meta que hoy alcanzo. Gracias por su cariño y amor incondicional, lo cual hace única a nuestra familia.

**Mis amigos**

Que estuvieron manifestándome de manera incondicional su ayuda y apoyo a lo largo de mi vida, que compartimos juntos alegrías y tristezas, pero sobre todo, de una gran amistad.

**Mis  
compañeros  
de EPS**

Josué Obregón y Sergio Villalta, mis amigos, con los cuales compartimos y experimentamos grandes momentos, que sin su ayuda y apoyo arduo no hubiera sido posible culminar esta meta.

## **Josué Rodolfo Obregón Velásquez**

**Dios** Por haberme dado vida, sabiduría y perseverancia en mis estudios, y que ahora me bendice permitiéndome alcanzar esta gran meta, en mi vida.

**Mi mamá** Por haber sido mamá y papá a la vez y siempre estar allí, dándome apoyo, amor, cuidado e impulso, para que me convierta en un hombre con valores.

**Mis hermanas y hermano** Porque al igual que mi mamá, siempre estuvieron allí para ayudarme de una u otra forma, y su amor de hermanos nunca faltó aunque estuvieran lejos.

**Mis abuelos, tíos y primos** Por haber compartido conmigo en mi niñez, y siempre preocuparse por mí. Gracias a todos.

**Mis amigos de colegio y de trabajo** Por haber compartido conmigo durante la mayor parte de mi vida, haberme brindado su amistad y contribuido de una u otra forma para que yo alcanzara este gran logro. Gracias a todos.

**Mis compañeros de EPS** Más que mis compañeros son mis amigos; y sin su ayuda y trabajo arduo, no hubiera logrado alcanzar esta meta.

## **Sergio Antonio Villalta Montenegro**

**Dios** Quien me ha permitido vivir y me ha rodeado de las personas necesarias para llegar a este momento clave de mi vida.

**Mis padres** Quienes me han brindado su amor a través de un apoyo total, fundamental e incondicional en el transcurso de mi vida personal y estudiantil.

**Mis hermanos** Con quienes he ido de la mano en el proceso de crecimiento dentro de mi familia y la sociedad.

**Toda mi familia** Por haber contribuido en mi formación y desarrollo como una persona con valores y principios.

**Mis amigos** Por el privilegio de contar con amistades eternas, así como nuevas amistades, que como todo lo bueno, necesitan del mejor de los cuidados. Qué suerte habernos encontrado en esta vida.

**Mis compañeros de EPS** Que mejor que dar este paso final, que con mis amigos.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	V
GLOSARIO.....	VII
RESUMEN .....	XIII
OBJETIVOS .....	XV
INTRODUCCIÓN .....	XVII
1 CREACIÓN DEL CURSO DE <i>IPE</i> ( <i>INTRODUCTION TO PROGRAMMING ENVIRONMENT</i> ).....	1
1.1 Justificación.....	1
1.2 Características del curso <i>IPE</i> .....	2
1.2.1 Alcance .....	3
1.2.2 Objetivo principal .....	3
1.2.3 Objetivos secundarios.....	3
1.2.4 Metas .....	4
1.3 Selección del material didáctico del curso de <i>IPE</i> .....	4
1.3.1 Programación Estructurada .....	5
1.3.2 Programación orientada a objetos .....	5
1.4 Definición de la guía del instructor para el curso <i>IPE</i> .....	7
1.4.1 Conceptos y fundamentos de la programación.....	8
1.4.1.1 Conceptos básicos de programación.....	8
1.4.1.2 Fundamentos de programación.....	9
1.4.2 Programación Estructurada .....	10
1.4.2.1. <i>Karel the Robot</i> .....	10
1.4.2.2 Instrucciones primitivas .....	12
1.4.2.3. Extendiendo el vocabulario de Karel.....	12

1.4.2.4	Extendiendo el vocabulario de Karel II.....	13
1.4.2.5	Ejecutando instrucciones condicionalmente.....	13
1.4.2.6	Instrucciones que repiten.....	14
1.4.2.7	Técnicas avanzadas de Karel.....	14
1.4.3	Programación Orientada a Objetos (POO) .....	15
1.4.3.1	Conociendo el paradigma de la POO.....	15
1.4.3.2	Alice.....	16
1.4.3.3	Diseño e implementación de un programa .....	17
1.4.3.4	Clases, objetos métodos y parámetros en Alice...18	
1.4.3.5	Uniendo todas las piezas.....	18
1.4.3.6	Clases, objetos métodos y parámetros en Alice...19	
1.4.3.7	Interactuando: eventos y manejo de eventos.....	20
1.4.3.8	Funciones y el <i>if/else</i> .....	20
1.4.3.9	Repetición: Ciclos definidos y condicionales.....	21
1.4.3.10	Repetición: Recursión.....	21
1.4.3.11	Listas y procesamiento de listas.....	22
1.5	Elaboración del material didáctico.....	23
1.6	Elaboración de la publicidad para el curso de <i>IPE</i> .....	24
1.7	Inicio del curso de <i>IPE</i> .....	25
1.8	Desarrollo del curso de <i>IPE</i> .....	25
1.9	Resultados del curso de <i>IPE</i> .....	26
1.10	Mejoras propuestas para el curso de <i>IPE</i> .....	30
1.11	Entrega del material didáctico del curso a las nuevas autoridades y personal administrativo del <i>ITCoE</i> .....	31
2	SEGUIMIENTO AL ESTUDIO DE LA METODOLOGÍA DE ENSEÑANZA DE TECNOLOGÍA DEL <i>ITCOE</i> .....	33
2.1	Antecedentes .....	33

2.2	Descripción del estudio .....	34
2.2.1	Fase de preparación .....	34
2.2.2	Fase de ejecución .....	36
2.2.3	Fase de evaluación .....	38
2.3	Resultados del estudio .....	38
2.3.1	Resultados de IPC1 .....	39
2.3.2	Resultados de IPC2 .....	42
2.4	Factores que influyeron negativamente en estudio .....	45
2.5	Recomendaciones del estudio .....	46
3	APOYO ADMINISTRATIVO PRESTADO AL <i>ITCOE</i> .....	47
3.1	Estructura organizacional .....	47
3.2	Administración de cursos .....	48
3.2.1	Generación de nuevos cursos .....	53
3.2.2	Pruebas de conocimiento del idioma inglés .....	53
3.2.3	Sondeo sobre preferencias para abrir cursos nuevos.....	54
3.2.4	Creación de instructivos de procedimientos básicos del <i>ITCoE</i> .....	54
3.2.5	Préstamo de los laboratorios a la Rectoría de la USAC .....	55
3.2.6	Préstamo de los laboratorios para el Congreso de Matemáticas .....	55
3.2.7	Préstamo de los laboratorios para COECYS .....	55
3.2.8	Instalación antivirus .....	56
3.2.9	Limpieza del equipo de los laboratorios del <i>ITCoE</i> .....	56
3.2.10	Reparación de la puerta del Laboratorio 2.....	57
3.2.11	Traducción de las diapositivas del curso ISE al idioma español .....	57
3.2.12	Capacitación del nuevo equipo de trabajo del <i>ITCoE</i> .....	57
3.2.13	Reuniones administrativas y de seguimiento .....	58

3.2.14 Apoyo administrativo en el cambio de autoridades del <i>ITCoE</i> .....	58
3.2.15 Utilización de herramienta para evaluaciones de cursos .....	59
CONCLUSIONES.....	61
RECOMENDACIONES .....	65
BIBLIOGRAFÍA .....	67
APÉNDICES .....	69
ANEXOS .....	111

## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Porcentaje de estudiantes aprobados y reprobados en <i>IPE</i> sección A (matutina) de la 1era iteración .....	28
2.	Porcentaje de estudiantes aprobados y reprobados en <i>IPE</i> sección B (vespertina) de la 1era iteración .....	30
3.	Porcentaje de estudiantes aprobados y reprobados en IPC1 dentro del <i>ITCoE</i> en el segundo semestre 2009 .....	40
4.	Porcentaje de aprobación del curso de IPC1 .....	41
5.	Porcentaje de estudiantes aprobados y reprobados en IPC2003 dentro del <i>ITCoE</i> en el segundo semestre 2009 .....	43
6.	Porcentaje de aprobación del curso de IPC2 .....	44
7.	Estructura organizacional del <i>ITCoE</i> .....	48
8.	Comparación de las cantidades de los estudiantes que completaron con los que desertaron .....	50
9.	Tendencia del porcentaje de aprobación y del PI mensual .....	51
10.	Tendencia del porcentaje de utilización de instructores y laboratorios ..	53

### TABLAS

I.	Resultados estadísticos de <i>IPE</i> sección A (matutina) .....	27
II.	Resultados estadísticos de <i>IPE</i> sección B (vespertina) .....	29
III.	Resultados estadísticos de IPC1 en el <i>ITCoE</i> .....	39
IV.	Resultados Estadísticos de IPC2 .....	42
V.	Cursos disponibles durante el segundo semestre de 2009 .....	49

VI. Total de estudiantes que se inscribieron en los cursos, los que completaron y los que desertaron .....	50
VII. Porcentaje de aprobación e indicador de rendimiento de cada mes ....	51
VIII. Porcentaje de utilización de recursos (instructores y laboratorios) durante cada mes .....	52

## GLOSARIO

<b><i>Assignment</i></b>	Evaluación de tipo teórico y práctica, la cual es efectuada al estudiante sobre los contenidos de un curso específico dentro del <i>ITCoE</i> .
<b><i>CentreManager</i></b>	Persona a cargo de la dirección y administración dentro del <i>ITCoE</i> .
<b>COECYS</b>	Congreso de Estudiantes de la Escuela de Ciencias y Sistemas.
<b><i>CoursesAdministrator</i></b>	Persona a cargo de la revisión, seguimiento y control de los contenidos de los cursos impartidos. Es responsable de la mejora continua del material didáctico del <i>ITCoE</i> .
<b><i>Courseware</i></b>	Documento bibliográfico de apoyo para el estudiante de un curso en específico. Generalmente, los estudiantes tienen acceso a la versión impresa y la versión digital es utilizada por el personal del <i>ITCoE</i> .

<b><i>DropOut</i></b>	Número de deserciones de estudiantes aplicado a un curso específico.
<b>ECCSS</b>	Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala.
<b><i>EvaluationForms</i></b>	Formulario utilizado para tener un control de asistencia, punteo y conformidad de nota del estudiante en relación a una tarea o examen.
<b><i>FeedbackSession</i></b>	Sesión donde el Instructor Técnico da una retroalimentación a los estudiantes de forma individual sobre las evaluaciones realizadas hasta el momento.
<b><i>FinalModuleTests</i></b>	Pruebas finales de carácter teórico para cada módulo de un curso en específico.
<b><i>Handbook</i></b>	Documento que guía al instructor en las sesiones prácticas de un curso. Este contiene ejemplos resueltos y ejercicios propuestos para ser utilizados durante las sesiones de laboratorio.
<b><i>Homeworks</i></b>	Tareas con complejidad baja, son asignadas a los estudiantes de un curso para que las resuelvan en sus casas. Estas tareas son preparatorias para los <i>Assignment</i> y refuerzan el contenido visto, tanto en las clases teóricas como prácticas.

<b><i>Instructor</i></b>	Documento que utiliza un Instructor para impartir un curso. El
<b><i>Guideline (IG)</i></b>	documento contiene en detalle los objetivos, pre requisitos, módulos, contenidos, planificación, ponderación, criterio de evaluación e instrucciones de ejecución de las actividades a realizar para un curso específico.
<b><i>IPC1</i></b>	Introducción a la Programación y Computación 1. Curso de la carrera de Ingeniería en Ciencias y Sistemas.
<b><i>IPC2</i></b>	Introducción a la Programación y Computación 2. Curso de la carrera de Ingeniería en Ciencias y Sistemas.
<b><i>ITCoE</i></b>	<i>Information Technology Education Centre of Excellence</i> . En español, Centro de Excelencia en Educación de Tecnologías de la Información.
<b><i>Malware</i></b>	Tipo de <i>software</i> que tiene como objetivo infiltrarse o dañar una computadora sin el consentimiento de su propietario.
<b><i>Module Breakup Form</i></b>	Forma donde se define la organización de los módulos de un curso. Se utiliza dentro del IG, ya que especifica el número de sesiones teóricas y prácticas para cada uno de los módulos que conforman un curso.

<b>Module Evaluation Form</b>	Es la forma que describe la ponderación de cada actividad para cada uno de los módulos que conforman un curso. Forma parte del contenido de un IG.
<b>PI</b>	<i>Performance Indicator</i> . Indicador de rendimiento de los cursos, que está entre 0 y 5, en donde cantidades cercanas a 5 indican que en el curso no solo han salido aprobados los estudiantes, sino que también han obtenido buenas calificaciones.
<b>Self Study Assignments</b>	Tareas con alta carga de auto estudio que sirven de refuerzo de los contenidos vistos en clase y laboratorio de un curso específico en el <i>ITCoE</i> .
<b>Seminars</b>	Seminarios donde el estudiante investiga un tema relacionado con el curso y posteriormente exponen sus conclusiones acerca del mismo.
<b>Session</b>	Clase presencial impartida por el instructor de un curso en específico dentro de las instalaciones del <i>ITCoE</i> .
<b>Session Breakup</b>	Muestra detalladamente el contenido y planificación de todas las sesiones teóricas y prácticas que conforman un curso en específico.
<b>Session Detail Form</b>	Forma donde se muestra la programación de sesiones teóricas y prácticas para cada uno de los módulos que conforman un curso dentro del <i>ITCoE</i> .
<b>Short Tests</b>	Pruebas cortas de complejidad media efectuadas a los

estudiantes de un curso. Se utilizan para medir el grado de comprensión de un tema específico dentro del curso.

**Technical Trainer** Instructor de conocimientos técnicos, encargado de impartir los cursos en el *ITCoE*. En español Instructor Técnico.

**Technical Trainer Reports** Reportes efectuados mensualmente por cada instructor del *ITCoE*. Este reporte contiene la información de notas, ejecución, deserciones, porcentaje de asistencia y porcentaje de rendimiento de los alumnos para cada uno de los cursos que son impartidos por el *Technical Trainer*.

**TI** Tecnologías de información.

**TIC** Tecnologías de información y comunicaciones.

**Trainer Administrator** Persona a cargo de capacitación, control y seguimiento de los *Technical Trainers* del *ITCoE*.

**Overflow** Tiempo en horas que describe el atraso de la ejecución de un curso acorde a la planificación inicial.

**Quality Administrator** Persona a cargo de las tareas de actualización, creación y mejora de los procesos del *ITCoE*. Es el encargado de la mejora continua de los procesos del *ITCoE*.



## RESUMEN

El mundo de las tecnologías de información y comunicación cambia rápida y constantemente. Para los individuos involucrados en este ámbito, adaptarse a este constante cambio requiere de un esfuerzo considerable. Este esfuerzo depende de los recursos que las personas tengan disponibles ya sea tanto a niveles empresariales, así como, a niveles educativos.

El proceso de enseñanza-aprendizaje de estas tecnologías debe ser lo suficientemente maduro y adaptarse lo mejor posible a este cambio constante, para brindar a los estudiantes una enseñanza acorde a las necesidades de la actualidad en que se encuentren.

En el área de ciencias de la computación dentro del proceso de enseñanza-aprendizaje, un elemento muy importante es la abstracción de los conceptos, el cual es vital para una comprensión adecuada de los conceptos básicos de programación que son la base para encontrar soluciones de tipo informáticas y que, posteriormente, serán plasmados en líneas de código fuente.

En el año 2006 el gobierno del país de la India apoyando al sector tecnológico de Guatemala, realizó la donación del India-Guatemala *IT Centre of Excellence*, el cual fue ubicado en la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala. Este centro se especializa en la capacitación de la población en general, en las tecnologías de la información y comunicación.

Este centro provee a la Facultad de Ingeniería, una metodología de enseñanza-aprendizaje que describe una serie de buenas prácticas que mejoran la calidad de la enseñanza. Éstas están contenidas dentro de un esquema rígido que permite llevar un control adecuado del desarrollo de los diferentes cursos que se imparten.

Con base en esta metodología se propone como proyecto de graduación la implementación de un curso que introduzca los conceptos básicos de programación, para quienes tengan interés en esta área. El curso tiene como característica principal contar con herramientas que se basan en animaciones 2D y 3D para agilizar y mejorar el desarrollo de la abstracción de los conceptos básicos de programación. También, como parte del proyecto, se encuentra el apoyo brindado tanto en el área administrativa, así como, en el área didáctica, además de dar seguimiento a proyectos desarrollados en dicho centro.

Los resultados de este proyecto se describen en el presente informe final.

## OBJETIVOS

### General

Crear las herramientas tecnológicas necesarias utilizando la metodología de enseñanza del *ITCoE*, para desarrollar una mejor abstracción de los conceptos básicos de programación.

### Específicos

1. Implementar el curso *Introduction to Programming Environment* aplicando la metodología utilizada en el *ITCoE* y sobre la base de herramientas de animaciones 2D y 3D.
2. Investigar las dos herramientas elegidas para el desarrollo del curso para estructurar bien los contenidos de los dos módulos en que se ha dividido el mismo.
3. Generar, con base en el análisis de las dos herramientas, la guía del instructor, la cual contendrá los aspectos generales de planificación de los dos módulos principales, así como, la orientación del desarrollo adecuado del curso.

4. Elaborar el material didáctico para los dos módulos principales en los que se divide el curso. El material tendrá versión español e inglés.
5. Realizar la primera iteración del curso *IPE* para obtener retroalimentación por parte de las personas que lo reciban y proponer mejoras al mismo.
6. Dar seguimiento al proyecto de adaptación de la metodología del *ITCoE* en los cursos de la Escuela de Ciencias y Sistemas, a través de la realización de la tercera iteración del curso “Introducción a la Programación y Computación 1” y la segunda iteración del curso “Introducción a la Programación y Computación 2”.
7. Apoyar el área administrativa del *ITCoE*, por medio de la elaboración del calendario de cursos, generación de reportes, mantenimiento de laboratorios, entre otras actividades.
8. Impartir cursos que se encuentren dentro del plan de estudios oficial del Centro como apoyo al área didáctica del *ITCoE*

## INTRODUCCIÓN

La información actualmente, es parte vital para el desarrollo del mundo. Esta información es manejada por sistemas informáticos cada vez más grandes y complejos. Los profesionales, a lo largo de su carrera, deben tener una abstracción de los conceptos que manejan estos grandes sistemas para comprender su funcionalidad y poder trabajar con ellos.

La abstracción es aquella que ayuda a entender y visualizar el funcionamiento de los elementos por medio de conceptos claves y básicos, tales como: variable, función, procedimiento, operador, estructura de control, entre otros; dejando a un lado los detalles técnicos. Por ello, los profesionales deben contar con una abstracción adecuada que haya sido desarrollada durante su formación académica, así como, en su experiencia laboral, con el objetivo que puedan hacer frente a las situaciones que involucren este aspecto.

En ese sentido, una solución para lo anterior, es la creación de un curso introductorio basado en herramientas que generan animaciones 2D y 3D para que la persona pueda aprender jugando e interaccionando con el entorno y de esta forma asimilar, con mayor facilidad, los conceptos que se le presentan y así desarrollar su abstracción.

Con el desarrollo de la abstracción, se pretende alcanzar una mayor comprensión de los contenidos, siendo éste un aspecto muy importante, sobre todo, en asignaturas más avanzadas de una carrera de informática.

El curso se encuentra estructurado de tal forma que ayuda a estimular al estudiante a seguir aprendiendo, así como a involucrarse más en los temas que se desarrollan en el mismo, despertando en él habilidades de investigación, indagación, estudio y análisis, las cuales se integran al proceso de aprendizaje y vuelven al alumno más proactivo.

En el presente trabajo se presenta el desarrollo de este curso que lleva por nombre *IPE*, éste fue desarrollado en el *ITCoE* que se encuentra en la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, con base en la metodología de enseñanza-aprendizaje que brinda ésta al centro de capacitación en tecnología de la información.

# **1 CREACIÓN DEL CURSO DE *IPE* (*INTRODUCTION TO PROGRAMMING ENVIRONMENT*)**

## **1.1 Justificación**

El nivel de educación media en Guatemala está sufriendo una etapa de grandes deficiencias, las cuales se ven reflejadas al momento en que la población estudiantil ingresa y cursa la educación superior. Dichas deficiencias se observan con gran notoriedad en el área de las Ciencias de la Informática, debido a que es un área relativamente nueva en el ámbito guatemalteco y que sólo un pequeño sector de la población puede tener acceso a la misma.

Debido a este problema, se encuentra una gran brecha entre la educación media y la educación superior con respecto a las ciencias de la informática; misma que llega a sentir el estudiante cuando empieza a cursar la carrera de ciencias y sistemas.

En vista de este problema que posee el estudiante, se planteó elaborar un curso que contribuya a eliminar dicha brecha, además de reforzar el contenido base de la carrera de Ciencias y Sistemas, enfocándose tanto en la teoría como en la práctica, ya que ambas áreas necesitan refuerzo.

Este curso lleva por nombre *IPE Introduction to Programming Environment* (Introducción al ambiente de Programación) el cual contiene un esquema de los temas principales a reforzar en los estudiantes que ingresan por primera vez a la carrera de Ciencias y Sistemas, para que de esta forma complementen sus conocimientos básicos.

El curso de *IPE* se estructuró con base en la metodología que se utiliza en el *ITCoE*, la cual está basada en buenas prácticas implementadas por los empleados de *TATA Consultancy Services*. Esta metodología ha tenido buenos resultados dentro de la Universidad de San Carlos de Guatemala, específicamente en el área de la Informática, en cursos orientados a la carrera de Ciencias y Sistemas.

## **1.2 Características del curso *IPE***

El curso de *IPE* está diseñado para reforzar o generar los conocimientos base al estudiante de primer ingreso de la carrera de ciencias y sistemas. Esto debido a que se ha percibido en los últimos años, que algunos alumnos que inician esta licenciatura, no poseen, en su totalidad, los conocimientos básicos y necesarios para obtener buenos resultados, ya que se da por hecho que dichos conocimientos ya se poseen. Esta situación es uno de los factores que afectan al educando y provoca que este fracase en los primeros cursos de la carrera.

### **1.2.1 Alcance**

El alcance del curso de *IPE* en su primera iteración, será impartido a cualquier persona que tenga la iniciativa y el interés de querer aprender los términos básicos del ambiente de programación. De este conjunto de individuos, se seleccionará un grupo piloto para evaluar el proceso enseñanza aprendizaje y tomar decisiones sobre los cambios que pueda tener la estructura del curso o la metodología de enseñanza, para mejorarlo.

### **1.2.2 Objetivo principal**

Brindar y reforzar los conceptos básicos de programación que son necesarios en la carrera de ciencias y sistemas.

### **1.2.3 Objetivos secundarios**

- Crear buenas prácticas y hábitos de programación al estudiante
- Incentivar al estudiante a ser proactivo
- Adentrar al estudiante con los conceptos claves de programación
- Vincular los conocimientos adquiridos por medio de la teoría a través de la práctica

#### **1.2.4 Metas**

- Crear un curso introductorio para los cursos de IPC1 e IPC2, con el propósito de fortalecer los conocimientos adquiridos por el estudiante en los niveles precedentes del sistema educativo nacional.
- Estructurar un curso que sea una conexión entre la formación media que trae el estudiante con la educación superior, para que de esta forma pueda adaptarse al nuevo sistema de educación profesional.
- Aplicar la metodología de enseñanza utilizada dentro del *ITCoE* para la creación de este curso.
- Dejar a la disposición del *ITCoE* un curso que contribuya a satisfacer las necesidades de conocimiento sobre introducción básica del ambiente de programación, así como, un material didáctico acorde a dichas necesidades.

#### **1.3 Selección del material didáctico del curso de *IPE***

El material didáctico que se seleccionó fue orientado a los dos módulos principales del curso de *IPE*: programación estructurada y programación orientada a objetos. Se realizó una búsqueda de material didáctico que estuviera bien respaldado con un historial de buenos resultados, así como, orientado a un nivel de aprendizaje básico, es decir, lo más claro posible para que cualquier estudiante lo pudiese comprender no importando sus conocimientos previos.

### **1.3.1 Programación Estructurada**

Para el área de programación estructurada, se encontró el material didáctico de “*Karel the Robot*”, creado por Richard E. Pattis, un catedrático de ciencias informáticas de la Universidad de Carnegie Mellon y de la Universidad Irvine de Carolina en los Estados Unidos de América. “*Karel the Robot*” es una aplicación que cuenta con su propio lenguaje de programación el cual se enfoca en el aprendizaje de la programación estructurada, a un nivel bastante básico y simple. Este consta de un simulador integrado, que sirve para poder ejecutar instrucciones de su propio lenguaje de programación, llamado Karel+.

“*Karel the Robot*” consiste en una aplicación orientada a la educación, su objetivo principal es enseñar el paradigma de programación estructurado a través del manejo de un robot llamado “Karel” dentro de un mundo de dos dimensiones. Dentro de este mundo puede interactuar con calles, avenidas, paredes y beepers, todo esto utilizando su propio lenguaje, Karel +.

### **1.3.2 Programación orientada a objetos**

Para el área de la programación orientada a objetos, el material didáctico que se seleccionó fue “Alice” de la Universidad de Carnegie Mellon. “Alice” es un lenguaje de programación orientado a objetos el cual es educativo, libre y abierto. Este posee su propio entorno de desarrollo integrado (IDE, por sus siglas en inglés), está basado en el lenguaje de programación JAVA y es multiplataforma, lo cual significa que puede ejecutarse en cualquier sistema operativo. Fue diseñado por los investigadores de la Universidad de Carnegie Mellon encabezados por Randy Pausch.

Randy Pausch fue un ingeniero en Informática, que impartió clases en la Universidad de Carnegie Mellon de Pittsburgh en los Estados Unidos de América, además escribió una buena cantidad de libros, consiguiendo la fama por su trabajo "*The Last Lecture*". El 18 de septiembre de 2007, dio discurso titulado "*Really Achieving Your Childhood Dreams*" que impactó a muchas personas por su mensaje de seguir nuestros sueños de la infancia. En agosto del 2006 se le diagnosticó cáncer de páncreas. Él luchó para poder curar su mal, sin embargo, el cáncer lo tenía demasiado avanzado y fallece el 25 de Julio de 2008.

El programa de Alice se desarrolló prioritariamente para solucionar los tres problemas fundamentales del *software* educativo:

- a. La mayoría de los lenguajes de programación están diseñados para producir otros programas cada vez más complejos. Alice está diseñado únicamente para enseñar a programar;
- b. Alice está íntimamente unido a su IDE. No hay que recordar ninguna sintaxis especial. De todas formas, acepta tanto el modelo de programación orientada a objetos como la dirigida a eventos;
- c. Alice está diseñada para el público que normalmente no se enfrenta a problemas de programación, tales como: alumnos de secundaria, mediante un sistema de «arrastrar y soltar».

En estudios realizados en el *Ithaca College* y en *Saint Joseph's University*, las notas medias de estudiantes sin experiencia de programación y en su primer curso de informática subieron de suficiente a notable y su retención del 47% al 88%. La nueva versión del *software* de Alice está siendo elaborado de manera conjunta con la empresa de *Electronic Arts* y los desarrolladores de la Universidad de Carnegie Mellon, ya que *Electronics Arts* utilizará los modelos de los personajes en el juego titulado, “*The Sims 2*”.

#### **1.4 Definición de la guía del instructor para el curso IPE**

El *Instructor Guideline* o IG es una guía esquematizada de un curso específico dentro del *ITCoE*. Cada curso dentro del *ITCoE* posee un IG el cual contiene todas las actividades y temas debidamente esquematizados y calendarizados, para que de esta forma el monitor lo tome como guía para poder impartir el curso. Este documento le sirve al administrador del *ITCoE* para identificar el estado del curso, si se encuentra a tiempo o si posee alguna demora conforme a lo calendarizado.

El IG del curso de *IPE* está dividido en 3 módulos principales los cuales se enfocan en dos áreas, el de la teoría y el de la práctica. Esto debido a que en la metodología que se encuentra definida dentro del *ITCoE*, primero se debe impartir una clase teórica para que el estudiante adquiera el conocimiento y lo pueda estudiar y, posterior a ello, pasar a la práctica para reafirmar dichos conocimientos. De esta forma el educando asimila la información y pone en práctica todo lo aprendido.

Los módulos principales del IG de *IPE* están orientados de manera que el estudiante vaya capacitándose de los conceptos básicos a los más complejos. Los módulos que conforman el IG de *IPE* son los siguientes:

### **1.4.1 Conceptos y fundamentos de la programación**

Este módulo sirve como introducción al curso porque posee conceptos bastante básicos los cuales ayudan al estudiante a formar una base que le servirá para los módulos subsiguientes. Además de contener conceptos, también aclara mucha jerga utilizada en el ambiente de programación, la cual podría causar confusión en el alumno al momento de utilizarla en el curso. El esquema de este es el siguiente:

#### **1.4.1.1 Conceptos básicos de programación**

Como primer punto se abarcan conceptos, tales como: programación, programa, algoritmo, pseudocódigo, etc. Dichos conceptos se listan a continuación:

- ¿Qué es programación?
- Programa
- Algoritmo
- Lenguaje de programación
- Pseudocódigo
- Abstracción
- Paradigma/Paradigma de programación
- Tipos de lenguajes de programación

- Según el nivel de Abstracción
  - Lenguaje de máquina
  - Lenguaje de bajo nivel
  - Lenguaje de medio nivel
  - Lenguaje de alto nivel
- Según el paradigma de programación que posean
  - Imperativo
  - Funcional
  - Lógico
  - Orientado a objetos
- Proceso de compilación
  - Definición de compilación
- Tipos de errores
  - Léxicos
  - Sintácticos

#### **1.4.1.2 Fundamentos de programación**

Este temase enfoca en los términos y jerga de programación, la cual le será de gran ayuda al educando para comprender los módulos. Y en la última sección se tratan conceptos de operadores y de expresiones aritméticas, que son temas base en toda programación. Los temas son los siguientes:

- Programa
  - Variable
  - Constante
  - Tipo de dato

- Identificador
- Palabra clave
  
- Operadores
  - Aritmético
  - Relacional
  - Lógico
  
- Expresiones
  - Aritméticas
  - Precedencia de operadores

#### **1.4.2 Programación Estructurada**

Este módulo se enfoca en el paradigma de programación estructurada, el cual se usa, generalmente, como base para empezar a programar ya que se dan a conocer las estructuras de control básicas para la programación: secuenciales, selectivas o condicionales y repetitivas.

Este módulo se basa en el material didáctico de “*Karel, the robot*” el cual ayuda a fomentar, en el estudiante, los conceptos básicos del paradigma de programación estructurada de una forma dinámica y muy entretenida. El esquema de este módulo es el siguiente:

##### **1.4.2.1 *Karel the Robot***

El primer punto abarcado en este módulo es el conocimiento del *software* de “*Karel the robot*”, se explica los principales componentes y distintos paneles

que posee la aplicación, para que de esta forma el alumno se familiarice con la herramienta a utilizar. Los conceptos que se abarcan, son los siguientes:

- Conociendo el entorno de Karel
  - o Componentes principales
    - Karel
    - Cursor
    - Pared
    - *Beeper*
    - Calles y avenidas
  - o Paneles
    - Panel del mundo inicial
      - *Walls*
      - *Beepers*
      - *World File*
      - *Karel's Initial Situation*
      - Cursor
      - *Load, Save and Clear World*
    - Panel del Programa
      - *Open Program*
      - *New Program*
      - *Edit Program*
      - *Save Program*
      - *Compile*
      - *Save As...*
      - *Print Program*
      - *Abort Changes*

- Panel de Ejecución
  - *Step*
  - *Unstep*
  - *Run*
  - *Initialize*
  - *Steps/Min*
  - *Beepers in Bag*
  - *Next Instruction*
  - *World Low*
  - *Inside*
  - *Stack*
  
- Ejemplo de un Programa en Karel

### 1.4.2.2 Instrucciones Primitivas

Estas son las instrucciones base para ejecutar y elaborar instrucciones más complejas dentro de Karel, estas manejan el accionar básico del robot Karel, tales como: cambiar de posición, manejo de beepers, terminar una tarea y manejo de errores, dentro de del entorno 2D de la aplicación. Las instrucciones abarcan los siguientes puntos:

- Cambiando de posición
- Manejando los Beepers
- Terminando una tarea
- Mi primer programa en Karel
- *Error-Shutoffs*
- Errores de programación

### **1.4.2.3 Extendiendo el vocabulario de Karel**

Este contenido programático inicia con el estudio del vocabulario de Karel para poder realizar el primer programa formal de Karel, de esta manera se empieza a aplicar todo lo aprendido, hasta el momento, dentro del curso. Karel facilita la tarea debido a que posee un lenguaje de programación más natural, situación que contribuye a que el estudiante comprenda conceptos más complejos por medio de un lenguaje más humano. El contenido se lista a continuación:

- Creando un lenguaje de programación más natural
- Un mecanismo para definir nuevas instrucciones
- Estructurando bloques
- Lo correcto y el significado de nuevas instrucciones
- Definiendo nuevas instrucciones en un programa

### **1.4.2.4 Extendiendo el vocabulario de Karel II**

En este tema se amplía el vocabulario de Karel, utilizando técnicas en el diseño de los programas para que de esta forma se pueda construir código más entendible y ordenado. El contenido del tema se lista a continuación:

- *Boxing*: cómo Karel entiende un programa
- Un programa no gramatical
- Herramientas para diseñar y escribir programas en Karel
- Ventajas de utilizar nuevas instrucciones
- Escribiendo programas entendibles

### 1.4.2.5 Ejecutando instrucciones condicionalmente

En este apartado se tratan las instrucciones con condición, las cuales realizan una acción dependiendo de la situación en que se encuentre Karel dentro del entorno 2D en el cual se maneja. También veremos la manera de optimizar estas instrucciones anidándolas unas con otras o simplificándolas, para una mejor ejecución. El contenido se lista a continuación:

- La instrucción *IF/THEN*
- Condiciones conocidas
- La instrucción *IF/THEN/ELSE*
- Instrucciones *IF* anidadas
- Cuándo utilizar una instrucción *IF*
- Transformaciones para simplificar una instrucción *IF*
- Un *ELSE* indeciso

### 1.4.2.6 Instrucciones que repiten

En este tema se verá las instrucciones repetitivas llamadas también ciclos o bucles, las cuales permiten repetir una o un conjunto de instrucciones una cantidad de veces, así como, la manera de optimizar el uso de dichas instrucciones. Las instrucciones que se abarcan se listan a continuación:

- La instrucción *ITERATE*
- La instrucción *WHILE*
- Por qué el *WHILE* es necesario
- Errores a evitar con el ciclo *WHILE*
- Ciclos *WHILE* anidados
- Instrucciones *WHILE* e *IF*

- Analizando los ciclos

#### **1.4.2.7 Técnicas Avanzadas de Karel**

Como último punto de la programación estructurada se verán temas más complejos, como lo son: la recursión, las búsquedas y una aplicación de estos términos con el concepto de “haciendo aritmética”, el cual recaba toda la información adquirida para poder realizar esta actividad, éste se divide en:

- Recursión
- Búsquedas
- Haciendo aritmética

#### **1.4.3 Programación Orientada a Objetos(POO)**

Este módulo se enfoca en el paradigma de programación orientado a objetos, el cual es uno de los paradigmas que más se utiliza en el medio de programación contemporáneo. La mayoría de lenguajes modernos están basados en este paradigma, debido a su abstracción y versatilidad para la creación de aplicaciones modernas.

Este se basa en el material didáctico de “Alice”, el cual ayuda a fomentar los conceptos básicos del paradigma de programación en el estudiante de una forma dinámica y muy entretenida. El esquema para este módulo, es el siguiente:

##### **1.4.3.1 Conociendo el paradigma de la POO**

El primer contenido programático de este módulo es una breve reseña histórica del paradigma orientado a objetos (POO), además, cómo fueron

evolucionando las ideas dentro del ambiente de programación hasta llegar a concebir dicho paradigma. También se tratarán los conceptos básicos del paradigma orientado a objetos (POO), tales como: objeto, clase, herencia, método, evento, de igual forma, las características del paradigma: abstracción, encapsulamiento, polimorfismo y herencia. El contenido se lista a continuación:

- Historia
- ¿Qué es un objeto?
- Conceptos fundamentales
- Objeto
- Clase
- Herencia
- Método
- Evento
- Características
- Abstracción
- Encapsulamiento
- Polimorfismo
- Herencia

#### **1.4.3.2 Alice**

El segundo punto del módulo es la introducción de la aplicación de “Alice”, es decir, el material didáctico base para el área de la programación orientada a objetos. También se tratarán las nociones fundamentales que rodean el concepto de la aplicación, tales como: un mundo virtual, objetos y modelos 3D, dimensiones, así como, las características propias de un objeto, como lo son su centro, distancia, posición y animación. Los temas son los siguientes:

- Historia
- ¿Por qué “Alice”?
- Conceptos fundamentales
- Mundo virtual
- Objetos y modelos 3D
- 3 dimensiones y 6 direcciones
- Centro de un objeto
- Distancia
- Posición
- Animación

#### **1.4.3.3 Diseño e implementación de un programa**

En el tercer punto del módulo, se empezará a tratar temas relacionados con el diseño e implementación de un programa en Alice, tales como: la elaboración de escenarios y guiones que son los esquemas principales para elaborar un programa dentro de Alice. También se realizará el primer programa dentro de la aplicación de Alice traduciendo el guión a código, para ello se utilizará acciones, manejando errores o *bugs*, propiedades y comentarios. El contenido se lista a continuación:

- Escenarios y guiones
  - o Conoce el escenario
  - o Diseño
    - Guiones visuales
    - Guiones textuales
- Mi primer programa
  - o ¿Qué es un programa?

- Traduciendo el guión a código del programa
- Acciones secuenciales vs. acciones simultáneas
- *Bugs*
- Usando propiedades
- Comentarios

#### **1.4.3.4 Clases, objetos, métodos y parámetros en Alice**

En este contenido programático se empezará a enseñar funciones y expresiones dentro de Alice, con el objeto de iniciar a interactuar dentro del ambiente 3D. También se verán, dentro de este módulo, las estructuras de control de ejecución condicional y los bucles o ciclos. El contenido se lista a continuación:

- Definición clase, objeto y método.
- Método a nivel global
  - Métodos
  - Creando mis propios métodos
  - Invocando un método
- Parámetros
  - Parámetros
  - Objetos como parámetros
  - Múltiples parámetros
- Métodos a nivel de clase y herencia
  - Métodos a nivel de clase
  - Creando una nueva clase
  - Herencia

### **1.4.3.5 Uniendo todas las piezas**

En este contenido del programa se verán los temas de funciones y expresiones, que de manera conjunta con las estructuras de control, ayudan a realizar programas con más interacciones dentro del mundo 3D de Alice. El contenido abarcado es el siguiente:

- Funciones y expresiones predefinidas
  - o Funciones
  - o Expresiones
  
- Estructuras de control
  - o Ejecución condicional
  - o La necesidad de repeticiones

### **1.4.3.6 Clases, objetos, métodos y parámetros en Alice**

Este es el sexto punto del módulo, en este se empezará a ver los conceptos de la programación orientada a objetos pero desde el ambiente de Alice, para que de esta forma el estudiante comprenda los conceptos anteriormente adquiridos y los lleve a la práctica dentro de la aplicación de Alice. El contenido es el siguiente:

- Definición clase, objeto y método.
- Método a nivel global
  - o Métodos
  - o Creando mis propios métodos
  - o Invocando un método
- Parámetros

- Parámetros
- Objetos como parámetros
- Múltiples parámetros
  
- Métodos a nivel de clase y herencia
  - Métodos a nivel de clase
  - Creando una nueva clase
  - Herencia

#### **1.4.3.7 Interactuando: eventos y manejo de eventos**

En esta fase del módulo se orienta al estudiante al tipo de programación utilizada en Alice, la cual es de manera interactiva, debido a que dentro del entorno 3D de Alice se necesita de un flujo de control de actividades y de eventos que desencadenan una cantidad determinada de acciones. Además se da a conocer el concepto y el uso de los métodos al interactuar con los eventos. El contenido es el siguiente:

- Programación interactiva
  - Flujo de control
  - Eventos
  - Métodos para manejar eventos
  - Métodos que responden a eventos
  - Uniendo eventos con métodos

#### **1.4.3.8 Funciones y el *if/else***

Este contenido trata de las funciones de condición pero desde el punto de vista del paradigma orientado a objetos. Los conceptos son los mismos que en

la programación estructurada sin embargo se manejan de forma diferente dentro de la aplicación 3D de Alice, de esta forma el estudiante podrá observar, de mejor forma, la diferencia que existe de un concepto de un paradigma a otro. El contenido se lista a continuación:

- Funciones
  - o Usando funciones predefinidas
  - o Escribiendo mis propias funciones
  - o La sentencia de retorno
  
- Control de ejecución con el *if/else*
  - o Utilizando funciones Booleanas
  - o Una expresión con múltiples condiciones
  - o *If* anidados

#### **1.4.3.9 Repetición: Ciclos definidos y condicionales**

En esta parte del módulo se verán los temas de bucles o ciclos de manera normal y anidada, contenido, que al igual que el anterior, le brindará al estudiante las diferencias que hay de un paradigma a otro manejando un mismo concepto de programación, cómo interactúa y cómo se comporta dentro del mundo 3D de Alice. El contenido es el siguiente:

- Ciclos definidos
  - o Ciclos anidados
  - o Ciclos infinitos
  
- *While* – Ciclo condicional

### **1.4.3.10 Repetición: Recursión**

En este tema el estudiante podrá aprender la recursión en el ambiente 3D de Alice, para que de esta manera el propio estudiante pueda ver la relación que existe entre un paradigma y otro. El contenido de este tema se lista a continuación:

- Introducción a la recursión
- Otro tipo de recursión

### **1.4.3.11 Listas y procesamiento de listas**

En este contenido programático se verán los temas de listas y búsquedas dentro del paradigma orientado a objetos. Estos temas requieren una mayor abstracción debida a que el procesamiento (creación e iteración) de listas en la práctica, son más complejos. Este tema abarca los siguientes puntos:

- Listas
  - o Creando una lista
  - o Iterando secuencialmente
  - o Iterando simultáneamente
- Búsquedas
  - o Utilizando funciones Booleanas
  - o Una expresión con múltiples condiciones

De esta forma es como están estructurados los temas dentro del IG (*instructor guideline*) del curso de *IPE*, este contenido será distribuido en

sesiones, por lo tanto se tendrá una sesión teórica y una sesión práctica para poder abarcar todos los contenidos anteriormente descritos.

## **1.5 Elaboración del material didáctico**

El primer paso para la elaboración del material didáctico fue la adquisición de los libros de texto de “Karel, the robot” y “Alice”. Ya que se debía contar con un material profesional y calificado para cumplir con los objetivos del curso de *IPE*.

Los libros se compraron a través de un portal de ventas por internet debido a que en Guatemala no había existencia. El material se tardó, aproximadamente, dos semanas en venir al país.

Al momento de tener el material didáctico, se procedió a evaluar los temas incluidos en estos dos libros con el objetivo de seleccionar los temas indicados para cubrir el contenido del curso *IPE*, así como elaborar el IG del mismo.

Estructurado el IG del curso, se pasó a elaborar el material didáctico propio del *IPE*. Este material cuenta con diapositivas, manuales de laboratorio y folletos de texto, los cuales estarán a disposición de los estudiantes al momento que tomen el curso, además éste fue realizado tanto en idioma inglés como en español, esto para darle continuidad al objetivo de familiarizar a los estudiantes con el idioma inglés.

Además de los libros de “*Karel, the Robot*” y “*Alice*”, así como la base del contenido del curso, también se utilizaron varios libros de la biblioteca del *ITCoE* para completar el contenido del mismo. Uno de ellos fue el de Timothy Budd, que tiene como título *Introducción a la programación orientada a objetos*

(1994), De este se tomaron los primeros conceptos del paradigma de programación estructurada y cómo se dio el cambio de este paradigma al de orientado a objetos, esto ayudó al equipo de trabajo a elaborar un cronograma de cómo fueron cambiando los conceptos a través del tiempo.

## **1.6 Elaboración de la publicidad para el curso de *IPE***

La primera iteración del curso procedimos se llevó a cabo como un curso de inscripción abierta para estudiantes o personas interesadas en obtener el conocimiento básico de la programación, ya que el curso está orientado para individuos que tienen poco o ningún conocimiento en el tema.

Se procedió a divulgar información sobre el curso de *IPE* (ver Apéndice A), con el fin de conseguir personas interesadas en el tema y que tuvieran la disposición de asistir al curso. En la publicidad se indicó: El *IPE* es un curso gratuito que tiene como tema principal la introducción al entorno de programación, orientado a todo aquel estudiante que desee cursar la carrera de ciencias y sistemas, en donde reforzarán los temas base de la misma. También que el curso es impartido en los laboratorios del *ITCoE*, y el único prerrequisito es tener un conocimiento básico en el uso de la computadora (saber cómo encenderla, apagarla, utilizar los componentes básicos de la misma, tales como: mouse, teclado y monitor).

La publicidad fue colocada en las carteleras de información de los edificios de ingeniería y aledaños a la misma, para que toda aquella persona interesada se avocara a las oficinas del *IT Centre of Excellence (ITCoE)* a realizar su inscripción al curso.

## **1.7 Inicio del curso de *IPE***

Se habilitaron, para el curso de *IPE*, dos secciones, en horario de la mañana de 7 a.m. a 9 a.m. y el otro de 2 p.m. a 4 p.m., para que las personas interesadas pudiesen seleccionar el horario de su conveniencia.

A este curso se inscribieron 38 personas, quienes eran procedentes de diferentes colegios de educación media, personas particulares y estudiantes de las distintas carreras y escuelas de la Universidad de San Carlos.

El curso inició el 2 de Diciembre del 2009, pero lamentablemente de los 38 alumnos inscritos solo asistieron 19 repartidos. El curso de la jornada matutina contó con 10 alumnos y la vespertina con 9.

## **1.8 Desarrollo del curso de *IPE***

El curso se desarrolló conforme a la metodología del *ITCoE*, la cual consiste, principalmente, en dar una clase teórica y luego, una clase práctica para corroborar que los estudiantes hayan aprendido los conceptos de la clase teórica. Aparte de ello, se aplica el concepto de “Yo hago, nosotros hacemos y tú haces”, por ejemplo: el instructor realiza un ejemplo práctico, y los estudiantes prestan atención; luego se lleva a cabo un ejercicio de manera conjunta, instructor y estudiantes, y por último, los estudiantes realizan un ejercicio solos, tratando que los tres ejercicios sean del mismo grado de complejidad. La meta de esta actividad es apreciar el desenvolvimiento de los estudiantes en ambas áreas (teórica y práctica) y así efectuar una retroalimentación para reafirmen los conceptos adquiridos por los estudiantes.

Las clases se desarrollaron de lunes a viernes, dos horas diarias, en los laboratorios del *ITCoE*. Para impartir la docencia se utilizaron herramientas, tales como: pizarrones, marcadores, cañonera, y computador y computadoras personales para los alumnos, para que realizaran sus respectivas prácticas.

Se utilizó el material elaborado para el curso, como diapositivas, folletos de texto y manuales de laboratorios. No obstante el estudiante tenía acceso a los libros de texto que se encuentran en la biblioteca del *ITCoE*, para ampliar de manera individual, el contenido de un tema.

El curso se desarrolló en orden y sin ningún retraso hasta la sesión número 8 del módulo de programación estructurada, porque fueron interrumpidas las clases, debido al cierre, por dos semanas, aproximadamente, de las instalaciones de la facultad de ingeniería, por un grupo de estudiantes molestos e inconformes por la mala administración de esta Unidad Académica. Este evento ocasionó que las clases se retrasaran y que los estudiantes interesados en el curso se frustraran y decidieran por renunciar al mismo.

Debido a este incidente ya no se pudo realizar la evaluación previamente calendarizada del módulo de programación estructurada.

A pesar de este suceso, los datos que se pudieron recabar fueron las evaluaciones realizadas en las clases prácticas sobre el desempeño de los estudiantes, de esta forma se pudo identificar las debilidades en el aprendizaje de los estudiantes.

### **1.9 Resultados del curso de *IPE***

Los resultados que pudimos recabar en esta primera iteración del curso de *IPE*, los obtuvimos de las evaluaciones realizadas en las clases prácticas,

esto debido a que tanto la evaluación final del modulo de programación estructurada como el modulo de programación orientada a objetos no se pudieron llevar a cabo por el cierre de las instalaciones de la facultad de ingeniería anteriormente comentado.

Por lo tanto los datos procesados para la sección matutina del curso de *IPE* fueron los siguientes:

**Tabla I.Resultados estadísticos de *IPE* sección A (matutina)**

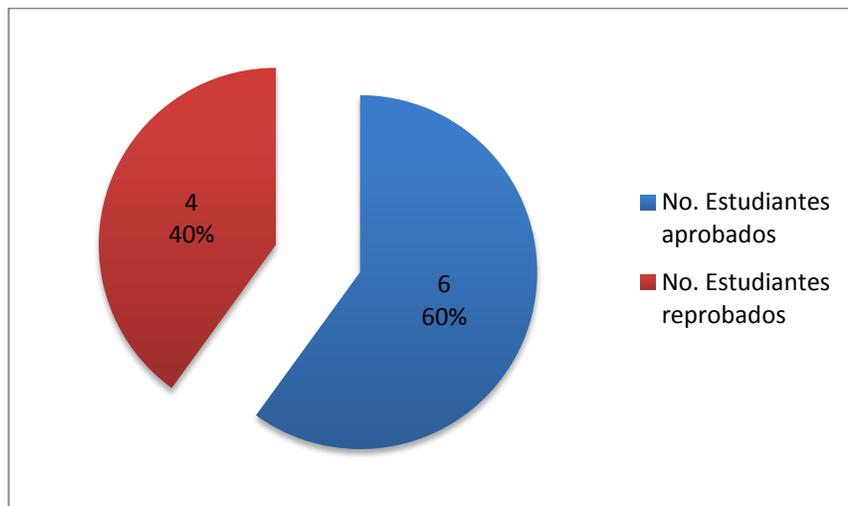
<b>Dato</b>	<b>Valor</b>
Media	19.62916667
Error típico	2.811444852
Mediana	22.1625
Moda	21
Desviación estándar	157.1571875
Varianza de la muestra	19.64464844
Curtosis	-0.938691268
Coefficiente de asimetría	-0.963864736
Rango	0.056338028
Mínimo	12.0375
Máximo	23.925
Suma	176.6625
Total Estudiantes	10
No. Estudiantes aprobados	6
No. Estudiantes reprobados	4

Fuente: elaborada por los autores del documento

Los datos reflejan que el 60% de los estudiantes aprobaron y un 40 % reprobó, esto nos indica que apenas la mitad de los alumnos tuvieron una nota satisfactoria. Los alumnos que reprobaron, tenían características en común, las

cuales pudimos identificar estas eran desinterés, falta de participación en clase y poca retención del contenido visto.

**Figura 1. Porcentaje de estudiantes aprobados y reprobados en IPE sección A (matutina) de la 1era iteración**



Fuente: elaborada por los autores del documento

Y los datos procesados para la sección vespertina del curso de *IPE* fueron los siguientes:

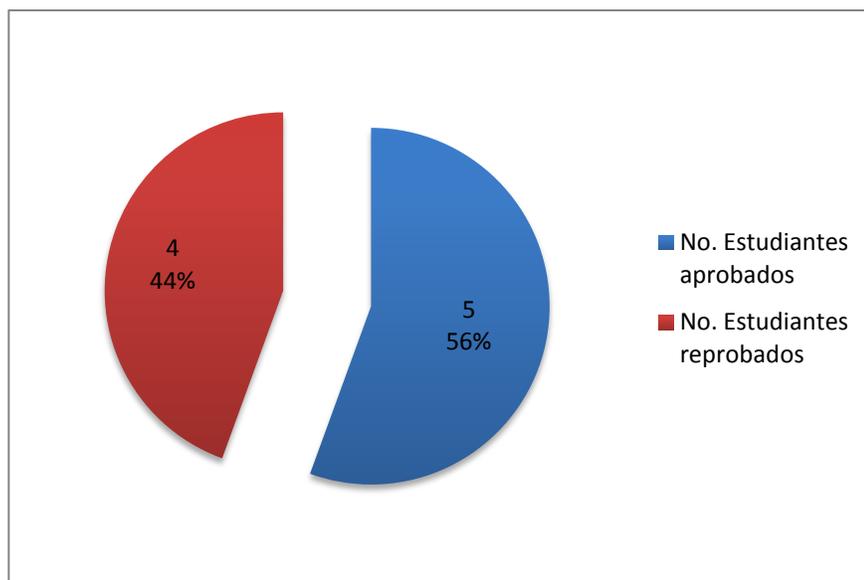
Tabla II. Resultados estadísticos de *IFE* sección B (vespertina)

Dato	Valor
Media	18.375
Error típico	4.425590154
Mediana	18.5625
Moda	21
Desviación estándar	296.3671875
Varianza de la muestra	37.04589844
Curtosis	-0.791752891
Coficiente de asimetría	-0.078735445
Rango	0.154929577
Mínimo	8.625
Máximo	27.75
Suma	165.375
Total Estudiantes	9
No. Estudiantes aprobados	5
No. Estudiantes reprobados	4

Fuente: elaborada por los autores del documento

Como se puede observar, el porcentaje de alumnos aprobados es del 55%, este es bastante favorable si se evalúa que ese porcentaje de alumnos aprobados es prácticamente el mismo a la cantidad de alumnos constantes, lo cual indica que los alumnos que reprobaron fueron aquellos que no asistieron al curso con regularidad.

Figura 2. **Porcentaje de estudiantes aprobados y reprobados en IPE sección B (vespertina) de la 1era iteración**



Fuente: elaborada por los autores del documento

### 1.10 Mejoras propuestas para el curso de IPE

Las mejoras que se propusieron para optimizar el curso de IPE fueron las siguientes:

- Realizar una retroalimentación constante con los estudiantes a través de una serie de preguntas al inicio de cada sesión, para poder verificar qué conceptos necesitan ser repasados y cuáles le quedaron claros al alumno.
- Ampliar los manuales de laboratorio creando ejercicios según el nivel de complejidad, en categorías: básicos, medios y complejos, para que el educando pueda ir desafiando dichos niveles de complejidad.

- Generar varios exámenes para poder evaluar el desempeño de los estudiantes.
- Actualizar el contenido del material elaborado, debido al constante surgimiento de nuevas versiones del *software* utilizado, para estar actualizados con el contenido del curso de *IPE*.
- Debido a que *IPE* es un curso bastante básico y enfocado a estudiantes que están empezando a integrarse al ambiente de programación, se sugiere, al momento de impartir el curso, que este proceso sea dinámico y entretenido, para brindarle al estudiante un enfoque agradable de lo que es la programación, ya que muchas veces la idea que se tiene de este concepto, es que es un tema tedioso y aburrido.
- Proveer al estudiante del *software* utilizado en clase y capacitarlo en la instalación del mismo, para que de esta forma él pueda utilizar y experimentar con las herramientas utilizadas en clase y así poder agilizar sus habilidades en el control y manejo de las mismas.

### **1.11 Entrega del material didáctico del curso a las nuevas autoridades y personal administrativo del *ITCoE***

Al finalizar el estudio profesional supervisado, se efectuó la entrega de todo el material didáctico elaborado a las autoridades administrativas y académicas del *ITCoE*, para que sea utilizado por el Centro en futuras iteraciones del curso de *IPE*. Este será de gran ayuda a todo aquel estudiante de primer ingreso de la carrera de Ciencias y Sistemas, que quiera reforzar los conceptos base del ambiente de programación.



## **2 SEGUIMIENTO AL ESTUDIO DE LA METODOLOGÍA DE ENSEÑANZA DE TECNOLOGÍA DEL *ITCOE***

### **2.1 Antecedentes**

A mediados del año 2008, la administración del *ITCoE* fue dejada en manos de autoridades locales. Un grupo de estudiantes de la Facultad de Ingeniería brindó el apoyo necesario para que el *ITCoE* continuara en funcionamiento.

Los estudiantes que se encargaron de administrar el centro en el segundo semestre de 2008, propusieron al director de la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería, la realización de un estudio sobre la metodología de enseñanza del *ITCoE* aplicada a los cursos de dicha escuela. Este estudio consistió en seleccionar, al inicio de cada semestre, una muestra de 75 estudiantes (este número varió a lo largo del estudio) del curso IPC1 e impartirles el mismo contenido que el programa de la Escuela de Ciencias y Sistemas, pero adaptado a la metodología del *ITCoE*.

Los resultados fueron positivos al comparar el porcentaje de aprobación entre los estudiantes de la muestra y el resto de alumnos de este curso. Este resultado dio la pauta para que otro grupo de instructores encargados del centro diera continuidad al estudio, realizando una segunda iteración del curso IPC1 además de agregar la primera iteración del curso IPC2. Los resultados de este segundo estudio fueron similares al primero.

## **2.2 Descripción del estudio**

Como parte del apoyo que se brinda al *ITCoE*, se dio continuidad al estudio de la aplicación de la metodología del *ITCoE* a los cursos de Introducción a la Programación y Computación 1 e Introducción a la Programación y Computación 2 que pertenecen a la Escuela de Ciencias y Sistemas. Para ello se aplicaron las fases de planificación, ejecución y evaluación que describe la metodología.

### **2.2.1 Fase de preparación**

La preparación de los cursos de IPC1 e IPC2 tuvo cambios significativos en la calendarización de ambas asignaturas, estos cambios fueron ocasionados debido a dos factores principales, los cuales se describen a continuación:

- El acuerdo al que se llegó con las autoridades de la Escuela de Ciencias y Sistemas de hacer equivalente el estudio, únicamente, con el laboratorio del curso y no con la ponderación total del mismo, como se había realizado en las iteraciones anteriores.
- El inicio del semestre en la Facultad de Ingeniería se vio afectado por el cierre de la Universidad debido a disturbios provocados por grupos de estudiantes inconformes con las autoridades universitarias.

Por lo anterior, se tomó la decisión de disminuir de dos secciones por curso, a sección única para IPC1 e IPC2. Además, la calendarización dentro del *Instructor Guideline* sufrió modificaciones, las cuales se describen a continuación:

- Las secciones de *Session Detail Form* y *Module Breakup Form*, sufrieron una disminución de sesiones asignadas debido a los problemas descritos anteriormente.
- La parte de *Self Study* tuvo un rol aún más importante ya que varias de las sesiones que se eliminaron de las clases presenciales del *Session Details Form* y *Module Breakup Form* fueron distribuidas entre las horas de estudio en casa.

Además, se plantearon los siguientes factores de selección de estudiantes para los cursos:

- Aprobación de la prueba de idioma inglés que contiene una serie escrita y otra escuchada.
- Originalmente se planteó que los estudiantes interesados no debían haber cursado y reprobado cualquiera de los dos cursos IPC1 e IPC2 en alguna de las dos iteraciones anteriores. Para este punto se hizo una excepción en el curso de IPC2, que estos estudiantes podían utilizarse como muestra para observar el comportamiento de alumnos repitentes en el estudio.

A diferencia de las iteraciones anteriores, no se tomó en cuenta el factor que aceptaba, únicamente, a los estudiantes que nunca habían cursado IPC1 o IPC2 en la Escuela de Ciencias y Sistemas. Por último, aplicando los factores anteriormente descritos, la muestra de alumnos para cada curso, no debía ser menor a 15 ni mayor de 25 personas, para no tener una muestra tan pequeña y por las capacidades del laboratorio.

El horario seleccionado para los cursos fueron los siguientes:

- El curso de IPC2 se impartió en el Laboratorio 1 de lunes a viernes de 9:00 a 11:00 horas.
- El curso de IPC1 se impartió en el Laboratorio 1 de lunes a viernes de 11:00 a 13:00 horas.

### **2.2.2 Fase de ejecución**

La ejecución de la tercera iteración del estudio se llevó a cabo en las instalaciones del *ITCoE* en el período comprendido de julio a noviembre de 2009.

Esta fase inició con el proceso de selección de estudiantes para cada curso. Después de aplicar los factores de selección descritos en la fase de preparación, la distribución de estudiantes quedó de la siguiente manera:

- Para IPC1, 18 estudiantes en el curso con código IPC1006
- Para IPC2, 25 estudiantes en el curso con código IPC2003

Ya establecida la distribución de alumnos en cada curso y con los filtros de selección aplicados, se principio con el desarrollo de las clases con base en las sesiones definidas en la guía del instructor. Esta parte del proceso se llevó a cabo con los siguientes aspectos generales:

- Se dejó a disposición del instructor el uso del idioma inglés en forma oral para impartir el curso como opcional, esto por la necesidad de agilizar el desarrollo del curso, por los acontecimientos descritos anteriormente que afectaron el tiempo del semestre.
- La atención al estudiante por parte del instructor en horarios fuera del horario de clase, se puso a disposición en la oficina administrativa del *ITCoE*, la cual también fue utilizada para la revisión de tareas y evaluaciones.

Los cursos se desarrollaron con base en la estructura generada por la primera iteración del estudio, la cual indica lo siguiente:

- Desarrollar los módulos definidos en la guía del instructor en los que se imparte teoría y práctica, estos módulos son los descritos en la sección 1.4;
- Finalizar con un módulo de desarrollo de un proyecto, en el cual se pone en práctica lo aprendido en los módulos anteriores, con la característica especial que este proyecto es propuesto por los mismos estudiantes. La propuesta es sometida a una evaluación con el fin que dicha propuesta cumpla con los requerimientos mínimos definidos por el instructor y personal administrativo.

Además, durante esta fase de ejecución, se presentaron dos factores externos que provocaron ciertos cambios en la metodología utilizada, estos se describen a continuación:

- Disturbios que provocaron el cierre de las instalaciones del edificio T-3 de la Facultad de Ingeniería, lo cual llevó a realizar una redistribución de las sesiones programadas para ser impartidas por el instructor, en el área de *Self Study*.
- Debido a que algunos de los estudiantes dentro del estudio debían cursar otras asignaturas de la Escuela de Ciencias y Sistemas existió el traslape de horarios con respecto a la revisión de algunos proyectos de las otras asignaturas calendarizadas en horarios en los que se impartía el curso de IPC1 o IPC2, por lo que fue necesario que el instructor mostrará un nivel de flexibilidad con respecto al horario de entrada de los alumnos que solicitaban previamente permiso, para asistir a las actividades obligatorias en sus otros cursos.

### **2.2.3 Fase de evaluación**

La fase de evaluación se llevó a cabo durante los meses de Diciembre 2009 y Enero 2010, en ella se realizó un análisis de los resultados finales de los cursos provistos por los instructores, así como de los reportes mensuales que ellos realizaban, en este análisis se detalla el progreso de los estudiantes mes a mes.

## **2.3 Resultados del estudio**

Los resultados estadísticos del estudio se basan en los datos de los estudiantes que concluyeron el curso de IPC1 o IPC2, ya sea satisfactoria o insatisfactoria, dejando de lado a los alumnos que fueron considerados como *dropout*, es decir, alumnos que desertaron del curso.

## Resultados de IPC1

En la Tabla III se muestran los resultados estadísticos obtenidos del curso IPC1006 que se impartió en el segundo semestre de 2009, como sección única.

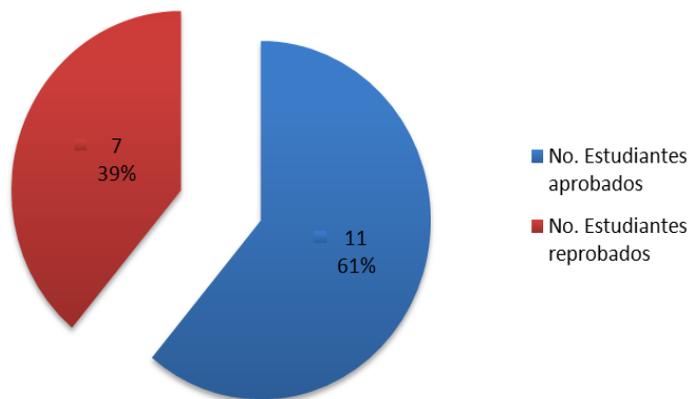
**Tabla III. Resultados estadísticos de IPC1 en el *ITCoE***

<b>Dato</b>	<b>Valor</b>
Media	57
Mediana	62.05
Moda	67
Desviación estándar	17.2198307
Varianza de la muestra	296.522568
Curtosis	1.23301707
Coefficiente de asimetría	-1.33384495
Rango	0.314
Mínimo	17.78
Máximo	76.49
Suma	1026.09
Total Estudiantes	18
No. Estudiantes aprobados	11
No. Estudiantes reprobados	7

Fuente: elaborada por los autores del documento

En la Figura 3 se muestra el porcentaje de aprobación obtenido en la tercera iteración, en este caso fue del 61%.

Figura 3. **Porcentaje de estudiantes aprobados y reprobados en IPC1 dentro del ITCoE en el segundo semestre 2009**



Fuente: elaborada por los autores del documento.

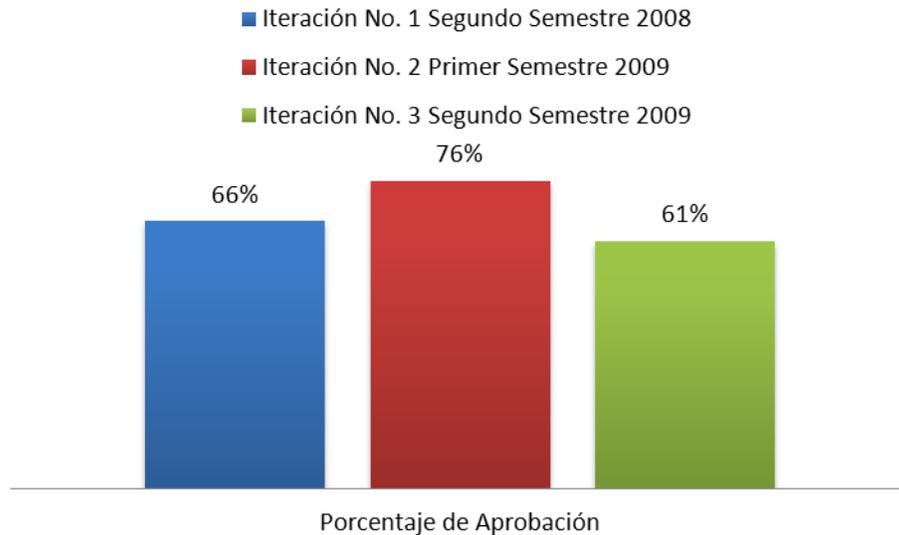
El resultado del estudio fue del 61%, este porcentaje es menor con respecto al 66%<sup>1</sup> obtenido en la primera iteración, en el segundo semestre del año 2008 y al 76%<sup>2</sup>, de la segunda iteración, en el primer semestre del año 2009. En la figura 4 se puede observar un gráfico comparativo de los porcentajes de aprobación a través de las 3 iteraciones.

---

<sup>1</sup>Ver Anexo A.

<sup>2</sup> Ver Anexo B.

Figura 4. **Porcentaje de aprobación del curso de IPC1 en iteración 1, 2 y 3 del Estudio**



Fuente: elaborada por los autores del documento.

Cabe mencionar que la disminución del porcentaje de aprobación está sujeta, principalmente, al factor tiempo, el cual se vio afectado de manera negativa debido a dos eventos de inconformidad por parte de grupos de estudiantes que desembocaron; primero en el cierre de la Facultad de Ingeniería y más adelante en el cierre del campus central de la Universidad de San Carlos. Dichos eventos retrasaron el desarrollo planificado de los cursos de IPC1, sin dejar de lado otros factores que pudieron afectar el rendimiento de los estudiantes, los cuales se describirán en las conclusiones del estudio (Sección 1.4).

### 2.3.1 Resultados de IPC2

En la Tabla IV se muestran los resultados estadísticos obtenidos del curso IPC2003 que se impartió en el segundo semestre de 2009, como sección única.

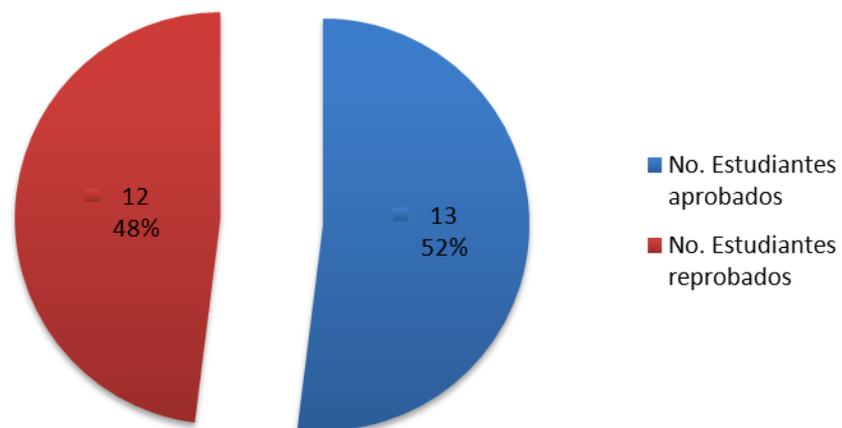
Tabla IV. Resultados Estadísticos de IPC2

<b>Dato</b>	<b>Valor</b>
Media	56.88
Mediana	61.31
Moda	72
Desviación estándar	16.652264
Varianza de la muestra	277.297896
Curtosis	1.10663133
Coeficiente de asimetría	-1.04959594
Rango	0.479
Mínimo	14.78
Máximo	81.24
Suma	1422.1
Total Estudiantes	25
No. Estudiantes aprobados	13
No. Estudiantes reprobados	12

Fuente: elaborada por los autores del documento.

En la Figura 5 se muestra el porcentaje de aprobación obtenido en esta tercera iteración, en este caso fue del 52%

Figura 5. **Porcentaje de estudiantes aprobados y reprobados en IPC2003 dentro del ITCoE en el segundo semestre 2009**



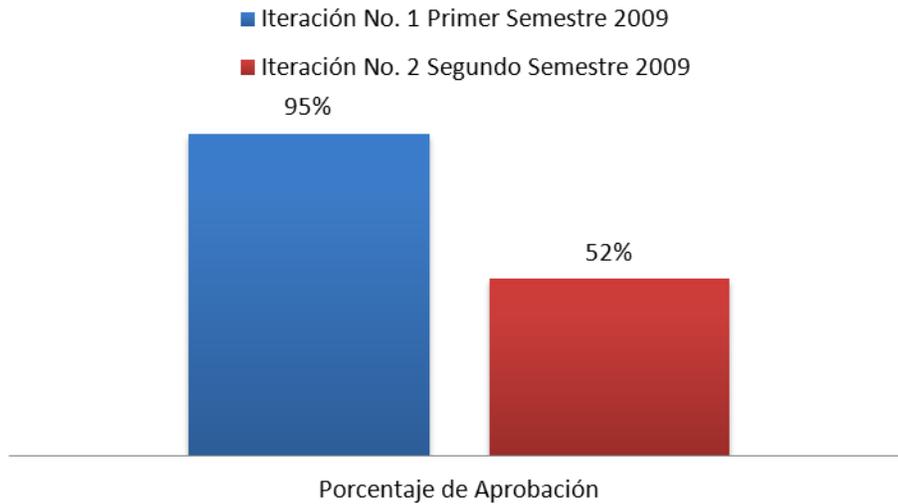
Fuente: elaborada por los autores del documento.

El resultado del estudio fue del 52%, el cual es mejor con respecto al 95%<sup>3</sup> obtenido en la primera iteración en el primer semestre del año 2009. En la figura 6 se puede observar un gráfico comparativo de los porcentajes de aprobación a través de las dos iteraciones.

---

<sup>3</sup> Promedio aproximado de los porcentajes mostrados en el Anexo C.

Figura 6. **Porcentaje de aprobación del curso de IPC2 en iteración 1 y 2 del Estudio**



Fuente: elaborada por los autores del documento.

Al igual que en los resultados de IPC1 se debe mencionar que la disminución del porcentaje de aprobación está sujeta, principalmente, al factor tiempo, el cual se vio afectado, por los sucesos descritos en la sección 2.3.1, en el transcurso del semestre, de manera repetida y negativa para efectos del estudio durante varias ocasiones, aunque sin dejar de lado otros factores que pudieron afectar el rendimiento de los estudiantes, los cuales se describirán en las conclusiones del estudio (Sección 1.4).

## 2.4 Factores que influyeron negativamente en estudio

Existen factores que pudieron ser la causa de la disminución en el rendimiento tanto en IPC1 como en IPC2, los cuales se describen a continuación:

- Como se mencionó en la sección 2.3.1, los diferentes sucesos que desembocan en el cierre de las instalaciones de la Facultad de Ingeniería o de la Universidad de San Carlos que afectaron, de manera negativa, el calendario para llevar a cabo las planificaciones que se realizan a principio del semestre. Estos hechos sucedieron en forma repentina y repetidamente durante el desarrollo del semestre, situación que contribuyó a la acumulación de carga académica para los estudiantes. Además, el objetivo de tener una planificación que permita el desarrollo adecuado de los contenidos del curso, perdió totalmente su sentido.
- Traslape de horario de las asignaturas de la Escuela de Ciencias y Sistemas con los horarios propuestos por el *ITCoE*. Debido a que en dichas asignaturas existen proyectos para los estudiantes, las calificaciones de los mismos son realizadas en horarios que el auxiliar de la asignatura tiene disponible, es decir, no es un horario programado, por lo que en diversas ocasiones, las calificaciones se traslapaban con los horarios programados del *ITCoE*.
- Barrera del idioma inglés. A pesar de realizar una evaluación del idioma inglés para los estudiantes, al inicio del estudio; se pudo notar que dicho idioma fue un factor que influyó en la disminución del rendimiento, ya que los estudiantes participaban más cuando el instructor hacía uso del idioma español.

- Adaptación a una nueva metodología. Los estudiantes se tienen que adaptar a una metodología más rígida en horarios y responsabilidades en comparación con los cursos de la Escuela de Ciencias y Sistemas.

Cabe mencionar que para el curso de IPC1, aunque existió esa disminución de rendimiento debido a los factores anteriormente descritos, la muestra respondió de una manera adecuada en relación al tiempo que se tenía disponible para desarrollar el curso.

## **2.5 Recomendaciones del estudio**

El seguimiento a este estudio es necesario para obtener resultados que reflejen que la metodología del *ITCoE* es viable en un ciclo o semestre en el que los sucesos extraordinarios sean controlados de una manera adecuada y que no tengan el alto impacto que fue reflejado en esta iteración del estudio. Además, el seguimiento permitirá que la barrera del idioma vaya en disminución cada vez más, así como, la adaptación a otra metodología no se vería como un cambio tan abrupto.

Con respecto a los traslapes de horario de los estudiantes se recomienda que exista una mejor comunicación entre las autoridades de la Escuela de Ciencias y Sistemas y las del *ITCoE* con el objetivo de evitar este tipo de conflictos y que existan horarios definidos para las calificaciones de los proyectos de los estudiantes.

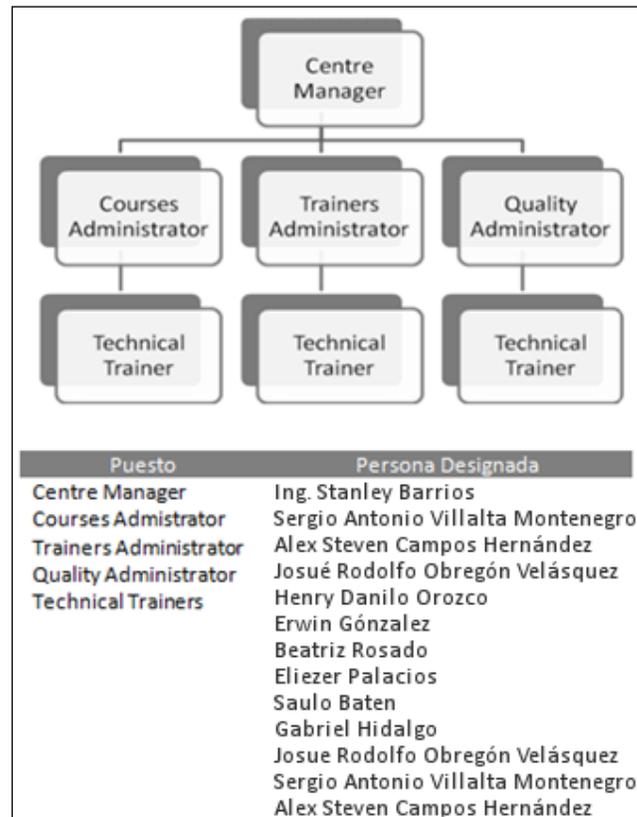
### **3 APOYO ADMINISTRATIVO PRESTADO AL *ITCOE***

#### **3.1 Estructura organizacional**

Se formó parte del equipo técnico y administrativo del *ITCoE* durante el período de agosto 2009 a febrero 2010, con el objetivo de aprovechar la metodología de enseñanza utilizada en el centro y poder utilizarla en el desarrollo del nuevo curso *IPE*. En la Figura 7, se muestra la estructura organizacional del período mencionado.

El apoyo al área administrativa del centro permitió conocer de una manera más detallada los procesos y la metodología utilizada dentro del mismo, para poder aplicarla en un proceso de enseñanza más efectivo. Este involucramiento fue importante ya que sentó las bases de creación del curso *IPE* y del material de apoyo utilizado para impartirlo. En las siguientes secciones se describen los detalles del apoyo administrativo brindado.

Figura 7. Estructura organizacional del *ITCoE*



Fuente: Hernández, César y Rodas, David (2009). “Análisis, implementación y evaluación de la metodología utilizada en el proceso de enseñanza de tecnología implementada en el *ITCoE*”.

Página 21. Modificado por los autores del documento.

### 3.2 Administración de cursos

En la administración del *ITCoE* llevada a cabo en el primer semestre de 2009, por estudiantes del Ejercicio Profesional Supervisado, se creó una división lógica de los distintos cursos impartidos en el centro, con el objetivo de mejorar el acoplamiento del manejo de este al sistema administrativo universitario y de una mejor satisfacción de las necesidades de los distintos sectores de estudiantes que posee el mismo. Se dio un seguimiento a esta

división y se trabajó sobre la misma línea durante el período comprendido de agosto de 2009 a febrero de 2010.

Durante este período se impartieron 6 cursos los cuales se distribuyeron en 15 secciones. Los cursos y la cantidad de secciones habilitadas para cada uno se observa en la Tabla V.

**Tabla V. Cursos disponibles durante el segundo semestre de 2009**

<b>Código</b>	<b>Curso</b>	<b>Secciones</b>
<i>ISE</i>	<i>Introduction to Software Engineering</i>	3
<i>SQM</i>	<i>Software Quality Management</i>	5
<i>SA</i>	<i>Solution Architecture</i>	3
<i>ORR</i>	<i>Relational Database Management Systems Concepts</i>	1
<i>ORS</i>	<i>Introduction to SQL</i>	2
<i>PBJ</i>	<i>Programming Basics with Java</i>	1

Fuente: elaborada por los autores del documento

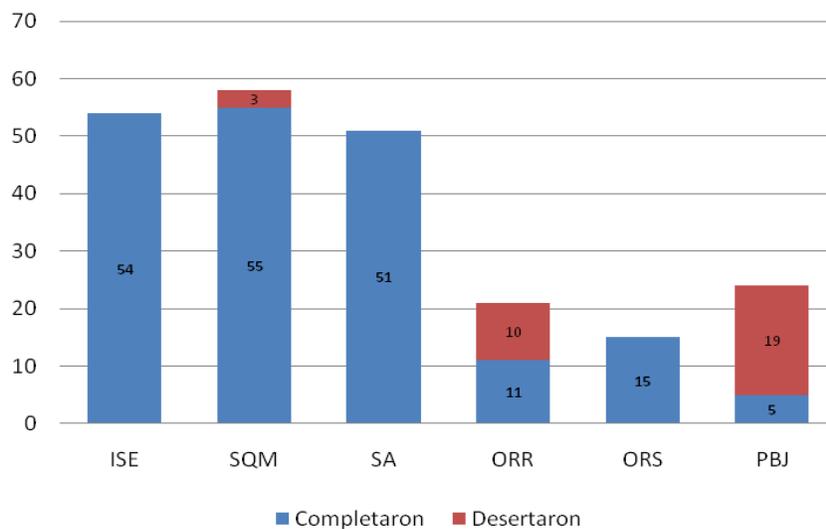
En el *ITCoE* mantener la regularidad de los alumnos durante el desarrollo de los cursos es un factor muy importante, por lo que se lleva el control de las deserciones en cada sección habilitada, estos datos se ven reflejados en la Tabla VI.

Tabla VI. **Total de estudiantes que se inscribieron en los cursos, los que completaron y los que desertaron**

<b>Código</b>	<b>Total Alumnos</b>	<b>Completaron</b>	<b>Desertaron</b>
ISE	54	54	0
SQM	58	55	3
SA	51	51	0
ORR	21	11	10
ORS	15	15	0
PBJ	24	5	19
<b>Total</b>	<b>223</b>	<b>191</b>	<b>32</b>

Fuente: elaborada por los autores del documento.

Figura 8. **Comparación de las cantidades de los estudiantes que completaron con los que desertaron**



Fuente: elaborada por los autores del documento.

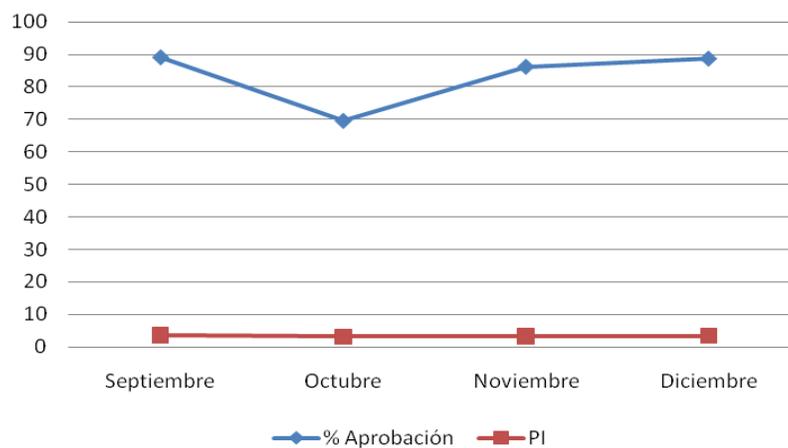
Además, los cursos de manera frecuente, generaban un porcentaje de aprobación mensual, el cual se ve detallado en la Tabla VII.

Tabla VII. **Porcentaje de aprobación e indicador de rendimiento de cada mes**

Mes	% Aprobación	PI
Julio	NA	NA
Agosto	NA	NA
Septiembre	89.14	3.56
Octubre	69.66	3.16
Noviembre	86.25	3.28
Diciembre	88.75	3.41
<b>Total Promedio</b>	<b>83.45</b>	<b>3.35</b>

Fuente: elaborada por los autores del documento.

Figura 9. **Tendencia del porcentaje de aprobación y del PI<sup>4</sup> mensual**



Fuente: elaborada por los autores del documento.

---

<sup>4</sup>Performance Indicator. Indicador de rendimiento de los cursos, que está entre 0 y 5, en donde cantidades cercanas a 5 indican que en el curso no solo han salido aprobados los estudiantes, sino que también han obtenido buenas calificaciones.

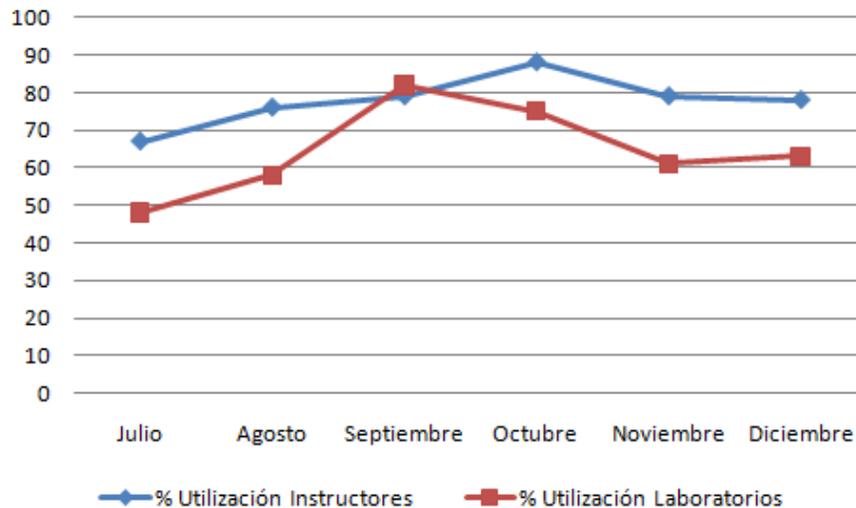
También se llevó el control tanto de la utilización del recurso humano como del físico, en este caso, instructores y laboratorios respectivamente. Los porcentajes mensuales de los recursos se muestran en la Tabla VIII.

**Tabla VIII. Porcentaje de utilización de recursos (instructores y laboratorios) durante cada mes**

<b>Mes</b>	<b>% Utilización Instructores</b>	<b>% Utilización Laboratorios</b>
<b>Julio</b>	67	48
<b>Agosto</b>	76	58
<b>Septiembre</b>	79	82
<b>Octubre</b>	88	75
<b>Noviembre</b>	79	61
<b>Diciembre</b>	78	63
<b>Total Promedio</b>	<b>78</b>	<b>65</b>

Fuente: elaborada por los autores del documento.

Figura 10. **Tendencia del porcentaje de utilización de instructores y laboratorios**



Fuente: elaborada por los autores del documento.

### 3.2.1 Generación de nuevos cursos

Este es el objetivo principal del trabajo realizado en el *ITCoE*, la creación del curso *Introduction to Programming Environment*. Una revisión exhaustiva y detallada del proceso de creación de todo el material necesario para impartir este curso, se encuentra en la sección 3 de este trabajo.

### 3.2.2 Pruebas de conocimiento del idioma inglés

Al momento de iniciar las actividades de enseñanza dentro del *ITCoE*, se realizó una prueba preliminar de conocimientos del idioma inglés a los estudiantes de nuevo ingreso, a los cursos del centro. La prueba consistía en reproducir un vídeo en inglés en tres iteraciones. Seguidamente, se pasaba un conjunto de preguntas acerca del video, con el objetivo de comprobar la

comprensión del idioma hablado y el nivel de redacción en inglés, de los estudiantes.

### **3.2.3 Sondeo sobre preferencias para abrir cursos nuevos**

En la segunda semana del mes de Noviembre 2009, durante tres días, se llevó a cabo un sondeo en las instalaciones de la Facultad de Ingeniería tomando como muestra la población de estudiantes que se encontraban en dicho lugar en horario de las 10:00 a las 15:00 horas. El objetivo principal de la investigación fue crear una lista de intereses y preferencias tecnológicas entre los estudiantes, para obtener una mayor afluencia a los cursos impartidos en el *ITCoE*.

### **3.2.4 Creación de instructivos de procedimientos básicos del *ITCoE***

Debido a que se contó con varias personas nuevas que deseaban colaborar con las actividades del centro, se procedió a la realización de una serie de instructivos detallados de los distintos procesos básicos realizados en el centro. Estos instructivos describen paso a paso las actividades que desarrollan el personal del *ITCoE* dentro de estos procesos. Los instructivos creados fueron los siguientes:

- Configuración del proxy en Linux por medio de Squid
- Configuración del servicio DHCP en los servidores del centro
- Características principales de las computadoras dentro del centro
- Generación de certificados y constancias de cursos
- Como encender y apagar los servidores del centro
- Instalación remota de aplicaciones

### **3.2.5 Préstamo de los laboratorios a la Rectoría de la USAC**

Durante el segundo semestre del año 2009, los laboratorios del *ITCoE* fueron utilizados por el personal que labora en la Rectoría de la USAC, a manera de préstamo. La Rectoría solicitó varios cursos de actualización tecnológica para sus empleados, los cuales fueron impartidos por alumnos de la Escuela de Ciencias de Sistema quienes realizaban su práctica final. Estos se desarrollaron los días lunes, miércoles y viernes de 7:00 a 9:00 horas y de 13:00 a 14:00 horas.

### **3.2.6 Préstamo de los laboratorios para el Congreso de Matemáticas**

En la última semana de noviembre del año 2009, se realizó el XII Congreso de Matemática Educativa en las instalaciones de la Facultad de Ingeniería. Las personas organizadoras del Congreso solicitaron el uso de los laboratorios debido al tipo de equipo con que se cuenta. Únicamente se instaló la aplicación Scientific Notebook, a solicitud de los organizadores del evento.

### **3.2.7 Préstamo de los laboratorios para COECYS**

En la segunda semana de noviembre del año 2009, se llevó a cabo el IX Congreso de Estudiantes de Ciencias y Sistemas. Como actividades pre-congreso se realizaron una serie de conferencias con temas de interés para los estudiantes de la Escuela de Ciencias y Sistemas de la Facultad. Para llevar a cabo las conferencias se requirió la utilización de los dos laboratorios del *ITCoE*, en horario de 19:00 a 20:00 horas, de lunes a viernes durante dos semanas. No se instaló ninguna aplicación extra puesto que ya se contaban con todos los requisitos para las conferencias.

### **3.2.8 Instalación antivirus**

En el segundo semestre del año 2009, en la Facultad de Ingeniería, se inició una campaña de instalación del antivirus NOD32 debido a su comprobada eficacia en la protección de equipos informáticos contra ataques no deseados. La licencia de este antivirus fue provista por Centro de Cálculo. En los laboratorios del *ITCoE*, también debía realizarse dicha instalación. Para ello se contó con estudiantes de prácticas iniciales de la carrera de Ciencias y Sistemas quienes realizaron con éxito la instalación del *software* de protección.

Esto permitió una mejor protección de la información y las aplicaciones utilizadas en el laboratorio, ya que previamente se había tenido la mala experiencia de infecciones con virus y malware de todo tipo dentro de la red de los laboratorios del *ITCoE*, provocando el mal funcionamiento del equipo y mermando la capacidad de utilización de los laboratorios.

### **3.2.9 Limpieza del equipo de los laboratorios del *ITCoE***

Para que el equipo de cómputo de los laboratorios del centro funcione de manera correcta no solo necesita que su *software* este actualizado y protegido contra programas maliciosos; sino también la parte física, es decir el *hardware* del equipo, se encuentre bien cuidado, limpio y en óptimas condiciones para permitir un rendimiento excelente. Por ello, nuevamente, se contó con la colaboración de estudiantes de prácticas iniciales de la carrera de Ciencias y Sistemas para realizar un proceso controlado y supervisado de limpieza de equipo de los laboratorios. Este proceso debe ser controlado para evitar que el equipo no se dañe al limpiarlo, y pueda continuar funcionando para cumplir su objetivo dentro del centro. El proceso de limpieza se llevó a cabo de manera exitosa.

### **3.2.10 Reparación de la puerta del Laboratorio 2**

La puerta del Laboratorio 2 sufrió daños debido al uso. Las bisagras se dañaron y esto no permitía que se pudiera abrir y cerrar con facilidad y comodidad, tanto para los estudiantes como para los *Trainers* y cualquier visitante que ingresara al laboratorio. Se gestionó la reparación de esta puerta, lo cual se logró con éxito, permitiendo su uso normal, nuevamente.

### **3.2.11 Traducción de las diapositivas del curso ISE al idioma español**

Se solicitó, por parte del director del *ITCoE*, la traducción de 9 archivos de presentación con un promedio de 50 diapositivas por archivo del curso *Introduction to Software Engineering* al idioma español.

Las diapositivas fueron revisadas para mantener la misma línea de información contenida en las de idioma inglés.

### **3.2.12 Capacitación del nuevo equipo de trabajo del *ITCoE***

Una de las tareas de carácter obligatorio dentro del *ITCoE* es la de capacitar al equipo de trabajo que quedará a cargo del centro por parte del grupo de alumnos que terminará el ciclo de práctica.

Esta capacitación incluye el conocimiento de todos los procesos administrativos del centro, tales como:

- Entrega de reportes por curso
- Integración de todos los reportes en un reporte mensual

- Conocimiento de los distintos formularios que existen en el centro
- Forma en que la metodología administrativa soporta la metodología de enseñanza del centro
- Forma de utilización del material didáctico proveído por el centro por parte de los *Technical Trainers*
- Evaluaciones periódicas de los estudiantes hacia sus *Trainers*
- Utilización de los laboratorios
- Utilización de los servidores

### **3.2.13 Reuniones administrativas y de seguimiento**

Durante el período de realización de este trabajo de graduación se llevaron a cabo varias reuniones administrativas y de seguimiento a las actividades desarrolladas en el *ITCoE*.

En estas reuniones participaron el encargado administrativo del centro, el ingeniero encargado de todas las actividades concernientes al centro y los estudiantes que colaboraron como *Technical Trainers* y como ayuda administrativa. En las reuniones se trataron temas diversos con el fin de evaluar el progreso de todas las personas involucradas vía EPS o Práctica Final.

### **3.2.14 Apoyo administrativo en el cambio de autoridades del *ITCoE***

Debido a factores diversos, el *ITCoE* sufrió dos cambios de encargados durante la realización de este trabajo. Por lo anterior, se brindó el apoyo administrativo necesario, para que el centro continuara con sus actividades diarias de una manera normal.

Las tareas más importantes desempeñadas consistieron en registrar las horas trabajadas por los *Technical Trainers*, así como, su evaluación mensual, brindar información referente a cursos a los visitantes, la evaluación de los reportes mensuales del centro, entre otras.

### **3.2.15 Utilización de herramienta para evaluaciones de cursos**

Se comprobó las funcionalidades y beneficios brindados por la herramienta Teach2000 en las evaluaciones de las pruebas de algunos cursos impartidos en el *ITCoE*. En las pruebas realizadas, el resultado fue exitoso. Las notas se obtienen instantáneamente después de que el estudiante finaliza la prueba, esto permite una mejora considerable en el desarrollo de los cursos impartidos. La configuración de la herramienta es sencilla, por lo que se recomienda realizar pruebas en cursos más complicados y utilizarla para mejorar en los tiempos de entrega de notas, de reportes mensuales y de finalización de cursos.



## CONCLUSIONES

1. La creación de herramientas tecnológicas de enseñanza es necesaria para reforzar la abstracción en los conceptos básicos de programación y así permitirle a los estudiantes asimilar, de mejor manera, los nuevos conocimientos que adquirirán en la carrera de Ciencias y Sistemas.
2. Al realizar el estudio del curso *IPE*, se pudo constatar que el método utilizado por el *ITCoE* para enseñar, se adapta muy bien a las necesidades de los cursos de la Escuela de Ciencias y Sistemas. La forma “Yo hago, nosotros hacemos, tú haces”, es apropiada para el tipo de contenido que se imparte en la carrera. El estudiante se siente en confianza al trabajar junto con el *Technical Trainer*, para luego tener la capacidad de realizar la misma tarea por su propia cuenta.
3. Las aplicaciones de enseñanza *Karel the Robot*, que hacen uso de animaciones 2D para transmitir los conceptos básicos de la programación estructurada; y *Alice* que utiliza animaciones 3D para los conceptos de programación orientada a objetos, brindan una perspectiva innovadora que agiliza la comprensión de los conceptos básicos del área de programación.

4. La investigación y el análisis de las dos herramientas seleccionadas (*Karel the Robot* y *Alice*) proporcionaron el soporte necesario para poder desarrollar e implementar la estructura del curso de *IPE*. Generando de esta manera la guía del instructor, el libro de texto, manual de laboratorio y material de apoyo, que son todos los artefactos con los cuales cuenta el curso para su distribución.
5. Se generó el material didáctico para los dos módulos principales del curso, utilizando como base los programas *Karel the Robot* y *Alice*. El material está basado, principalmente, en diapositivas y ejercicios de laboratorio desarrollados en los idiomas español e inglés.
6. Se culminó la primera iteración del curso *IPE*. Se obtuvo la retroalimentación necesaria para poder realizar las mejoras al curso y actualizarlo con respecto a la segunda iteración. Se basó, principalmente, en los resultados obtenidos, problemas encontrados y opinión de los estudiantes que cursaron el mismo, para realizar dichas mejoras.
7. Se dio seguimiento a los proyectos de IPC1 e IPC2 para dar continuidad a los estudios previos a este trabajo. Los resultados obtenidos mostraron que la aplicación de la metodología sigue siendo apropiada en un entorno ideal, ya que el estudio se vio afectado frecuentemente por factores externos al centro.

8. El apoyo administrativo brindado al *ITCoE* permitió que el centro siguiera trabajando en las mejores condiciones posibles, generando calendarios de cursos, reportes mensuales, mantenimiento de laboratorios y atención al estudiante que estuviera interesado en alguna de las áreas académicas que el *ITCoE* apoya.
  
9. El apoyo didáctico brindado al *ITCoE*, por medio de cursos que se encuentran dentro del plan de estudios oficial del centro, permitirá que nuevas generaciones de estudiantes se capaciten y adquirieran los conocimientos deseados en las áreas de la informática, que ellos mismos eligieron.



## RECOMENDACIONES

1. Dar seguimiento a la implementación del curso *IPE*, de tal forma que, en algún momento llegue a ser un curso que brinde las herramientas básicas de comprensión de la programación estructurada y orientada a objetos, a los estudiantes que se inician en la carrera de Ciencias y Sistemas.
2. Tomar en cuenta la metodología de enseñanza del *ITCoE* dentro de los laboratorios prácticos que se imparten en la Escuela de Ciencias y Sistemas, ya que el tipo de metodología se ajusta perfectamente al objetivo de los laboratorios: aplicar los conocimientos teóricos en ejemplos y proyectos de aplicación.
3. Fomentar el uso de herramientas interactivas al momento de transmitir hacia los estudiantes los conceptos teóricos, ya que con estas se puede agilizar la comprensión de los mismos, disminuyendo el tiempo que se invierte en esta parte del proceso de aprendizaje.
4. Seguir el apoyo para el *ITCoE*, ya que es fundamental para mantener un nivel de calidad en los cursos impartidos y en la metodología utilizada en el centro.

5. Velar porque las actividades de administración del centro se realicen correctamente, tales como mantener los laboratorios a su máxima capacidad, actualizar los contenidos de los cursos conforme la realidad guatemalteca, evaluar la calidad de los *Technical Trainers* y generar los indicadores de rendimiento.
  
6. Desarrollar una mejor comunicación entre la Escuela de Ciencias y Sistemas y el *ITCoE* con el objetivo de generar horarios flexibles y con varias secciones para el set de cursos que posee el centro, para que de esta manera el estudiante tenga más opciones al momento de elegir el asignatura que desea cursar y evite traslapes con otras materias de igual importancia. Esto implica aumentar la cantidad de instructores que dan soporte al *ITCoE*.
  
7. Contar con laboratorios de respaldo ubicados fuera de la Universidad de San Carlos de Guatemala, para evitar los hechos que obstaculizan el desarrollo regular de las actividades del *ITCoE*. También, implementar varias vías de comunicación alternas con el estudiante (telefónica y correo electrónico) para que cuando se llegue a presentar algún inconveniente con el cierre de instalaciones, se pueda establecer una comunicación que ayude a continuar con el desarrollo de las actividades académicas del centro.

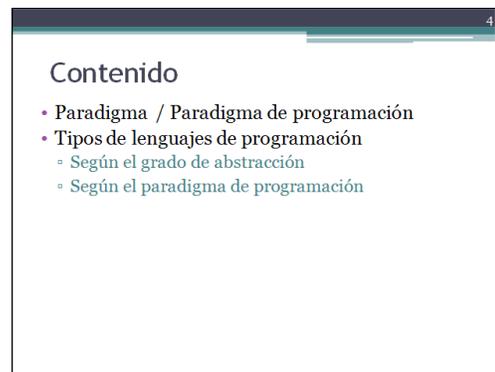
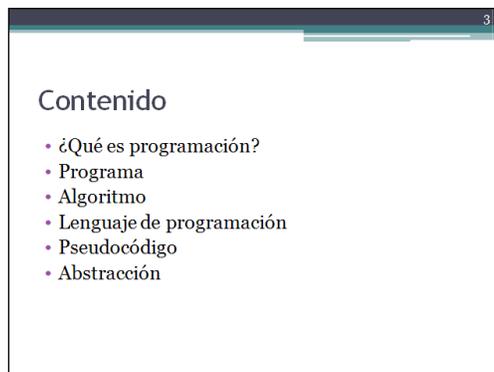
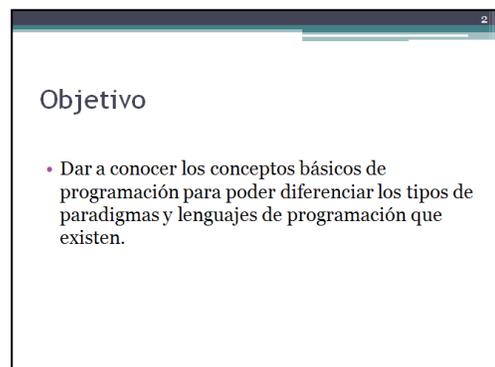
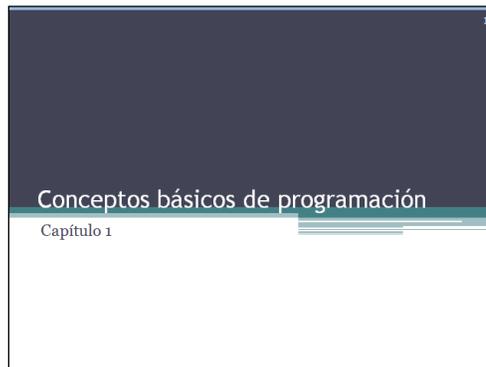
## BIBLIOGRAFÍA

1. DANN, Wanda P., COOPER, Stephen y PAUSCH, Randy. *Learning to Program with Alice*. 2a. ed. New Jersey: Pearson Education Inc., 2006. 354 p. ISBN: 0-13-208516-X
2. HERNÁNDEZ, César y RODAS, David. “Análisis, implementación y evaluación de la metodología utilizada en el proceso de enseñanza de tecnología implementado en el ITCoE”, Escuela de Ciencias y sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala- Mayo 2010.
3. PATTIS, Richard E. *Karel the Robot, A gentle Introduction to the Art of Programming*. 2a. ed. Estados Unidos: John Wiley & Sons, Inc., 1995. 160 p. ISBN: 0-471-59725-2
4. RAMÍREZ, Adin y PANIAGUA, Carlos. “Implementación de la metodología del IT Education Centre India-Guatemala”. Escuela de Ciencias y sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala. Marzo 2009.



# APÉNDICE A

## Capítulo #1 Módulo de programación estructurada “Karel the robot”



5

## Contenido

- Proceso de compilación
- Tipo de errores
  - Léxicos
  - Sintácticos
  - Semánticas

## Capítulo #2 Módulo de programación estructurada “Karel the robot”

1

## Fundamentos de programación

Capítulo 2

2

## OBJETIVO

- Que el estudiante conozca los fundamentos de la programación por medio de los conceptos básicos necesarios para entender los elementos que se utilizarán al momento de realiza un programa de computadora.

3

## CONTENIDO

- Programa
  - Variable
  - Constante
  - Bloque
  - Tipo de dato
  - Identificador
  - Palabra clave
  - Literal
- Operadores
  - Aritmético
  - Relacional
  - Lógico

4

## CONTENIDO

- Condicional
- Unario
- Asignación
- Expresiones
  - Aritméticas
  - Precedencia de operadores

## Capítulo #3 Módulo de programación estructurada “Karel the robot”

1

### Conociendo el entorno de Karel

Capítulo 3



2

### Objetivo

- Dar a conocer el entorno de la aplicación “Karel el Robot”, para familiarizar al estudiante con este nuevo entorno y que aprenda las funciones de cada uno de los componentes de la aplicación.

3

### Contenido

Componentes Principales

- Karel
- Cursor
- Pared
- Beeper
- Calles y Avenidas

4

### Contenido

- Paneles
  - Panel del Mundo Inicial
    - Walls
    - Beepers
    - World File
    - Karel's Initial Situation
    - Cursor
    - Load, Save and Clear World

5

### Contenido

- Panel del Programa
  - Open Program
  - New Program
  - Edit Program
  - Save Program
  - Compile
  - Save As...
  - Print Program
  - Abort Changes

6

### Contenido

- Panel de Ejecución
  - Step
  - Unstep
  - Run
  - Initialize
  - Steps/Min
  - Beepers in Bag
  - Next Instruction
  - World Low
  - Inside
  - Stack
- Ejemplo de un programa en Karel

## Capítulo #4 Módulo de programación estructurada “Karel the robot”

1

# Instrucciones primitivas

Capítulo 4

2

## OBJETIVO

- Permitirle al estudiante conocer los comandos primitivos existentes y la función que realiza cada uno de estos y así tener la bases necesarias para controlar a Karel.

3

## CONTENIDO

- Cambiando de posición
- Manejando los beepers
- Terminando una tarea
- Mi primer programa en Karel
- Error - Shutoffs
- Errores de programación

## Capítulo #5 Módulo de programación estructurada “Karel the robot”

1

# Extendiendo el Vocabulario de Karel

Capítulo 5

2

### OBJETIVO

- Que el estudiante conozca el mecanismo que proporciona Karel para crear nuevas instrucciones por parte del programador en base a las instrucciones primitivas. Esto permite extender el vocabulario de Karel lo cual ayuda a simplificar el trabajo de escribir los programas.

3

### CONTENIDO

- Creando un lenguaje de programación más natural
- Un mecanismo para definir nuevas instrucciones
- El significado de las instrucciones
- Examinando nuevas instrucciones

## Capítulo #6 Módulo de programación estructurada “Karel the robot”

1

### Extendiendo el vocabulario de Karel II

Capítulo 6

2

#### OBJETIVO

- Que el estudiante continúe con el entendimiento del mecanismo para definir nuevas instrucciones de Karel, además de que profundice un poco más en como Karel entiende los programas. También introducir al estudiante a algunas herramientas para diseñar y escribir los programas en Karel.

3

#### CONTENIDO

- Boxing: Como Karel entiende un programa
- Un programa no gramatical
- Herramientas para diseñar y escribir programas en Karel
- Ventajas de utilizar nuevas instrucciones
- Escribiendo programas comprensibles

## Capítulo #7 Módulo de programación estructurada “Karel the robot”

1

### Ejecutando Instrucciones Condicionalmente

Capítulo 7

2

### OBJETIVO

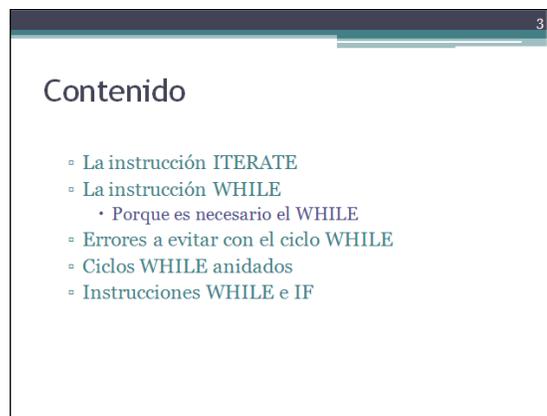
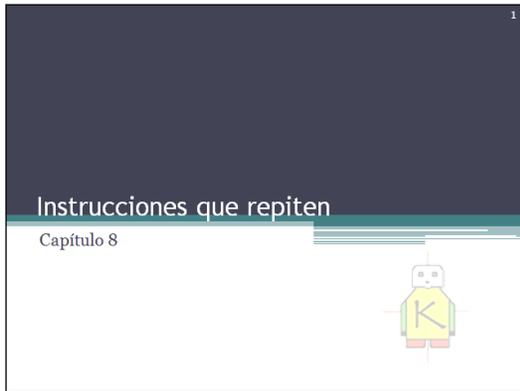
- Que el estudiante conozca las instrucciones condicionales que Karel provee, las cuales le sirven para adaptarse a situaciones que se presenten en su mundo y así no depender únicamente de una situación inicial ideal.

3

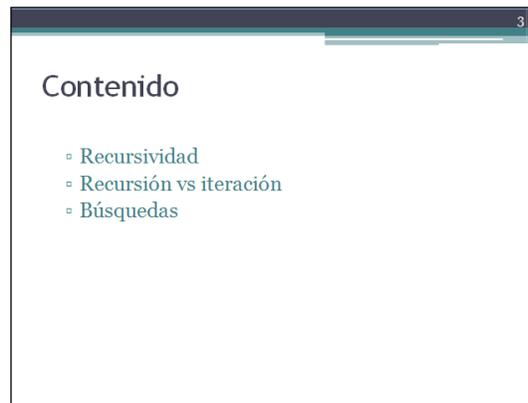
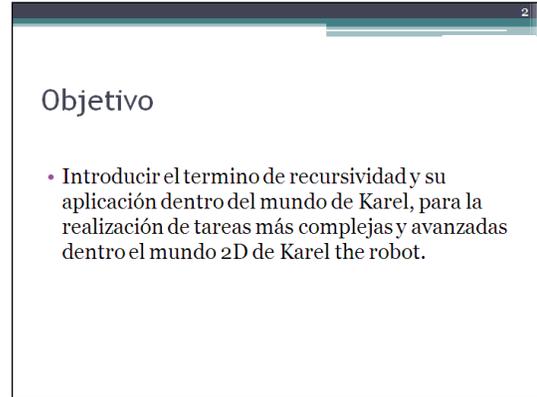
### CONTENIDO

- La Instrucción IF/THEN
- Condiciones conocidas
- La Instrucción IF/THEN/ELSE
- Instrucciones IF anidadas
- Transformando un IF complejo
- Un ELSE Indeciso

## Capítulo #8 Módulo de programación estructurada “Karel the robot”



## Capítulo #9 Módulo de programación estructurada “Karel the robot”





# APÉNDICE B

## Capítulo #1 Módulo de programación orientada a objetos “Alice”

1

### Conociendo el paradigma de la POO

Capítulo 1



2

### OBJETIVO

- Que el estudiante pueda entender y asimilar los conceptos del paradigma de programación orientada a objetos, con los conceptos básicos que rodean el paradigma.

3

### CONTENIDO

- ¿Qué es el paradigma de la POO?
- Historia
- Introducción a los conceptos del POO
- Conceptos fundamentales
  - Objeto
  - Clase
  - Método
  - Atributo

### CONTENIDO

- Características
  - Abstracción
  - Encapsulamiento
  - Polimorfismo
  - Herencia

## Capítulo #2 Módulo de programación orientada a objetos “Alice”

1

# Diseño e implementación de un programa

Capítulo 2

2

### OBJETIVO

- Que el estudiante conozca una manera de visualizar la solución de un problema dado por medio de escenarios y guiones visuales, es decir, diseñar la solución, para que más adelante se pueda implementar la solución en el mundo de Alice.

3

### CONTENIDO

- Escenarios y guiones
  - Conoce el escenario
  - Diseño
    - Guiones visuales
    - Guiones textuales
- Un primer programa
  - ¿Qué es un programa?
  - Traduciendo un guión a código de programa
  - Acciones secuenciales vs. simultáneas
  - Bugs
  - Usando propiedades
  - Comentarios

## Capítulo #3 Módulo de programación orientada a objetos “Alice”

1

# Uniendo todas las piezas

Capítulo 3

2

### OBJETIVO

- Que el estudiante conozca como combinar las diferentes “piezas” de código que ha aprendido en los capítulos previos con funciones predefinidas y estructuras de control para lograr que los programas hagan lo que se desea.

3

### CONTENIDO

- Funciones y expresiones predefinidas
  - Funciones
  - Expresiones
- Estructuras de control
  - Ejecución condicional
  - La necesidad de repeticiones

## Capítulo #4 Módulo de programación orientada a objetos “Alice”

1

# Clases, objetos, métodos y parámetros en Alice

Capítulo 4

2

### OBJETIVO

- Que el estudiante amplíe su conocimiento acerca de programación orientada a objetos, y aplique estos conceptos dentro del entorno de Alice para poder crear animaciones más complejas de forma más fácil utilizando los conceptos de clases, objetos, métodos, etc.

3

### CONTENIDO

- Definición clase, objeto y método.
- Métodos a nivel global
  - Métodos
  - Creando mis propios métodos
  - Invocando un método
- Parámetros
  - Parámetros
  - Objetos como parámetros
  - Múltiples parámetros
- Métodos a nivel de clase y herencia
  - Métodos a nivel de clase
  - Creando una nueva clase
  - Herencia

## Capítulo #5 Módulo de programación orientada a objetos “Alice”

1

# Interactuando: eventos y manejo de eventos

Capítulo 5



2

### OBJETIVO

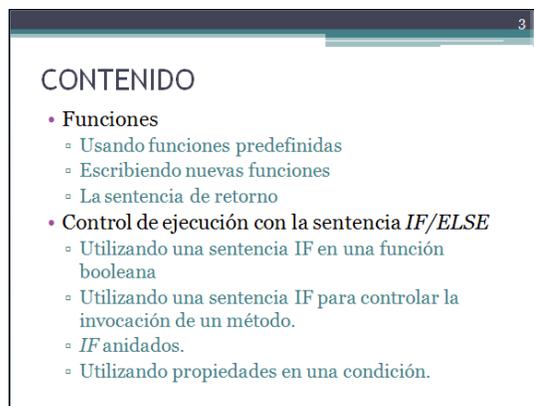
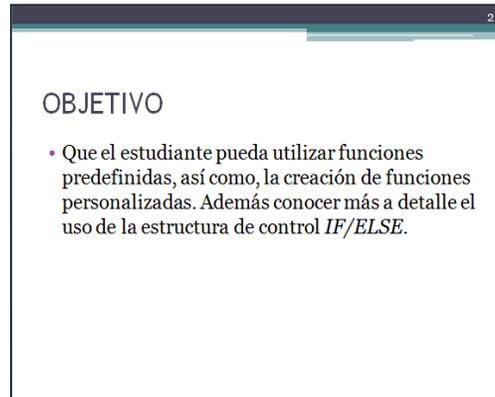
- Que el estudiante pueda crear, manejar y manipular eventos dentro del mundo de Alice por medio de objetos 3D.

3

### CONTENIDO

- Programación interactiva
  - Flujo de control
  - Eventos
  - Métodos para manejar eventos
  - Métodos que responden a eventos
  - Uniendo eventos con métodos
  - Parámetros y métodos para manejar eventos.

## Capítulo #6 Módulo de programación orientada a objetos “Alice”



## Capítulo #7 Módulo de programación orientada a objetos “Alice”

1

# Repetición: ciclos definidos y condicionales

Capítulo 7

2

### OBJETIVO

- Que el estudiante amplíe su conocimiento acerca de programación orientada a objetos, específicamente en el tema de ciclos definidos y condicionales, para que la resolución de problemas que implican la repetición de sentencias se resuelvan de formas más ordenada y rápida.

3

### CONTENIDO

- Ciclos definidos
  - Ciclos anidados
  - Ciclos infinitos
- While – Ciclo condicional

## Capítulo #8 Módulo de programación orientada a objetos “Alice”

1

# Repetición: recursión

Capítulo 8

2

### OBJETIVO

- Que el estudiante amplíe su conocimiento acerca de programación orientada a objetos, específicamente en el tema de recursión, para que la resolución de problemas que implican la repetición indefinida de sentencias se resuelvan de formas más ordenada y rápida.

3

### CONTENIDO

- Introducción a la recursión

## APÉNDICE C

### INSTRUCTOR GUIDELINES CONTENTS

<b>COURSE NAME:</b>	Introduction to a Programming Environment
<b>MODULE NAME:</b>	Introduction to Elemental Programming
<b>MODULE NAME:</b>	Introduction to Object Oriented Programming
<b>MODULE NAME:</b>	Project

FORM NO.: CGF01.01V01

## MODULE OVERVIEW

**COURSE NAME:** Introduction to a Programming Environment

**MODULE NAME:** Introduction to Elemental Programming

**PRE-REQUISITES:** Proficient in computer fundamentals

**OBJECTIVE:** To introduce the students to the basic programming statements, where they can create algorithms that control a simple robot with basic commands provided by the program Karel the Robot.

**TARGET AUDIENCE:** Under Graduates

**MODULE NAME:** Introduction to Object Oriented Programming

**PRE-REQUISITES:** Introduction to Elemental Programming

**OBJECTIVE:** To introduce the students to the concepts of the Object Oriented Programming by designing 3D animations with the help of a 3D environment provided by the program Alice.

**TARGET AUDIENCE:** Under Graduates

FORM NO.: CGF01.02V01

**MODULE BREAKUP FORM**

**COURSE NAME:** Introduction to a Programming Environment

<b>Modules</b>	<b>Theory Hrs.</b>	<b>Lab Hrs.</b>	<b>Total Hrs.</b>	<b>Assignment (No.)</b>	<b>Short Test (No.)</b>	<b>Homework (No.)</b>	<b>Project</b>	<b>Seminar</b>
Introduction to Elemental Programming	20	20	40	2	4	4	-	-
Introduction to Object Oriented Programming	28	28	56	2	4	4	-	-
<b>Total</b>	48	48	96	4	8	8	1	-

FORM NO.: CGF01.03V01

**MODULE EVALUATION FORM**

**COURSE NAME:** Introduction to a Programming Environment

**MODULE NAME:** Introduction to Elemental Programming

**INTERNAL WEIGHTAGE OF ASSIGNMENTS**

<b>ASSESSMENT S</b>	<b>NO. OF ASSESSMENTS</b>	<b>WEIGHTAGE</b>	<b>TOTAL WEIGHTAGE THEORETICAL CLASS</b>	<b>TOTAL WEIGHTAG E LAB</b>
Test	1	60%	60%	
Homework	4	10 %	40%	
Lab Assignment	2	30%		60%
Short Test	4	10 %		40%
Project	-	-	-	
Seminar	-	-	-	
<b>TOTAL</b>	<b>-</b>	<b>-</b>	<b>100%</b>	<b>100%</b>

**MODULE NAME:** Introduction to Object Oriented Programming

**INTERNAL WEIGHTAGE OF ASSIGNMENTS**

<b>ASSESSMENTS</b>	<b>NO. OF ASSESSMEN TS</b>	<b>WEIGHTAG E</b>	<b>TOTAL WEIGHTAGE THEORETICAL CLASS</b>	<b>TOTAL WEIGHTAG E LAB</b>
Test	1	60%	60%	
Homework	4	10 %	40%	
Lab Assignment	2	30%		60%
Short Test	4	10 %		40%
Project	-	-	-	
Seminar	-	-	-	
<b>TOTAL</b>	<b>-</b>	<b>-</b>	<b>100%</b>	<b>100%</b>

FORM NO.: CGF01.04V01

**SESSION DETAILS FORM**

**COURSE NAME:** Introduction to a Programming Environment  
**MODULE NAME:** Introduction to Elemental Programming  
**MODULE NAME:** Introduction to Object Oriented Programming

Session No.	Session Objective	Session No.	Session Objective	Session No.	Session Objective
1	Programming Concepts	2	Programming fundamentals	3	Karel the robot
4	Karel Handbook Session1	5	Primitive Instructions	6	Karel Handbook Session 2
7	Extending Karel's Vocabulary	8	Karel Handbook Session 3	9	Extending Karel's Vocabulary II
10	Karel Handbook Session 4	11	Conditionally Execution	12	Karel Handbook Session 5
13	First Evaluation	14	Instructions that repeat	15	Karel handbook session 6
16	Karel Advanced Techniques	17	Karel handbook session 7	18	Second Evaluation
19	Module Evaluation	20	Introduction to the Object Oriented Paradigm	21	Alice
22	Alice Handbook Session 1	23	Program design and implementation	24	Alice Handbook session 2
25	Putting all pieces together	26	Alice handbook session 3	27	Classes, objects, methods and parameters
28	Alice Handbook session 4	29	Events and event handling	30	Alice handbook session 6
31	First Evaluation	32	Functions and IF statement	33	Alice handbook session 7
34	Defined and conditional loops	35	Alice handbook session 7	36	Recursion
37	Alice handbook session 8	38	Second evaluation	39	Module Evaluation
40	Final Evaluation	41			

FORM NO.: CGF01.05V01

## SESSION BREAKUP

**COURSE NAME:** Introduction to a Programming Environment

**MODULE NAME:** Introduction to Elemental Programming

### MODULE OBJECTIVE:

*To enable the student to understand the principles and basics concepts of structured programming, applying those concepts in the 2D environment provided by the program Karel the Robot.*

### SESSION 1

**OBJECTIVE** Programming concepts

- What is programming?
- Program
- Algorithm
- Programming Language
- Pseudo code
- Abstraction
- Paradigm/Programming Paradigm
- Types of Programming Language
  - o Based on Abstraction
    - Machine Language
    - Low Level
    - Medium Level
    - High Level
  - o Based on Paradigm
    - Imperative

- Functional
- Logic
- Object Oriented
- Compilation
  - o Definition
- Types of Errors
  - o Lexical
  - o Syntactical

## **SESSION 2**

### **OBJECTIVE**

### **Programming Fundamentals**

- Program
  - o Variable
  - o Constant
  - o Data Types
  - o Identifier
  - o Keyword
- Operators
  - o Arithmetical
  - o Relational
  - o Logical
- Expressions
  - o Arithmetical
  - o Operators precedence

## **SESSION 3**

### **OBJECTIVE**

### **Karel the Robot**

- Introducing Karel's Environment
  - o Principal Elements
    - Karel
    - Cursor
    - Pared
    - Beeper
    - Streets and Avenues

- Panels
  - Initial World
    - Walls
    - Beepers
    - World File
    - Karel's Initial Situation
    - Cursor
    - Load, Save and Clear World
  - Program Panel
    - Open Program
    - New Program
    - Edit Program
    - Save Program
    - Compile
    - Save As...
    - Print Program
    - Abort Changes
  - Execution Panel
    - Step
    - Unstep
    - Run
    - Initialize
    - Steps/Min
    - Beepers in Bag
    - Next Instruction
    - World Low
    - Inside
    - Stack
- Example of a Program in Karel

#### **SESSION 4**

#### **OBJECTIVE**

**To introduce the student with Karel's environment.**

#### **SESSION 5**

#### **OBJECTIVE**

#### **Primitive Instructions**

- Changing Position
- Handling Beepers
- Finishing a Task
- My First Program

- Error-Shutoffs
- Programming Errors

## **SESSION 6**

**OBJECTIVE**      **To practice the primitive instructions.**

## **SESSION 7**

**OBJECTIVE**      **Extending Karel's Vocabulary**

- Creating a More Natural Programming Language
- A Mechanism that Defines New Instructions
- The Meaning of New Instructions
- Defining New Instructions

## **SESSION 8**

**OBJECTIVE**      **To apply the new instructions mechanism to extend Karel's vocabulary**

## **SESSION 9**

**OBJECTIVE**      **Extending Karel's Vocabulary II**

- Boxing: How Karel Understands a Program
- An Ungrammatical Program
- Tools for Designing and Writing Karel Programs
- Advantage of Using New Instructions
- Writing Understandable Programs

## **SESSION 10**

**OBJECTIVE**      **To practice the tools for designing and writing Karel programs.**

## **SESSION 11**

**OBJECTIVE**      **Conditionally Executing Instructions**

- The IF/THEN instruction
- The conditions Karel can test
- The IF/THEN/ELSE instruction
- Nested IF instructions
- Transformation for simplifying an IF
- The dangling ELSE

## **SESSION 12**

### **OBJECTIVE**

**To apply the conditional instructions to move Karel across the world.**

## **SESSION 13**

### **OBJECTIVE**

**Perform the first evaluation of the module.**

## **SESSION 14**

### **OBJECTIVE**

**Instructions that repeat**

- ITERATE instruction
- WHILE instruction
  - o Why the WHILE instruction is needed
- Errors to avoid with WHILE
- Nested WHILE loops
- WHILE and IF

## **SESSION 15**

### **OBJECTIVE**

**To apply the instructions that repeat to move Karel across the world.**

**SESSION 16**

**OBJECTIVE**      **Karel Advanced Techniques**

- Recursion
- Searches

**SESSION 17**

**OBJECTIVE**      **To practice Karel advanced techniques**

**SESSION 18**

**OBJECTIVE**      **To perform the second evaluation.**

**SESSION 19**

**OBJECTIVE**      **To perform the final evaluation of the module.**

**MODULE NAME:      Introduction      to      Object      Oriented  
   Programming**

## **MODULE OBJECTIVE:**

*To enable the student to understand the principles and basics concepts of object oriented programming, applying those concepts in the 3D environment provided by the program Alice.*

## **SESSION 20**

### **OBJECTIVE**

### **Introducing the Object Oriented Paradigm**

- History
- What is an object?
- Fundamentals
  - o Object
  - o Class
  - o Inheritance
  - o Method
  - o Event
- Characteristics
  - o Abstraction
  - o Encapsulation
  - o Polymorphism
  - o Inheritance

## **SESSION 21**

### **OBJECTIVE**

### **Alice**

- History
- Why "Alice"?
- Fundamentals
  - o Virtual World
  - o Objects and 3D models
  - o 3 dimensions and 6 directions
  - o The centre of the object
  - o Distance

- Position
- Animation

## **SESSION 22**

**OBJECTIVE**      **To explore the features of the virtual world of Alice.**

## **SESSION 23**

**OBJECTIVE**      **Program design and implementation**

- Scenarios and storyboards
  - Read the scenario
  - Design
    - Visual storyboards
    - Textual storyboards
- My first program
  - What is a program?
  - Translating the storyboard to program code
  - Sequential vs. Simultaneous
  - Bugs
  - Using a property
  - Comments

## **SESSION 24**

**OBJECTIVE**      **To apply the knowledge acquired to perform a first program.**

## **SESSION 25**

**OBJECTIVE**      **Putting all the pieces together**

- Functions and predefined instructions

- Functions
- Expressions
- Control structures
  - Conditional execution
  - The need of repetition

## **SESSION 26**

### **OBJECTIVE**

**To apply the functions and predefined instructions to handle a program with control structures.**

## **SESSION 27**

### **OBJECTIVE**

**Classes, objects, methods and parameters**

- Definition
- Global method
  - Methods
  - Creating my own methods
  - Calling a method
- Parameters
  - Definition
  - Objects as parameters
  - Multiple parameters
- Class methods and inheritance
  - Class level methods
  - Creating a new class
  - Inheritance

**SESSION 28**

**OBJECTIVE**            **To apply the concepts of class, method and parameters to a new program in the virtual world.**

**SESSION 29**

**OBJECTIVE**            **Interaction: Events and Handling Events**

- Interactive programming
  - o Control flow
  - o Events
  - o Methods to handle events
  - o Methods that respond to events
  - o Putting events and methods together

**SESSION 30**

**OBJECTIVE**            **To apply the events and handle of events to a new program in the virtual world.**

**SESSION 31**

**OBJECTIVE**            **To perform the first evaluation of the module.**

**SESSION 32**

**OBJECTIVE**            **Functions and if/else**

- Functions
  - o Using predefined functions
  - o Writing my own functions
  - o The return statement

- Execution control with if/else
  - o Using Boolean functions
  - o Expressions with multiple conditions
  - o Nested if

### **SESSION 33**

**OBJECTIVE**                    **To apply the functions and the execution control with the knowledge of the previous session.**

### **SESSION 34**

**OBJECTIVE**                    **Repetition: Defined and conditional loops**

- Defined loops
  - o Nested loops
  - o Infinite loops
- *While* – conditional loops

### **SESSION 35**

**OBJECTIVE**                    **To apply the repetition to control the flow of the elements in a scenario.**

### **SESSION 36**

**OBJECTIVE**                    **Repetition: Recursion**

- Introduction to recursion

### **SESSION 37**

**OBJECTIVE**                    **To apply the basic concepts of recursion in a new Alice scenario.**

**SESSION 38**

**OBJECTIVE**      **To perform the second evaluation of the module.**

**SESSION 39**

**OBJECTIVE**      **To perform the final evaluation of the module.**

FORM NO.: CGF01.08V01

**SEMINAR CONDUCT DETAILS**

**COURSE NAME:** Introduction to a Programming Environment

**MODULE NAME:** Introduction to Elemental Programming

**MODULE NAME:** Introduction to Object Oriented Programming

**MODULE NAME:** Project

**MAXIMUM MARKS:** -

**TOPICS TO BE ALLOTTED:** -

**TIME FOR EACH STUDENT:** -

**TOTAL MARKS PER STUDENT:-**

**EVALUATION CRITERIA:** -

FORM NO.: CGF01.09V01

**TEST CONDUCT DETAILS**

**COURSE NAME:** Introduction to a Programming Environment

**MODULE NAME:** Introduction to Elemental Programming

**MODULE NAME:** Introduction to Object Oriented Programming

**MAXIMUM MARKS:** 100

**TOPICS FOR TEST:** Entire module

**DURATION OF TEST:** 1 ½ - 2 hrs

**TEST BREAKUP:** Objective - 100 %

FORM NO.: CGF01.11V01

**SUGGESTED LIBRARY BOOKS**

**COURSE NAME:** Introduction to a Programming Environment

**MODULE NAME:** Introduction to Elemental Programming

2.1.1.1. NAME OF THE BOOK	2.1.1.2. AUTHOR	2.1.1.3. PUBLICATION
Karel The Robot: A Gentle Introduction to the Art of Programming	Richard E. Pattis	

**MODULE:** Introduction to Object Oriented Programming

2.1.1.4. NAME OF THE BOOK	2.1.1.5. AUTHOR	2.1.1.6. PUBLICATION
Learning To Program with Alice (2nd Edition)	Wanda P. Dann	

## APÉNDICE D

### Encuesta sobre preferencia en cursos

1. ¿Cuál es tu nombre?
2. ¿Qué edad tienes?
3. Sexo  
F\_\_\_\_\_ M\_\_\_\_\_
4. ¿En qué curso estarías interesado?
  - a) Windows 7 \_\_\_\_\_
  - b) Office 2010 \_\_\_\_\_
  - c) Google Avanzado \_\_\_\_\_
  - d) Diseño de Juegos \_\_\_\_\_
5. ¿Qué días te gustaría recibir el curso?
  - a) Lunes\_\_\_\_\_
  - b) Martes\_\_\_\_\_
  - c) Miércoles\_\_\_\_\_
  - d) Jueves\_\_\_\_\_
  - e) Viernes\_\_\_\_\_
  - f) Sábado\_\_\_\_\_
  - g) Domingo\_\_\_\_\_
6. ¿En qué horario te gustaría recibir el curso?
  - a) 8-9\_\_\_\_\_
  - b) 9-10\_\_\_\_\_
  - c) 10-11\_\_\_\_\_
  - d) 11-12\_\_\_\_\_
  - e) 12-13\_\_\_\_\_
  - f) 13-14\_\_\_\_\_
  - g) 14-15\_\_\_\_\_
  - h) 15-16\_\_\_\_\_

- i) 16-17\_\_\_
- j) 17-18\_\_\_
- k) 18-19\_\_\_
- l) 20-21\_\_\_

7. ¿Cuánto estarías dispuesto a pagar por un curso?

- a) Q0.00-5.00 quetzales\_\_\_
- b) Q5.00-10.00 quetzales\_\_\_
- c) Q10.00-15.00 quetzales\_\_\_
- d) Q15.00-20.00 quetzales\_\_\_
- e) Q20.00-25.00 quetzales\_\_\_
- f) Q25.00-30.00 quetzales\_\_\_
- g) Q30.00-35.00 quetzales\_\_\_
- h) Q35.00-40.00 quetzales\_\_\_
- i) Q40.00-45.00 quetzales\_\_\_
- j) Q45.00-50.00 quetzales\_\_\_

8. ¿Cuál es la duración razonable de un curso para ti?

- a) 3 días\_\_\_
- b) 1 Semana\_\_\_
- c) 15 Días\_\_\_
- d) 1 Mes\_\_\_

9. ¿Qué incentivo te gustaría recibir?

- a) Puntos\_\_\_
- b) Créditos\_\_\_
- c) Por Tarea\_\_\_
- d) Únicamente Diploma\_\_\_

10. ¿Hablas inglés?

- a) Muy Poco\_\_\_
- b) Regular\_\_\_
- c) Bien\_\_\_
- d) Excelente\_\_\_

11. ¿Dónde usas más frecuentemente el Internet?

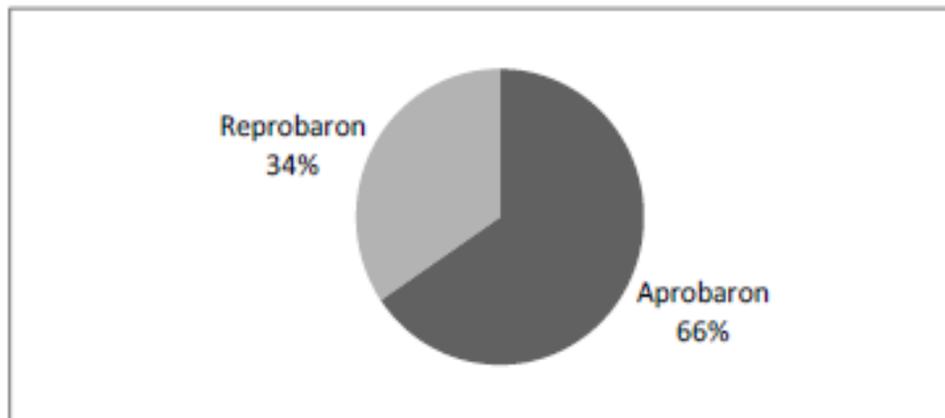
- a) Mi casa\_\_\_\_
- b) Laboratorio Gratuito de la Universidad\_\_\_\_
- c) Café Internet de la Universidad\_\_\_\_
- d) Café Internet Fuera de la Universidad\_\_\_\_
- e) Mi Celular\_\_\_\_
- f) En el Trabajo\_\_\_\_



## ANEXO A

### Datos estadísticos de la primera iteración de IPC1

Figura 11. **Porcentaje de estudiantes aprobados y reprobados en IPC1 dentro del *ITCoE* en el 2do. semestre 2008**



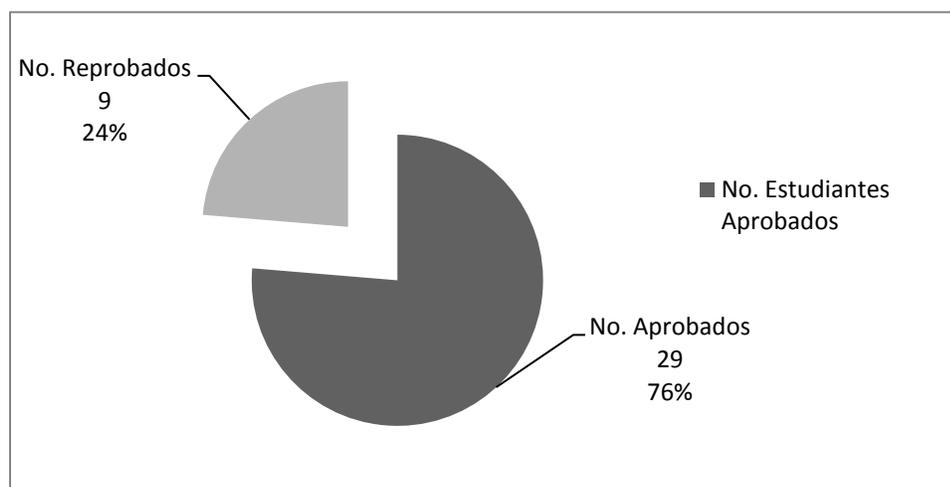
Fuente: Ramírez, Adín y Paniagua Carlos (2008). "Implementación de la metodología del IT Education Centre INDIA-GUATEMALA". Página 90.



## ANEXO B

### Datos estadísticos de la segunda iteración de IPC1

Figura 12. **Porcentaje de estudiantes aprobados y reprobados en IPC1 dentro del *ITCoE* en el 1er. semestre 2009**



Fuente: Hernández, César y Rodas, David (2009). "Análisis, implementación y evaluación de la metodología utilizada en el proceso de enseñanza de tecnología implementada en el *ITCoE*".

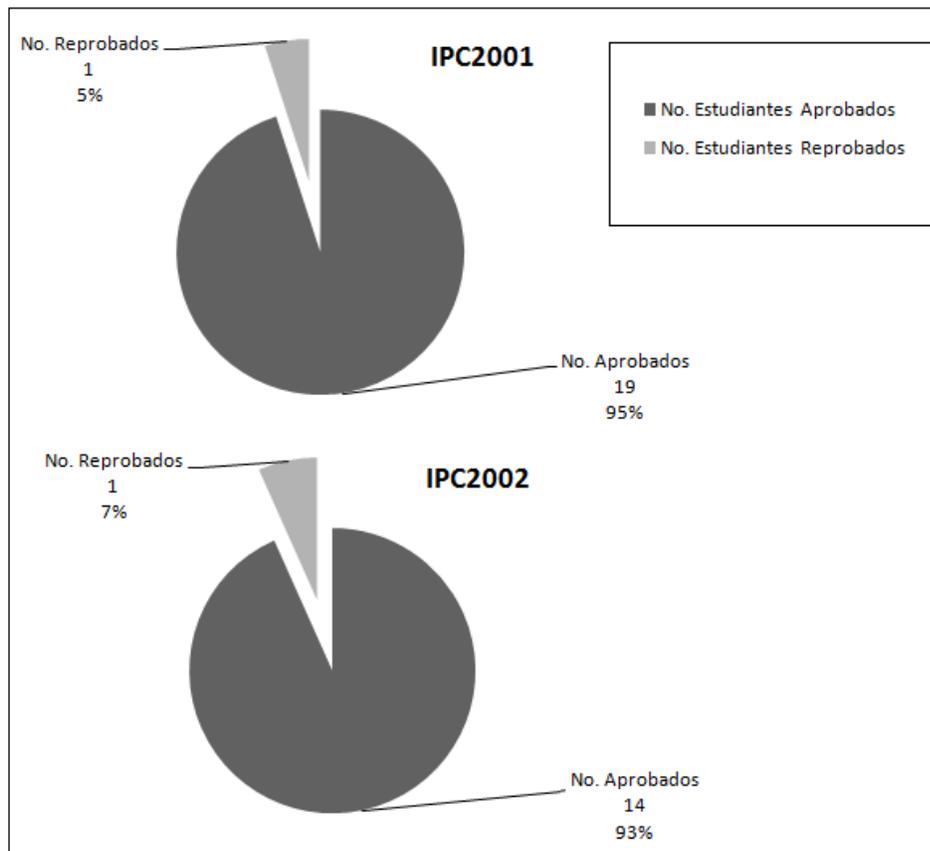
Página 11.



## ANEXO C

### Datos estadísticos de la primera iteración de IPC2

Figura 13. **Porcentaje de estudiantes aprobados y reprobados en IPC2001 e IPC2002 dentro del ITCoE en el 1er. semestre 2009**



Fuente: Hernández, César y Rodas, David (2009). "Análisis, implementación y evaluación de la metodología utilizada en el proceso de enseñanza de tecnología implementada en el ITCoE".



## ANEXO D

### 2. METODOLOGÍA ACADÉMICA DEL *IT EDUCATION CENTRE* *OF EXCELLENCE*<sup>5</sup>

El *ITCoE* se basa en un proceso académico bien definido, gracias a las buenas prácticas heredadas por TCS. Dicha empresa es fuerte y sólida en sus procesos al contar con un nivel de Capacidad Madurez 5 (CMM5), lo que garantiza alto grado de calidad a nivel de procesos.

TCS ha implantado en el *ITCoE* las mejores prácticas para asegurar la calidad en los procesos administrativos y académico. Estas mejores prácticas se traducen en una metodología fuertemente orientada a la documentación, donde las actividades deben estar planificadas y se fomenta la mejora continua a través de retroalimentación y análisis de resultados mensuales tanto del personal como del centro entero.

A continuación detallaremos un proceso maduro sobre la forma en que se manejan los cursos y la forma en que se debe de proseguir al estar impartiendo uno de ellos. Se detallan las formas relevantes utilizadas en el proceso académico del *ITCoE*, entre ellas: *Instructor Guidelines*, *Daily Attendance and Rercord Status*, *Attendance and Marks* y *el Examination Record Status*.

---

<sup>5</sup>Fuente: Ramírez, Adín y Paniagua Carlos (2008). "Implementación de la metodología del IT Education Centre INDIA-GUATEMALA". Páginas 13 - 40.

## 2.1 Proceso capacitación

### CAPACITACIÓN

#### Involucrados:

- Centre Manager
- Technical Trainer
- Beneficiado

#### Entradas

- Un conjunto de estudiantes a capacitar

#### Salidas

- Un conjunto de estudiantes capacitados en tecnología
- Una forma CEF1050B con solidando las notas de cada estudiante
- Formas CEF1050A con las notas de cada actividad de evaluación realizada
- Forma CEF1020A con la planificación y asistencia de los estudiantes (una por cada mes)
- Las formas de ingreso de cada estudiante
- Un listado con los estudiantes inscritos

#### Formas involucradas:

#### Recursos

- *Technical trainer*
- Un laboratorio/salón dependiendo de la naturaleza del curso
- *Hardware y software* necesarios para la capacitación
- El IG del curso
- El *handbook* del curso
- Las diapositivas del curso
- El libro del curso

## Descripción:

- **Creación de listado de estudiantes**

El primer día de la capacitación el *technical trainer* debe crear una forma CEF1020A con los datos de los estudiantes en formato digital.

- **Enseñanza sesión teórica**

El *Instructor Guideline* de la mayoría de los cursos está dividido en sesiones prácticas y sesiones teóricas. Una sesión teórica se basa en las diapositivas del curso y una sesión teórica del IG corresponde a un conjunto de *slides* (presentación). Así mismo un capítulo del libro esta mapeado a una presentación. El *Technical Trainer* debe impartir la sesión teórica y todo debe realizarse bajo la filosofía de enseñanza del *ITCoE*.

- **Enseñanza sesión practica**

Por cada sesión teórica se tiene comúnmente una sesión práctica que permita a los estudiantes poner en práctica los conceptos aprendidos en la sesión teórica correspondiente. Generalmente una sesión práctica se realiza luego de una sesión teórica y la relación entre ellas es de uno a uno. La sesión teórica se encuentra descrita en el *handbook* del curso y debe ser realizada con el mismo tiempo que una sesión teórica. Tanto las sesiones teóricas como las sesiones prácticas se encuentran documentadas en el IG del curso. Estas sesiones al igual que las teóricas deben realizarse bajo la filosofía de enseñanza del *ITCoE*.

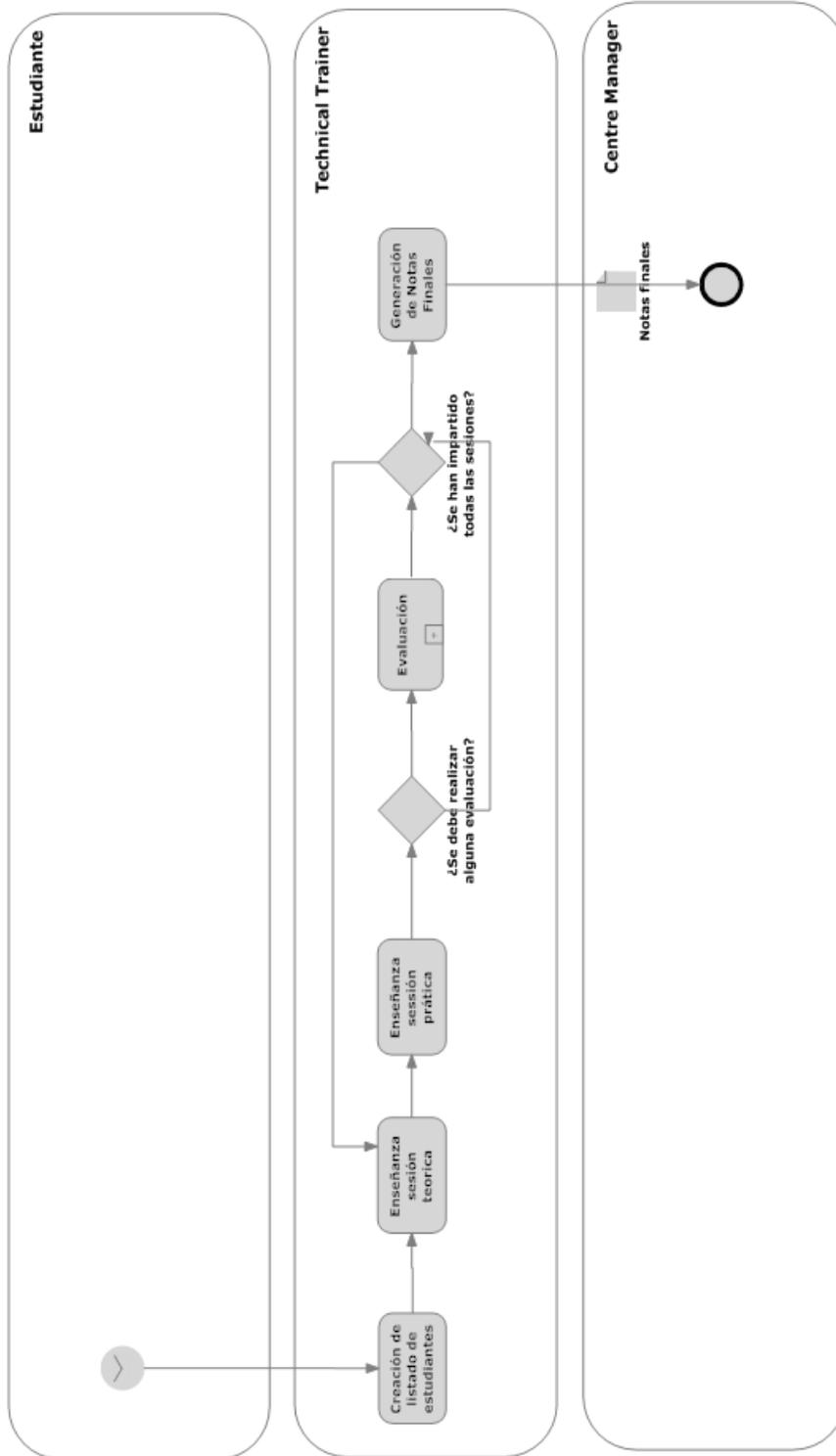
- **Evaluación**

En el instructor *guideline* están definidas las actividades de evaluación que deben realizarse durante el curso. La información completa de la evaluación se especifica en el IG (sesión cuando debe realizarse, la manera de calificación, tiempo de la prueba, contenidos, etc.). Estas actividades generalmente tienen alguna correspondencia con la nota final del curso.

- **Generación de notas finales**

El *Technical Trainer* al final del curso debe consolidar los resultados de todas las actividades de acumulación de puntos y consolidarlos en la forma CEF1050B y generar la nota final del estudiante, nota que será utilizada para saber si el estudiante aprobó o reprobó el curso. Estas notas deben ser enviadas al Centre Manager en formato digital y en formato físico en los próximos 5 días hábiles después de concluido el curso. Las notas también deben ser enviadas a los estudiantes por correo electrónico en formato PDF no modificable.

Figura 1. Proceso capacitación



## 2.2 Proceso de evaluación

### EVALUACIÓN

#### Involucrados:

- *Technical Trainer*
- Estudiante

#### Formas involucradas

- CEF1050A

#### Entradas

- Un enunciado de evaluación
- Un conjunto de estudiantes a evaluar

#### Recursos

- *Technical trainer*
- Un laboratorio o salón dependiendo de la naturaleza del curso
- *Hardware* y *software*(dependientes de la naturaleza del curso)
- *Handbook*
- IG del curso

#### Salidas

- Una evaluación realizada, calificada y revisada, debidamente documentadas

## Descripción:

- **Imprimir forma de control**

El *trainer* debe crear una copia física de la forma CEF1050A para llevar control de la evaluación.

- **Solicitar forma de asistencia**

El *trainer* solicita una firma de asistencia a los estudiantes para tener registro de los estudiantes que están presentes en la prueba.

- **Firma de asistencia**

Los estudiantes presentes en la prueba deben firmar la hoja de control(CEF1050A) para tener derecho a la prueba y poder realizarla.

- **Realización de la prueba**

Es la realización de la prueba como tal. Los detalles de la misma están especificados en el IG del curso.

- **Revisión de la prueba**

El *technical trainer* debe calificar la prueba en base a los criterios establecidos en el IG documentando los resultados en la forma CEF1050A.

- **Entrega de notas**

El *technical trainer* entrega los resultados a los estudiantes por correo en formato PDF no modificable. Las notas son validadas por el *technical trainer* y el *Centre Manager* de forma física y el *technical trainer* las envía de forma digital en formato de Excel al *Centre Manager*.

- **Solicitar firma de conformidad**

El *technical trainer* solicita a los estudiantes una firma de conformidad con la nota obtenida. Si los estudiantes solicitan revisión esta debe ser dada por el *technical trainer* y luego documentarla en la forma CEF1050A con la nueva nota y posteriormente el estudiante debe firmar la forma de conformidad.

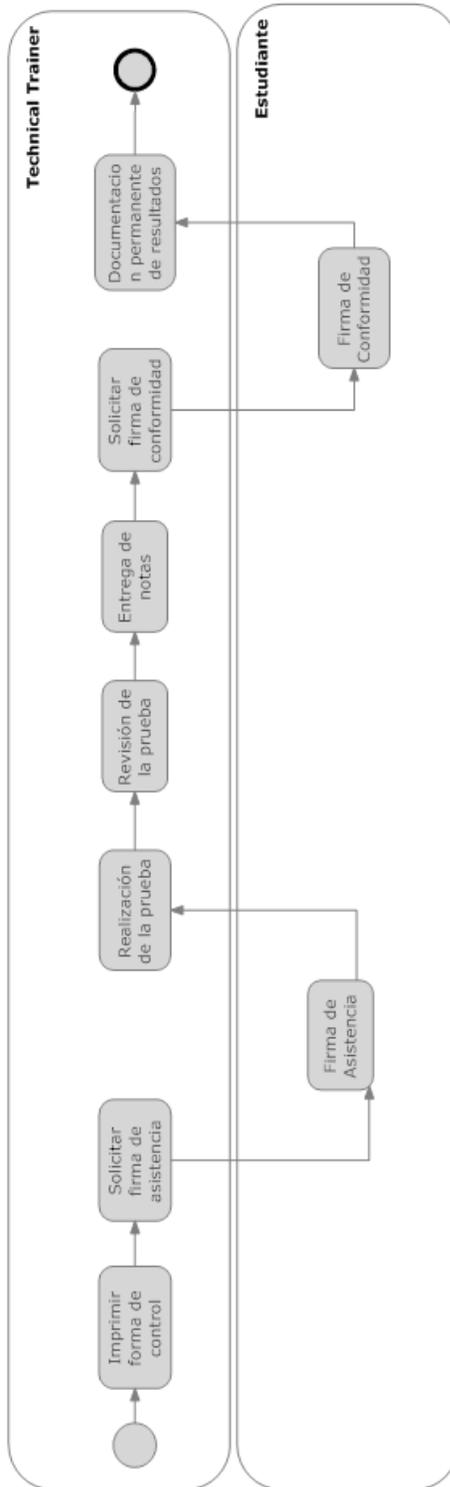
- **Firma de conformidad**

Los estudiantes firman la forma CEF1050A de conforme con la nota obtenida en la prueba.

- **Documentación de resultados permanentemente**

El *technical trainer* documenta la evaluación en las formas involucradas y se alimentan de la forma CEF1050A. La documentación se realiza en las formas CEF1050B (reporte de notas finales) y la forma CEF1070A (reporte mensual del *trainer*).

Figura 2. **Proceso evaluación**



## **2.3 Instructor Guidelines**

Como parte de los procesos sobre los que descansa el *ITCoE* y de su fuerte documentación los *Instructor Guidelines* (IG) son documentos que contienen la manera en que un curso debe ser llevado a cabo. Estos son resultado de las metodologías tanto operativo como de enseñanza del *ITCoE*.

Por el lado operativo el *Centre Manager* encuentra en el *Instructor Guideline* una herramienta de planificación, sabiendo cuantas horas se deben invertir en diferentes actividades del curso y programar el consumo de recursos tanto humanos como tecnológicos. Para los *Technical Trainers* el *Instructor Guidelines* representa su mapa de ruta de cómo conducir el curso.

El valor del *Instructor Guidelines* es más representativo desde el punto de vista académico que operativo, pero forma parte importante de ambas metodologías.

### **2.3.1 Codificación**

Todas las secciones del IG están codificadas para su fácil interpretación y seguimiento en algún momento dado. Esta codificación está dada por un correlativo de tres letras y dos números que indican el centro donde se imparte y creó el documento, i.e. CGF01, por ser el primer centro de Guatemala. Seguidas de un número de sección y versión separados por un punto, i.e. 01V02, indica que es la primera sección y en su segunda versión.

Así un código válido es **CGF01.01V05**, lo cual indica que es un documento del primer centro y se está en la primera sección del documento y en su quinta versión. El listado de códigos de sección se puede apreciar en la Tabla II.

Tabla II. Listado de secciones de un *Instructor Guideline*

<b>Código</b>	<b>Nombre de la Sección</b>	<b>Detalle</b>
<b>01</b>	<i>Module Overview</i>	Descripción de los módulos del curso.
<b>02</b>	<i>Module Breakup</i>	Detalle de los módulos del curso: horas, exámenes, pruebas,
<b>03</b>	<i>Module Evaluation</i>	Detalle de la ponderación y cantidad de actividades de cada
<b>04</b>	<i>Sessions Details</i>	Lista de todas las sesiones en el curso con una descripción opcional.
<b>05</b>	<i>Session Breakup</i>	Detalle de los tópicos a abordar en cada
<b>06</b>	<i>Assignments</i>	Detalle de las tareas y de los ejemplos de las mismas.
<b>07</b>	<i>Tutorial Conduct</i>	Detalle de los tutoriales y de los ejemplos de los mismos.
<b>08</b>	<i>Seminar Conduct</i>	Detalle de los seminarios y de los ejemplos de los mismos.
<b>09</b>	<i>Test</i>	Detalle de los exámenes y de los ejemplos de los mismos.
<b>10</b>	<i>Project Conduct</i>	Detalle de los proyectos y ejemplos de los mismos.
<b>11</b>	<i>Suggested Books</i>	Detalle de los libros sugeridos por

## 2.3.2 Secciones

### 2.3.2.1 Descripción

Consiste en una descripción del curso, de los temas que se tocarán y los módulos que lo forman parte. Se describe de forma general el contenido del curso y los conocimientos tecnológicos que se buscan transferir.

### 2.3.2.2 Encabezado

El encabezado contiene el nombre del curso así como los módulos que lo conforman. Por ejemplo:

COURSE NAME:	Software Engineering and OOAD
MODULE NAME:	Introduction to Software Engineering

En este ejemplo, el nombre del curso es *Software Engineering and OOAD* y está compuesto por un solo módulo llamado *Introduction to Software Engineering*. Si existieran más de un módulo estos también aparecerían inmediatamente después del primero y así sucesivamente. Inmediatamente después del encabezado aparece el índice del IG.

### 2.3.2.3 Course Overview

En esta sección del IG se describen los módulos que conforman al curso. Para el curso se describe:

- Nombre del curso
- Pre requisitos del curso
- Objetivo del curso
- Descripción de cada módulo que forma parte del curso

#### **2.3.2.4 Course BreakUp Form**

En esta sección se describe cada módulo y las actividades que están involucradas en dicho módulo. Se especifica:

- El número de horas que debe invertirse para impartir clases teóricas.
- El número de horas que se debe invertir para impartir clases prácticas (laboratorios, ejercicios, etc.)
- La cantidad de laboratorios que se deben realizar en el módulo.
- La cantidad de exámenes que se deben realizar en el módulo.
- La cantidad de actividades que se deben realizar fuera del laboratorio (*Home Tutorials*).
- La cantidad de exposiciones (*Seminars*).
- La cantidad de cualquier otra actividad contemplada dentro del curso.

#### **2.3.2.5 Course Evaluation Form**

En esta sección se describe la forma en que el curso debe evaluarse por módulos. Para cada módulo se describen las actividades de evaluación que lo forman y el porcentaje de la nota del curso que representan.

### **2.3.2.6    *Session Detail Form***

En esta sección define que actividades deben realizarse en cada sesión. Cada sesión puede ser invertida en

- Prácticas de laboratorio, ejercicios, proyectos, etc.
- Clases teóricas
- Exámenes
- Seminarios

Este detalle define el camino que debe llevar el curso. Este plan de base se utiliza para programar las actividades del mes en la forma CEF1020A.

### **2.3.2.7    *Session BreakUp***

Debido a que el *ITCoE* trabaja sobre un proceso de enseñanza bien definido se detalla que temas deben darse en cada sesión. En el *Session BreakUp* se describen por sesión los temas que se deben impartir en ella.

### **2.3.2.8    *Lab Assignments Details***

Se describe con detalle las tareas prácticas. Para cada tarea se detalla:

- Nombre del curso a la que pertenece
- Nombre del módulo del curso al que pertenece
- Número de laboratorio
- Objetivo de la tarea
- Sesión donde se entrega a los estudiantes la tarea para su realización

- Sesión donde los estudiantes deben entregar la tarea resuelta
- Nota máxima que se puede obtener en la realización del tutorial
- El criterio de evaluación de la tarea. Los aspectos que se deben evaluar y cuanta ponderación se le debe dar a cada uno de los mismos. Porejemplo en una práctica de laboratorio podría evaluarse la funcionalidad de la práctica y darle una ponderación de 20%. Esto significa que se debe calificar cuan funcional es la resolución del estudiante a una nota correspondiente al 20% de su nota total
- Documentación. Si el estudiante como parte de la tarea de laboratorio debe entregar o no documentación. Se considera como documentación tutoriales, manuales de usuario, manuales técnicos, etc. Cualquier documento que describa y tenga relación con la práctica que se realiza

### **2.3.2.9     *Lab Assignments Examples***

Se muestra un ejemplo del *Lab Assignment* (tarea de laboratorio). Se muestra un enunciado de ejemplo así como los requerimientos que debería de tener una tarea de este tipo. Se especifica el módulo a la que pertenece y la nota máxima que es posible alcanzar con la realización del laboratorio.

### **2.3.2.10    *Tutorial Conduct Details***

Se describe la forma en que se debe llevar a cabo un tutorial dentro del curso. Se considera tutorial cualquier investigación, trabajo de campo, manual, tutorial, etc. Para cada tutorial se detalla:

- Nombre el curso al que pertenece.
- Nombre del módulo del curso al que pertenece.

- Nombre o número del tutorial.
- Objetivo del tutorial.
- Nota máxima que se puede obtener en la realización del tutorial.
- Sesión donde se entrega a los estudiantes el enunciado del tutorial para su realización.
- Sesión donde los estudiantes deben entregar el tutorial resuelto.
- Metodología. Se especifican los temas que deben ser incluidos dentro del tutorial.
- El criterio de evaluación del tutorial. Los aspectos que se deben evaluar y cuanta ponderación se le debe dar a cada uno de los mismos. Por ejemplo en un tutorial podría evaluarse el conocimiento técnico y darle una ponderación de 20%. Esto significa que se debe calificar el conocimiento técnico plasmado en el tutorial a una nota correspondiente al 20% de su nota total.
- Documentación. Si el estudiante como parte del tutorial debe entregar documentación, de que tipo y en que formato debe ser entregada.

#### **2.3.2.11 *Tutorial Example***

Se muestra un ejemplo del Tutorial (tutorial o investigación). Se muestra un enunciado de ejemplo, así como los requerimientos. Se especifica el módulo a la que pertenece y la nota máxima que es posible alcanzar con la realización del tutorial.

### **2.3.2.12 Seminar Conduct Details**

Se considera seminario a todas aquellas actividades que involucren exposición de algún tema técnico o teórico. Por ejemplo, exposiciones, presentaciones, debates, etc. Para cada *Seminar* se describe:

- Nombre del módulo al que pertenece.
- La nota máxima que se puede obtener en el seminario.
- Los temas que deben ser incluidos en el seminario.
- Tiempo que cada estudiante tiene para realizar el seminario.
- La nota máxima que puede obtener un estudiante. En caso el seminario sea en grupo la nota podría estar dividida en una nota individual y una grupal. En ese caso la nota máxima que puede alcanzar un estudiante de manera individual se define en esta sección, y la nota máxima (tanto grupal como individual) se define en el segundo ítem de esta lista.
- Se define el criterio de evaluación que se debe de utilizar al calificar el seminario. Se detallan cada uno de los aspectos que se deben de tomar en cuenta así como su valor asociado en la nota.

### **2.3.2.13 Test Conducts Details**

Para cada examen que se deba realizaren el curso se detalla

- Nombre del módulo al que pertenece el examen
- La nota máxima que pueden alcanzar en el examen
- El listado de temas que serán evaluados en el examen
- La duración en horas del examen

- La forma en que estará estructurado el examen, su forma de evaluación y la nota asociada a cada parte del mismo. Por ejemplo un examen puede tener dos secciones. La primera utiliza una forma de evaluación objetiva realizando preguntas de opción múltiple; y la segunda parte podría utilizar una forma de evaluación subjetiva pidiendo al estudiante desarrollar algún tema o realizar algún tipo de análisis técnico/teórico. Asignándole un porcentaje sobre la nota a cada parte, un 40 % para la parte objetiva y un 60 % para la parte subjetiva, por ejemplo.

#### **2.3.2.14 *Sample Test Paper***

Se da un examen de ejemplo con el objetivo de guiar al trainer en la realización del mismo. Este examen tiene como objetivo ser una muestra de la estructura que debe seguir un examen así como la forma en que deben ser formuladas las preguntas.

#### **2.3.2.15 *Project Conducts Details***

Si un curso demanda la realización de un proyecto los detalles de cómo llevarlo a cabo se definen en el IG. Para cada proyecto se describe lo siguiente:

- Nombre del módulo.
- Nota máxima que puede ser obtenida en el proyecto.
- Objetivo que se persigue con la realización del proyecto.
- Si el proyecto está hecho para realizarse en grupos de trabajo se define la cantidad de personas por grupo.
- La sesión donde se debe entregar el enunciado a los estudiantes.
- La sesión donde los estudiantes deben entregar el proyecto resuelto.

- El criterio de evaluación que se debe utilizar para calificar el proyecto. Se define cada aspecto que debe calificarse así como su ponderación asociada a la nota del proyecto. Por ejemplo si el proyecto consiste en desarrollo de un *software* y este requiere ser presentado de manera formal se podría pensar en asignar un 40% de la nota a la funcionalidad del *software*, 20% a la documentación del mismo y 20% a la exposición del *software* en su versión final (VIVA).

#### **2.3.2.16 Suggested Library Books**

En esta sección se describen material bibliográfico que puede ser utilizado por el *Technical Trainer* como apoyo para impartir el curso.

#### **2.4 Daily Attendance/Session Record**

Esta forma constituye una de las más importantes dentro de la metodología académica del *ITCoE*. La forma de Asistencia Diario se plantea un documento de control para el *Trainer* sobre el curso que realiza. Así mismo, le permite al *Centre Manager* tener una idea muy precisa del flujo que lleva el curso y la forma en que se está realizando. A través de esta forma el *Centre Manager* y el *Trainer* pueden evaluar la velocidad en que se imparte el curso, planificarse y hacer ajustes con medidas correctivas en caso exista alguna situación anormal.

Esta forma le permite al *Trainer* tener control de la asistencia de sus estudiantes al curso y la frecuencia de la misma. Esta forma se debe realizar una vez al mes por cada curso ya que ésta le permite planificar las sesiones en el mes, y durante el mes tener control de si esa planificación se está cumpliendo o no.

### 2.4.1 Funciones de control

- Tener un mapa de los días del mes efectivos para las capacitaciones.
- Tener un control diario de la asistencia de los estudiantes.
- Tener un control de las sesiones que se llevan a cabo (en qué días).
- Tener una idea clara si la planificación se cumple.
- Planificar las sesiones del mes (en qué días se impartirán obviando los días que no son hábiles).
- Tener un control de cuantas sesiones se imparten al mes.
- Tener un control holístico entre asistencia y planificación durante un mes en un curso.

### 2.4.2 Secciones

#### 2.4.2.1 Encabezado

En esta sección se especifica la siguiente información:

- El código del Centro. Para el centro de Guatemala el código es ***ITCoE, Guatemala.***
- El código del curso al que pertenece la forma. Por ejemplo ISE001 implica que la forma pertenece al código de ISE en su primera vez impartido.

- Módulo. Algunos cursos están compuestos por más de un módulo, en esta sección se debe color el módulo del curso en el que se encuentra el curso.
- *Trainer*. El nombre del *Trainer* que está impartiendo el curso.
- Mes. Esta forma es válida una por mes, es decir, que por cada mes se debe realizar una forma de asistencia. Para cada forma se debe especificar a qué mes corresponde.
- El estatus del flujo del curso en relación a la planificación de las sesiones y como estas se están llevando. Esto permitirá saber cuántas horas extras se necesitaran para completar el curso o bien cuantas horas se han ahorrado en el curso.

#### **2.4.2.2 Cuerpo**

Se tiene una tabla cuyo encabezado está formado por:

- Listado de estudiantes junto a su identificador.
- Un listado de días del mes, cada columna corresponde a un día del mes.
- Un total de sesiones impartidas al mes y a la cantidad de sesiones a las cuales asistió cada estudiante.

El cuerpo de la tabla está formado por:

- La asistencia por día de cada estudiante. Se toma asistencia con una letra (P o A) en la casilla correspondiente al estudiante y al día en que se toma la asistencia. Si el estudiante está presente se coloca una P (Presente) si no asistió a la sesión se coloca una A (Ausente).

- Todos los días que sean descansos o días no hábiles deben ser marcados como tales y llevaran una breve frase que describa porque fue día no hábil (por ejemplo, sábado, domingo, edificio cerrado, día de congreso, etc.).
- Un total de las sesiones dadas en el mes por estudiante y un total de las sesiones a las que el estudiante asistió.
- Un total de horas impartidas y un total de horas de retraso que se tienen en el curso (si se tuvieran horas de retraso).

### 2.4.2.3 Forma de utilización

- Al inicio del mes el *Trainer* debe planificar las sesiones que impartirá en el mes. Se tienen tres secciones
  - **Session Master Plan.** Es el flujo ideal que debería tomar un curso tomando en cuenta los días de descanso (excepto sábados y domingo, suponiendo que se trabaja a una semana de 5 días sin interrupciones).
  - **Session Planned.** Es el flujo de sesiones que se planifica tomando en cuenta los días en los que no se podrá impartir el curso. Si un día es descanso o no es posible impartir el curso no se planifica sesión para este día, solo para aquellos días hábiles.
  - **Session Complete.** Este flujo representa la manera en que las sesiones se imparten y debe ser llenada conforme las sesiones son impartidas. En caso de que algo anormal suceda y la sesión debe suspenderse se llenara con un "-". Por cada sesión que no se

lleve a cabo según la planificación del *Session Planned* se provocará *overflow* sobre el curso, es decir, horas de retraso. En base a las *Session Complete* se contabilizan las sesiones dadas por curso para el mes.

- Al final del mes el *trainer* debe contabilizar las sesiones que impartió.
- Al final del mes el *trainer* contabiliza las sesiones a las que asistió cada estudiante.
- Al final del curso el *trainer* contabiliza el total de *overflow* del mes y el acumulado (de meses anteriores en que se impartió el curso)
- Por cada día que se imparte una sesión el *trainer* debe tomar la asistencia colocando en la casilla del estudiante una P si está presente o una A si está ausente.

## **2.5 Attendance and Marks Record**

Esta forma es resultado de un proceso maduro de enseñanza, es especialmente útil cuando se realizan actividades de evaluación en un curso. Por cada actividad de evaluación en un curso (sea válida por puntos o no) se debe crear una forma de este tipo que funcione como respaldo tanto para el estudiante como para el *Trainer*.

Por ejemplo, si para un curso se debe realizar un examen final para éste el *Trainer* debe crear una forma siguiendo este formato, donde se especifiquen los detalles de la prueba, un listado con los estudiantes inscritos en el curso, notas obtenidas en el curso y firmas tanto durante y después de la prueba por parte de los estudiantes.

### **2.5.1 Funciones de control**

- Tener un control de la cantidad de estudiantes que realizan la prueba a través de una firma de cada estudiante al realizarse la prueba.
- Tener un control de la cantidad de estudiantes que reciben su calificación de la prueba y están conformes con la misma, incluyendo a quienes han hecho uso de su derecho a revisión. Si un estudiante firma por segunda vez la hoja de control de notas acepta su calificación y su derecho a revisión en caso no lo haya utilizado todavía.
- Permite al *trainer* el tener control de las notas obtenidas por sus estudiantes en las actividades de evaluación para utilizarlas como entradas a la hora de preparar las notas finales del curso.
- Tener un control del performance en cada evaluación (*Performance Indicator*).

- Tener un control sobre el curso al que pertenece la prueba.
- Tener control de quien realiza la prueba.
- Tener control de quien califica la prueba.
- Tener control de la nota máxima que se puede obtener en la prueba.
- Tener control de las fechas en que se califican y realizan las pruebas.

## 2.5.2 Secciones

### 2.5.2.1 Encabezado

En esta sección se describe información como:

- Código del Centro. En el caso de Guatemala el código es ***ITCoE, Guatemala.***
- Nombre del curso en el que se realiza la prueba.
- Módulo del curso en el que se realiza la prueba.
- Tipo de prueba que se está realizando.
- Nota máxima que se puede obtener en la prueba.
- Fecha en que se realiza la prueba.

### 2.5.2.2 Tablas

Se tiene una tabla donde cada fila está compuesta por

- Un número correlativo
- El identificador del estudiante
- Nombre del estudiante

- Firma del estudiante en el momento de realizar la prueba (asistencia)
- Nota que el estudiante obtuvo en la prueba
- Firma del estudiante de conformidad con su nota (si esta firma se realiza por parte del estudiante se supone conforme con la nota y no podrá hacer uso de su derecho de revisión o bien ya lo hizo efectivo)

### **2.5.2.3 Resumen**

En el resumen se coloca información respecto a la calificación de la evaluación y acerca del rendimiento de la prueba

- Calculo del PI (*Performance Indicator*)
- Porcentaje de estudiantes que aprobaron la evaluación
- Nombre de la persona encargada de pasar la prueba y estar presente en la misma
- La firma de la persona encargada de pasar la prueba
- Fecha en que se realizó la prueba
- Nombre de la persona que califica la prueba
- Fecha en que la prueba es calificada
- Firma del *Centre Manager*
- Fecha en que el *Centre Manager* recibe la prueba de recibido

## **2.6 Examination Record Status**

Esta forma se utiliza para tener un consolidado de todas las actividades de evaluación y permite obtener las notas finales para curso. El *trainer* con base a las formas CEF1050A (formas para actividades de evaluación) consolida los

resultados en esta forma asignándole a cada una el porcentaje sobre la nota final que le corresponde y obtiene las notas finales del curso.

### **2.6.1 Funciones de control**

- Tener consolidadas las notas de todas las actividades de evaluación realizadas en el curso
- Tener de manera física y digital las notas finales del curso
- Proveer una herramienta de control para el Centre Manager sobre las notas de cada curso.

### **2.6.2 Secciones**

#### **2.6.2.1 Encabezado**

Se especifica la información del curso y el código del curso. Por ejemplo:

*COURSE: Introduction to software engineering*

*COURSE CODE: ISE001*

#### **2.6.2.2 Tablas**

Cada fila contiene la información y el consolidado de las notas de un estudiante. Además de

- Un serial para cada estudiante.
- El identificador correspondiente al estudiante.
- El nombre del estudiante.

- Se tienen las notas para las tareas prácticas (*Practical Assignments*). Se tienen dos columnas por cada tarea práctica. La primera columna es reservada para la nota que el estudiante obtenga en su primera oportunidad de realizar la prueba. En caso el estudiante repruebe la prueba puede solicitar retomar la prueba y la nota que obtenga en la segunda oportunidad deberá colocarse en la segunda columna (la inmediata a la derecha). Por ejemplo, se tiene la columna **PA1 y Re PA1**, la columna **PA1** será para la nota obtenida la primera vez que se tome la prueba y la columna **Re PA1** será para la nota obtenida en la segunda oportunidad que tome la prueba si lo hace.
- Se tienen las notas para las actividades de evaluación (*Test per Unit - TU*). Se tienen dos columnas por cada examen. La primera columna es reservada para la nota que el estudiante obtenga en su primera oportunidad de realizar la prueba. En caso el estudiante repruebe la prueba puede solicitar retomar la prueba y la nota que obtenga en la segunda oportunidad deberá colocarse en la segunda columna (la inmediata a la derecha). Por ejemplo, se tiene la columna **TU1 (Test Unit 1) y Re TU1**, la columna **TU1** será para la nota obtenida la primera vez que se tome la prueba y la columna **Re PA1** será para la nota obtenida en la segunda oportunidad que tome la prueba si lo hace.
- Se tiene una columna para la nota del *Seminar* (Exposición) que se realice
- Se tiene una columna para el proyecto si se realizará.
- Se tiene una columna para la nota final obtenida en base a todas las actividades anteriores

### 2.6.3 Formas de utilización

- El *Trainer* debe con base a las formas CEF1050A copiar las notas obtenidas por los estudiantes en cada actividad de evaluación y ubicarla en la columna que corresponda.
- Posteriormente en la columna de **Remarks** el *trainer* debe hacer el cálculo de la nota final asignándole a cada actividad su ponderación correspondiente.
- El *Trainer* manda las notas a los interesados (estudiantes y Centre Manager) en un formato no modificable (PDF).
  - El *Trainer* envía las notas al *Centre Manager* en formato editable (XLS).