



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

**HOMOLOGACIÓN DEL CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN Y  
COMPUTACIÓN 1 DE LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS,  
FACULTAD DE INGENIERÍA, USAC**

**Jhonatan Wilfredo Pú Morales**

Asesorado por el Ing. William Estuardo Escobar Argueta

Guatemala, marzo de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**HOMOLOGACIÓN DEL CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN Y  
COMPUTACIÓN 1 DE LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS,  
FACULTAD DE INGENIERÍA, USAC**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**JHONATAN WILFREDO PÚ MORALES**

ASESORADO POR EL ING WILLIAM ESTUARDO ESCOBAR ARGUETA

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, MARZO DE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Jurgén Andoni Ramírez Ramírez
VOCAL V	Br. Oscar Humberto Galicia Nuñez
SECRETARIA	Ing. Lesbia Magalí Herrera López

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. William Estuardo Escobar Argueta
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. Herman Igor Véliz Linares
SECRETARIA	Inga. Lesbia Magalí Herrera López.

## **HONORABLE TRIBUNAL EXAMINADOR**

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

**HOMOLOGACIÓN DEL CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN Y  
COMPUTACIÓN 1 DE LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS,  
FACULTAD DE INGENIERÍA, USAC**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 4 de marzo de 2016.

**Jhonatan Wilfredo Pú Morales**



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

Guatemala 20 de Julio de 2016

Ingeniero  
Marlon Pérez Türk  
Director  
Escuela de Ciencias y Sistemas  
Facultad de Ingeniería

Respetable Ingeniero Pérez Türk

Por este medio hago constar que el trabajo de graduación del estudiante Jhonatan Wilfredo Pú Morales de carné 200517730, que lleva por título **"HOMOLOGACIÓN DEL CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1 DE LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, USAC"**, el cual cumple con los objetivos trazados para su elaboración según el protocolo presentado.

Sin otro particular

Atentamente,

WILLIAM ESTUARDO ESCOBAR ARGUETA  
INGENIERO EN CIENCIAS Y SISTEMAS  
COLEGIADO 11.529

William Estuardo Escobar Argueta  
Ingeniero en Ciencias y Sistemas



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

Guatemala, 3 de Agosto de 2016

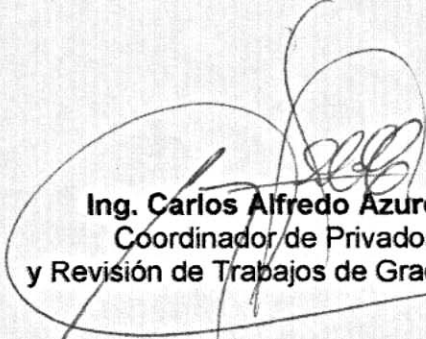
Ingeniero  
**Marlon Antonio Pérez Türk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **JHONATAN WILFREDO PÚ MORALES** con carné **200517730**, titulado: **"HOMOLOGACIÓN DEL CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1 DE LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, 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  
  
I  
N  
G  
E  
N  
I  
E  
R  
Í  
A  
  
E  
N  
  
C  
I  
E  
N  
C  
I  
A  
S  
  
Y  
  
S  
I  
S  
T  
E  
M  
A  
S

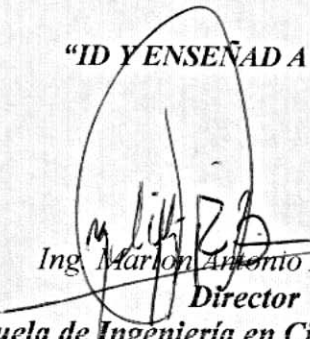
UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN  
CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“HOMOLOGACIÓN DEL CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1 DE LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, USAC”**, realizado por el estudiante **JHONATAN WILFREDO PÚ MORALES** aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**

  
Ing. Marlon Antonio Pérez Türk  
Director

**Escuela de Ingeniería en Ciencias y Sistemas**



Guatemala, 24 de marzo de 2017

Universidad de San Carlos  
de Guatemala

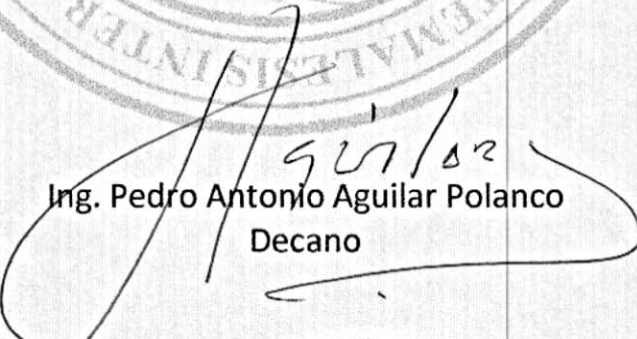


Facultad de Ingeniería  
Decanato

DTG. 152.2017

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: **HOMOLOGACIÓN DEL CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1 DE LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, USAC**, presentado por el estudiante universitario: **Jonathan Wilfredo Pú Morales**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

  
Ing. Pedro Antonio Aguilar Polanco  
Decano

Guatemala, marzo de 2017

/gdech





## **ACTO QUE DEDICO A:**

- Dios** Por darme la vida, brindarme sabiduría, ayuda y fortaleza; por nunca abandonarme y por todas las bendiciones que ha derramado sobre mí.
- Mi padres** Engracia Concepción Morales y Juan Wilfredo Pú Tzul, por el apoyo incondicional, el cuidado y los consejos brindados durante toda mi vida.
- Mis hermanos** Sharonn Pryscila Pú Morales y Junior Rogelio Díaz Morales, por las risas, el cariño y los ánimos dados en los momentos oportunos, en especial a Sharonn por su apoyo incondicional.
- Mis tíos** Fernando Figueroa Marín y Randy Lee Ruden, por el apoyo brindado en cada uno de los momentos difíciles y por el cariño que me han demostrado.
- Mis tías** Judith Maricela y Nidia Arminda Morales Morales, por inculcarme a ser agradecido siempre, inclusive en los momentos adversos; por formarme en todos los

sentidos para ser una mejor persona cada día y ser como mis madres.

**Mis primos**

Luis Fernando y Natalie Diana Figueroa Morales, Edward Mitchell y Ashley Nicole Ruden Morales, con quienes he compartido extraordinarios momentos y enseñarme el valor de una familia unida.

**Mi abuelo**

Rogelio Morales Pinto (q. e. p. d.), por enseñarme el valor de la vida y que lograría este triunfo.

## **AGRADECIMIENTOS A:**

<b>Dios</b>	Por estar siempre a mi lado en las dificultades, renovando mis fuerzas, abriendo puertas y derramando bendiciones para alcanzar esta meta.
<b>Mis padres</b>	Por su indiscutible amor, esfuerzo, y confianza mostrada a lo largo de mi vida.
<b>Mis hermanos</b>	Por apoyarme a no abandonar mis sueños, por las maravillosas aventuras que hemos compartido.
<b>Mis tíos y tías</b>	Por estar conmigo, apoyarme y quererme como un hijo.
<b>Mis amigos</b>	Por sus consejos, palabras de apoyo y sobre todo por los recuerdos inolvidables.
<b>Mis compañeros</b>	Por compartir experiencias inolvidables y ser como hermanos y hermanas en momentos difíciles.
<b>Universidad de San Carlos de Guatemala</b>	Por ser la casa de estudios que me recibió con los brazos abiertos para formarme como profesional.

**Facultad de Ingeniería**

Por recibirme en sus aulas y darme la oportunidad de adquirir nuevos conocimientos y ser parte de mi formación.

**Mi asesor**

Quien me brindó apoyo para mi preparación profesional y me instó a seguir adelante.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO .....	VII
RESUMEN.....	IX
OBJETIVOS.....	XI
Hipótesis.....	XII
INTRODUCCIÓN .....	XIII
1.    MARCO TEÓRICO.....	1
1.1.    Antecedentes.....	1
1.2.    Justificación .....	2
1.3.    Análisis Foda de la situación actual.....	2
1.4.    Contenido del curso.....	4
1.5.    Contenido del laboratorio.....	5
2.    TÉCNICA DE ENSEÑANZA .....	7
2.1.    Métodos de enseñanza .....	7
2.2.    Identificación del método de enseñanza utilizado en el curso de IPC1 .....	10
2.3.    Criterio de calificación.....	11
3.    DESCRIPCIÓN DE HOMOLOGACIÓN A UTILIZAR .....	13
3.1.    Propuesta de contenido.....	13
3.1.1.    Contenido de clase .....	13
3.1.2.    Contenido de laboratorio .....	14
3.2.    Cronograma de actividades a realizar .....	14

3.2.1.	Actividades de clase.....	15
3.2.2.	Actividades de laboratorio .....	18
3.3.	Programas.....	20
3.3.1.	Programa de clase .....	20
3.3.2.	Programa de laboratorio.....	21
3.4.	Material de apoyo actual .....	21
3.4.1.	Material de apoyo sugerido .....	22
3.4.2.	Material de apoyo para cursos posteriores.....	22
3.5.	Ponderación .....	22
3.5.1.	Ponderación de clase .....	23
3.5.2.	Ponderación de laboratorio .....	23
3.6.	Clases magistrales .....	24
3.6.1.	Clases teóricas .....	24
3.6.2.	Clases prácticas .....	25
3.7.	Módulo operativo y gestión de reportes .....	25
3.7.1.	Independencia de actividades .....	25
3.7.2.	Horario.....	26
3.7.3.	Control de notas .....	26
3.7.4.	Evaluaciones .....	26
3.8.	Recomendaciones.....	28
4.	IMPLEMENTACIÓN DE LA HOMOLOGACIÓN EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS .....	29
4.1.	Cadena de mando.....	30
4.1.1.	Autorización del claustro de ingenieros.....	32
4.1.2.	Autorización del coordinador a cargo de los tutores académicos .....	32
4.1.3.	Autorización del director de la Escuela de Ingeniería en Ciencias y Sistemas .....	33

4.2.	Presentación de propuesta de homologación a la cadena de mando.....	34
4.2.1.	Correcciones y cambios de la propuesta de homologación .....	34
4.3.	Ejecución de la propuesta de homologación .....	35
4.3.1.	Nombramiento de coordinadores.....	35
4.3.2.	Presentación de la propuesta de homologación .....	36
4.3.3.	Entrega de la propuesta de homologación .....	36
4.4.	Implementación de la homologación en otros cursos de la Escuela de Ingeniería en Ciencias y Sistemas.....	36
5.	RESULTADOS DE LA APLICACIÓN DE LA HOMOLOGACIÓN CON LOS BENEFICIOS QUE CONLLEVA.....	39
	CONCLUSIONES .....	51
	RECOMENDACIONES .....	53
	BIBLIOGRAFÍA.....	55
	APÉNDICES .....	57
	ANEXOS.....	65





## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Diagrama de la homologación.....	30
2.	Organización jerárquica de la homologación .....	31
3.	Promedio de datos de la tabla XIII, gráfica de línea .....	43
4.	Promedio de datos de la tabla XIII, gráfica de barras.....	43
5.	Promedio de datos de la tabla XIV, gráfica de línea .....	45
6.	Promedio de datos de la tabla XIV, gráfica de barras .....	45
7.	Promedio de datos de la tabla XV, gráfica de línea .....	47
8.	Promedio de datos de la tabla XV, gráfica de barras .....	47
9.	Promedio de datos de la tabla XVI, gráfica de línea .....	48
10.	Promedio de datos de la tabla XVI, gráfica de barras .....	49
11.	Promedio de datos de la tabla XVII, gráfica de línea.....	50
12.	Promedio de datos de la tabla XVII, gráfica de barras .....	50

### TABLAS

I.	Datos de notas de semestres anteriores.....	2
II.	Cuadro de análisis Foda .....	3
III.	Los distintos métodos y sus potencialidades pedagógicas .....	10
IV.	Cronograma de actividades de clase .....	15
V.	Cronograma de clases (clase) .....	17
VI.	Cronograma de actividades de laboratorio.....	18
VII.	Cronograma de clases (laboratorio) .....	19
VIII.	Desglose de ponderación de clase .....	23

IX.	Desglose de ponderación del laboratorio.....	23
X.	Datos del plan piloto de la implementación de la homologación.....	39
XI.	Notas de laboratorio segundo semestre de 2015 .....	40
XII.	Notas de clase segundo semestre de 2015.....	41
XIII.	Datos del primer semestre de 2015 .....	42
XIV.	Datos del segundo semestre de 2015 .....	44
XV.	Datos del primer semestre de 2016 .....	46
XVI.	Datos obtenidos en la encuesta a los tutores académicos .....	48
XVII.	Datos obtenidos en la encuesta a los catedráticos .....	49

## GLOSARIO

<b>Algoritmo</b>	Secuencia de pasos lógicos y ordenados, con las cuales se le da solución a un problema determinado.
<b>Codificación</b>	Trasladar un algoritmo a un lenguaje de programación.
<b>Diagramas de flujo</b>	Es una técnica que se ha utilizado desde los inicios de la programación de computadoras y es particularmente útil cuando se hace una introducción a la ciencia y el arte de programar.
<b>DSI</b>	Departamento de Soporte Informático, lugar donde se resuelve dudas al estudiantado de la Escuela de Ingeniería en Ciencias y Sistemas.
<b><i>Frameworks</i></b>	Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que pueden servir de base para la organización y desarrollo de software.
<b>Homologación</b>	Igualación o comparación de una cosa con otra, por tener ambas en común características referidas a su naturaleza, función o clase.

<b>HTML</b>	Por sus siglas en inglés " <i>Hyper Text Markup Language</i> ". Lenguaje de etiquetas diseñado para ser interpretado por un navegador y generar una página web.
<b>Java</b>	Lenguaje de programación orientado a objetos.
<b>Pseudocódigo</b>	Combinación entre la flexibilidad de un idioma con su contexto y las reglas sintácticas de un lenguaje de programación
<b>Sentencias de control</b>	Determinan el comportamiento de un programa.

## RESUMEN

Como un requisito fundamental para concluir los estudios de la carrera de Ingeniería en Ciencias y Sistemas, se presenta el tema "Homologación del curso de Introducción a la Programación y Computación 1 de la Escuela de Ingeniería en Ciencias Y Sistemas, Facultad de Ingeniería, USAC", el cual se implementó en forma experimental con fines académicos durante el primer semestre de 2015 en dos secciones, segundo semestre de 2015 en todas las secciones y el primer semestre de 2016 en todas las secciones, a los alumnos del curso.

Se utilizó para medir los diferentes aspectos, una encuesta tanto entre los alumnos participantes como el claustro de maestros y los catedráticos, realizando la labor de campo en las clases de las secciones A, B, C, D y E de los estudiantes del segundo semestre de 2015 y el primer semestre de 2016, realizando diferentes ponderaciones para evaluar el grado de avance y aprendizaje de cada uno de ellos, y se compararon los resultados obtenidos con los resultados de semestres anteriores con la finalidad de demostrar que el sistema de homologación ayuda a elevar el conocimiento y las bases para futuros cursos, así como el mejoramiento de la técnica de enseñanza en los diferentes tutores académicos designados para impartir este curso.

Los resultados obtenidos reflejan que hubo una mejora ostensible en los valores de calidad, dificultad, nivel, oportunidad, accesibilidad y conocimiento, obteniendo en todos ellos resultados satisfactorios.



## **OBJETIVOS**

### **General**

Preparar estudiantes que reciban una capacitación académica adecuada utilizando la metodología de homologación del curso de Introducción a la Programación y Computación 1.

### **Específicos**

1. Facilitar el proceso de aprendizaje a los estudiantes del curso de Introducción a la Programación y Computación 1.
2. Capacitar debidamente a los estudiantes para superar los cursos posteriores relacionados con la Escuela de Ingeniería en Ciencias y Sistemas.
3. Preparar a los catedráticos para que conozcan las técnicas de enseñanza, para transmitir sus conocimientos de la mejor forma.
4. Maximizar el uso del tiempo asignado, buscando cubrir los temas programados en el tiempo específico.

## **Hipótesis**

La homologación mejora la calidad en la experiencia del estudiante para garantizar su aprendizaje y concluir el curso con éxito y cumplir con el rol de Introducción a la Programación a la Computación 1, el cual consiste en enseñar las bases de la programación.



## INTRODUCCIÓN

El presente documento tiene como finalidad homologar el material didáctico, exámenes parciales, tareas preparatorias entre otros, utilizados en las clases magistrales (a cargo de los ingenieros) y en los laboratorios (a cargo de los tutores académicos) del curso de Introducción a la Programación y Computación 1 en forma eficiente y eficaz, procurando que los tutores académicos reciban la orientación necesaria para transmitir el conocimiento adquirido en la carrera.

Se busca prepararlos para ser profesionales debidamente capacitados, utilizando el tiempo asignado de laboratorio y horarios de atención al estudiantado (en el departamento de soporte informático) de forma óptima, promoviendo el curso como un proceso de preparación para los cursos posteriores y capacitar al estudiantado en forma general a través de las clases magistrales y clase regular de laboratorio y en forma personalizada (asesoría y orientación).



# 1. MARCO TEÓRICO

En este capítulo se analiza el estatus actual e histórico del curso de Introducción a la Programación y Computación 1, con la finalidad de buscar soluciones o mejoramiento del rendimiento y aprendizaje del curso.

## 1.1. Antecedentes

Se ha observado que actualmente los estudiantes de las distintas secciones tienen carga académica diferente, lo cual repercute en su preparación para superar los cursos posteriores.

Los auxiliares, al iniciar su práctica final, no tienen una guía que les permita orientarse en la realización de las diferentes actividades inherentes al curso.

Actualmente las horas de atención del estudiante en el Departamento de Soporte Informático (DSI, salón 104) no son utilizadas en su totalidad, debido a que los estudiantes eventualmente acuden a recibir asesoría para la aclaración de sus dudas en la realización de sus tareas, prácticas y proyectos.

Una recopilación de notas de semestres anteriores a la implementación de la homologación presenta los siguientes resultados:

Tabla I. **Datos de notas de semestres anteriores**

Semestre y año	<b>Segundo/2007</b>	<b>Primero/2012</b>	<b>Primero/2014</b>
Total de alumnos	44	54	71
Aprobados	14	7	15
Reprobados	30	47	56
Porcentaje de aprobados	31,82	12,96	21,63
Porcentaje de reprobados	68,18	87,04	78,87

Fuente: elaboración propia.

Al comparar los datos de la tabla I se visualiza que el promedio de estudiantes aprobados no supera el 22 %.

## **1.2. Justificación**

Durante mucho tiempo los estudiantes del curso de Introducción a la Programación y Computación 1 de la Escuela de Ingeniería en Ciencias y Sistemas han tenido que enfrentar las limitantes que causa la metodología de enseñanza que es diferente en cada sección, razón por la cual se propone homologar el curso, es decir que todas las secciones tengan una metodología uniforme; la metodología que se propone será de gran utilidad para todos los estudiantes, sin importar la sección en la que reciban el curso, lo que ayudará a que el estudiantado adquiera el mismo nivel de formación profesional con bases sólidas para superar los cursos posteriores.

## **1.3. Análisis Foda de la situación actual**

Las fortalezas, oportunidades, debilidades y amenazas de la situación actual del curso de Introducción a la Programación y Computación 1 se desglosan en la tabla del cuadro de análisis Foda.

Tabla II. Cuadro de análisis Foda

Aspectos	Positivos	Negativos
Internos	Fortalezas	Debilidades
	<ul style="list-style-type: none"> <li>• Proyecto retador que incluía muchos conceptos básicos de programación.</li> <li>• Los auxiliares están dispuestos a ayudar si se necesita.</li> <li>• Aprender java utilizando el libro del curso.</li> <li>• Desarrollar la lógica mental de cada estudiante.</li> <li>• La diversidad de proyectos evita que se entregue el mismo proyecto en una sección diferente y un tutor no lo note.</li> <li>• Las entregas o cambios dentro del proyecto no son dependientes de las actividades de otras secciones.</li> <li>• Se incluyen temas que otras secciones pueden llegar a excluir o ignorar.</li> </ul>	<ul style="list-style-type: none"> <li>• El ingeniero no se involucraba en el laboratorio.</li> <li>• No se da una guía de como se empieza a programar desde cero.</li> <li>• No se enfoca tanto en el análisis y diseño de código.</li> <li>• No se enfoca en buenas prácticas de codificación.</li> <li>• Solamente algunos tutores se interesan en que los alumnos aprendan.</li> <li>• Los proyectos para cada sección son diferentes en complejidad.</li> <li>• Poco análisis del problema a solucionar</li> <li>• Los criterios de evaluación pueden variar en cada sección.</li> <li>• La dificultad y el tamaño de los proyectos varía en cada sección.</li> <li>• Los proyectos no se encuentran definidos por un mismo estándar.</li> <li>• Se presta más a la subjetividad.</li> </ul>
Externos	Oportunidades	Amenazas
	<ul style="list-style-type: none"> <li>• Perderle miedo al desarrollo, ser autodidacta para el aprendizaje de lenguajes y <i>frameworks</i>.</li> <li>• Ganar experiencia bajo el paradigma de programación.</li> <li>• Incita al estudiante a ser autodidacta.</li> </ul>	<ul style="list-style-type: none"> <li>• Acostumbrarse a desarrollar sin analizar antes.</li> <li>• Desarrollar de manera desordenada.</li> <li>• Tener que repetir el curso varias veces para saber programar en java.</li> <li>• Muchos de los estudiantes se retiran del curso e incluso de la carrera.</li> </ul>

Continuación de la tabla II.

	<ul style="list-style-type: none"><li>• Crear conciencia en los estudiantes en querer aprender, es necesario fuerza de voluntad.</li><li>• Cada sección es independiente en cuanto a los criterios de evaluación de contenido del proyecto.</li><li>• Se da la libertad al ingeniero y al auxiliar de elaborar un proyecto según ellos lo crean conveniente.</li></ul>	<ul style="list-style-type: none"><li>• No se cubre el mismo contenido en las distintas secciones.</li><li>• Los estudiantes pueden ser más propensos a perder, dependiendo de su sección y la dificultad del proyecto que esté asignado en esa sección.</li><li>• Los estudiantes no aprenden algunos temas correctamente por aplicarlos muy pobremente si la orientación en el proyecto no está alineada al contenido.</li></ul>
--	--	--

Fuente: elaboración propia.

#### 1.4. Contenido del curso

El contenido del curso está integrado por ocho unidades que son:

- Introducción
- Programación modular y estructuras básicas
- Metodología orientada a objetos
- Programación orientada a objetos
- Estructuras algorítmicas
- Colecciones de datos
- Flujos de bytes y manipulación de archivos
- Los tipos de datos abstractos

El detalle de cada unidad mencionada anteriormente se encuentra en el anexo 1 al 5.

La información presentada en los anexos 1 al 5 tienen como finalidad realizar la propuesta del tema "Homologación del curso de Introducción a la Programación y Computación 1 de la Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería, USAC".

### **1.5. Contenido del laboratorio**

El contenido del laboratorio está integrado por once unidades que son:

- Fundamentos de programación
- Paradigma de objetos y UML
- Programación estructurada
- Introducción a Java
- Programación orientada a objetos en Java (POO)
- Clases abstractas, interfaces y paquetes
- Manejo de excepciones
- Manejos de hilos en Java
- Introducción a AWT y SWING
- Archivos
- Listas y colas

El detalle de cada unidad se encuentra en los anexos del 6 al 7. La información presentada en los anexos 6 y 7 tienen como finalidad realizar la propuesta de el tema "Homologación del curso de Introducción a la Programación y Computación 1 de la Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería, USAC"





## 2. TÉCNICA DE ENSEÑANZA

Existen muchos tipos de métodos y técnicas de enseñanza a, nivel primario, básico, diversificado y universitario, además de identificar cuál es la que se acopla a la implementación de la homologación del curso de Introducción a la Programación y Computación 1, algunos métodos y técnicas se listan a continuación.

### 2.1. Métodos de enseñanza

Un método de enseñanza puede ser diseñado por cualquier persona, según *Grappin* la definición de un método pedagógico es "un conjunto de reglas y principios normativos sobre los cuales descansa la enseñanza"<sup>1</sup>.

En esta sección se describen métodos de enseñanza que ya están establecidos, mencionados en el libro *Aprender a Enseñar*, escrito por *Oriol Amat*.

"La clasificación de los métodos de enseñanza propuestos por *Neciri*"<sup>2</sup>:

- Según la forma de razonamiento: en esta clasificación se encuentran los métodos deductivos, inductivos, analógicos, y algunos otros que no se describen.

---

<sup>1</sup> Grappin, J.P.: *Claves para la formación en la empresa*; Ceac, Barcelona, 1990, pág. 71.

<sup>2</sup> Neciri, I.: *Hacia una didáctica general dinámica*; Kapelusz, Buenos Aires, 1979, pág. 233-280.

- Los métodos deductivos son los comúnmente utilizados en las escuelas primarias donde el catedrático brinda su clase magistral, guiando a los estudiantes según una serie de pasos anteriormente planificada por el catedrático.
- Los métodos inductivos se empiezan a utilizar en el nivel básico, donde se inculca a los estudiantes involucrarse en las distintas actividades realizadas en clase.
- Los métodos analógicos están orientados a un nivel superior universitario, ya que el estudiante debe analizar información para identificar tendencias y presentar resultados.
- Según la actividad de los alumnos: en la clasificación de la actividad de los alumnos los métodos a mencionar son los pasivos y activos.
  - Los métodos pasivos son similares a los métodos deductivos donde el catedrático es el responsable de toda la actividad de enseñanza, el alumno es totalmente dependiente de lo que el catedrático indique.
  - Los métodos activos son lo contrario a los pasivos y similares a los analógicos, el alumno aprende de forma autodidacta y el catedrático facilita el proceso de aprendizaje mediante la resolución de dudas.
- En cuanto al trabajo del alumno: la clasificación en cuanto al trabajo del alumno en términos generales se puede listar el trabajo individual, colectivo y mixto.

- En el trabajo individual cada estudiante resuelve las actividades de la clase por su propia cuenta.
- En el trabajo colectivo los estudiantes forman grupos y resuelven las actividades en conjunto.
- El método de trabajo mixto los alumnos trabajan de forma individual y grupal.

Algunos de los métodos pedagógicos más relevantes según la obra "Aprender a enseñar"<sup>3</sup>, que se adaptan al curso de Introducción a la Programación y Computación 1 son:

- Lección magistral: es un método pasivo y utilizado por los catedráticos en cualquier nivel académico de preparación para transmitir un mensaje siendo los estudiantes los receptores del mensaje.
- Tutoría: el método de tutoría no es muy utilizado ya que requiere de un esfuerzo extra por parte de los catedráticos, el motivo del esfuerzo es porque se enseña a los alumnos en grupos muy pequeños con la finalidad de transmitir la mayor cantidad de conocimiento. Los resultados del método de tutoría son muy buenos debido a que los estudiantes son capacitados en los distintos temas de la clase en forma personal.
- Ejercicio: el método de ejercicio consiste en la práctica de la teoría que se brinda en la clase magistral por el catedrático. El alumno resuelve problemas prácticos a partir de la teoría.

---

<sup>3</sup> Amat, Oriol: *Aprender a enseñar: una visión práctica de la formación de formadores*. Barcelona: Gestión 2000, pág. 83-104.

Cada método anteriormente descrito tiene ventajas y desventajas para visualizar de una forma clara los beneficios de cada uno de ellos se presenta la tabla que describe las competencias que potencia cada metodología:

Tabla III. **Los distintos métodos y sus potencialidades pedagógicas**

Método	Competencias que potencian						
	Saber	Saber hacer	Trabajo en equipo	Comunicación	Liderazgo	Toma de decisión	Creatividad
Lección magistral	X						
Tutoría		X	X	X	X		X
Ejercicio	X	X				X	

Fuente: AMAT, Oriol. *Aprender a enseñar*. p. 56.

## 2.2. Identificación del método de enseñanza utilizado en el curso de IPC1

Basado en la descripción de las distintas metodologías definidas en la sección 2.1, donde se analiza el método utilizado en el curso de Introducción a la Programación y Computación 1 es:

- Para la clase el método utilizado es una combinación de la lección magistral y el método del ejercicio.
- Para el laboratorio el método utilizado es una combinación de la lección magistral, el método de la tutoría y el ejercicio.

### **2.3. Criterio de calificación**

Para realizar las evaluaciones de las actividades relacionadas con curso de Introducción a la Programación y Computación 1 y para validar que los estudiantes cumplan con los objetivos del curso se sugieren los siguientes criterios de calificación:

- Evaluar la comprensión de los temas planteados en las actividades.
- Evaluar la capacidad de análisis del estudiante sobre la actividad evaluada.
- Evaluar que el estudiante adquirió la habilidad de aprender de manera autodidáctica nuevas técnicas.
- Evaluar la proposición de respuestas claras con justificación en las preguntas planteadas de la actividad.
- Evaluar la explicación de respuestas con argumentos en orden lógico a las preguntas planteadas de la actividad.
- Evaluar el dominio y conocimiento adecuado en el lenguaje de programación utilizado para la actividad.



### **3. DESCRIPCIÓN DE HOMOLOGACIÓN A UTILIZAR**

La descripción de la homologación está conformada por una propuesta de contenido, cronograma de actividades, programas, material de apoyo, ponderación, clases magistrales, módulo operativo y gestión de reportes, además de recomendaciones.

#### **3.1. Propuesta de contenido**

La fuente inicial del contenido se obtiene de los programas del curso y laboratorio del semestre anterior; si el catedrático y el tutor académico desean realizar cambios sobre los programas se sigue la secuencia de pasos descritos en el sub inciso 3.1.1 y 3.1.2 del presente capítulo.

##### **3.1.1. Contenido de clase**

El contenido de la clase está indicado en el programa del curso, si se considera necesario realizar algunas modificaciones al mismo, el proceso es el siguiente:

- Realizar la propuesta de modificación al contenido de clase.
- Presentar la propuesta de modificación a los catedráticos.
- Llegar a común acuerdo para realizar la modificación al contenido de la clase.

- Realizar la modificación al contenido de la clase, si es aprobado por todos los catedráticos, si no hay acuerdo el contenido se conserva.

### **3.1.2. Contenido de laboratorio**

El contenido del laboratorio está indicado en el programa del laboratorio, si se considera necesario realizar algunas modificaciones al mismo, el proceso es el siguiente:

- Realizar la propuesta de modificación al contenido del laboratorio.
- Presentar la propuesta a los tutores académicos.
- Llegar a común acuerdo para realizar la modificación al contenido del laboratorio.
- Presentar la propuesta de modificación a los catedráticos del curso.
- Si hay visto bueno de parte de los catedráticos para la modificación, se procede a realizarla, de lo contrario se conserva el contenido.

### **3.2. Cronograma de actividades a realizar**

El cronograma de actividades a realizar está conformado por las actividades de clase y las actividades de laboratorio; cada uno tiene designado un cronograma tanto de laboratorio como de las clases a impartir.



### 3.2.1. Actividades de clase

El cronograma de actividades a realizar que se sugiere utilizar para evaluar el desempeño de las clases impartidas consta de cuatro puntos que son:

- **Actividades:** describe de forma general el nombre de la actividad a realizar.
- **Descripción:** detalle que describe la actividad.
- **Inicio:** indica la fecha de inicio de la actividad.
- **Entrega:** indica la fecha de finalización y entrega de la actividad.

El detalle de las actividades sugeridas a realizar se encuentra en la tabla IV.

Tabla IV. **Cronograma de actividades de clase**

<b>Actividades</b>	<b>Descripción</b>	<b>Inicio</b>	<b>Entrega</b>
Registro de datos	Actividad para centralizar a los estudiantes del curso de Introducción a la Programación y Computación 1, y realizar la distribución de secciones.	19/01/16	21/01/16
Tarea 1	Investigación del ciclo de vida del software y toma de requerimientos.	26/01/16	02/02/16
Hoja de trabajo 1	Resolución tres problemas representándolos con diagramas de flujo y pseudocódigo.	09/02/16	
Corto 1	Resolución de dos problemas utilizando programación orientada a objetos en forma algorítmica (pseudocódigo).	16/02/16	
Tarea 2	Resolución de un problema de arreglos y un problema de ordenamiento de arreglos.	08/04/16	10/04/16

Continuación de la tabla IV.

Hoja de trabajo 2	Resolución de un problema utilizando diagrama de clases y recursividad.	29/03/16	
Corto 2	Resolución de cuatro problemas, el primero realizar un diagrama de clases sobre un problema, el segundo responder a 5 preguntas teóricas, el tercero, a la resolución de un problema utilizando pseudocódigo y el cuarto, resolución de un problema con recursividad, utilizando pseudocódigo.	05/04/16	
Tarea 3	Investigación del proceso de escribir objetos en un archivo.	14/04/16	19/04/16
Hoja de trabajo 3	Implementación en pseudocódigo de una pila y una cola.	21/04/16	
Corto 3	Implementación de una cola utilizando una pila en lenguaje de programación Java.	28/04/16	

Fuente: elaboración propia.

El cronograma de clases a impartir que se sugiere utilizar para lograr los objetivos de la clase consta de cuatro puntos que son:

- Días de clase: indica la cantidad de días por unidad.
- Contenido a impartir: indica la unidad que hay que cubrir y en que fechas.
- Actividades: las actividades a realizar son las descritas en el cronograma de actividades de la clase.
- Observaciones: se anota cualquier observación respecto de la clase.

A continuación se presenta la propuesta del cronograma completo con todos sus componentes.

Tabla V. **Cronograma de clases (clase)**

<b>Días de clase</b>	<b>Contenido a impartir</b>	<b>Actividades</b>	<b>Observaciones</b>	
2	Unidad 1: introducción 19 y 21 de enero	Las actividades a realizar están descritas en el cronograma de actividades de la clase.		
4	Unidad 2: metodología orientada a objetos 26 y 28 de enero, 2 y 4 de febrero			
3	Unidad 3: programación modular y estructuras básicas 9, 11 y 16 de febrero			
3	Unidad 4: programación orientada a objetos 23, 25 de febrero y 1 de marzo			
3	Unidad 5: estructuras algorítmicas 3, 8, 10 y 29 de marzo			
4	Unidad 6: colecciones de datos 31 de marzo, 5, 12 y 14 de abril			
0	Unidad 7: flujos de bytes y manipulación de archivos			Investigación
4	Unidad 8: los tipos de datos abstractos 19, 21, 26 y 28 de abril			
1	Primer examen parcial			18 de febrero
1	Segundo examen parcial			7 de abril
1	Tercer examen parcial			3 de mayo
1	Examen final			17 de mayo
1	Entrega de zonas			10 de mayo

Fuente: elaboración propia.

### 3.2.2. Actividades de laboratorio

El cronograma de actividades a realizar que se sugiere utilizar para el laboratorio sigue la misma estructura que el cronograma de actividades de clase y es:

Tabla VI. **Cronograma de actividades de laboratorio**

Actividades	Descripción	Se deja	Se entrega
Actividad 0	Registro de datos	19/01/16	21/01/16
Tarea 1	Elaboración de una página HTML.	21/01/16	28/01/16
Tarea 2	Diseño de algoritmos haciendo uso de diagramas de flujo.	28/01/16	04/02/16
Tarea 3	Codificación en Java de la tarea 2.	04/02/16	11/02/16
Tarea práctica 1	Completar las plantillas FP (formato de proyecto) que se proveerán con información relevante para la construcción y avance del proyecto 1, ver anexos 8 al 11 y apéndice 1.	11/02/16	18/02/16
Tarea práctica 2	Completar las plantillas FP (formato de proyecto) que se proveerán con información relevante para la construcción y avance del proyecto 2, ver anexo 8 al 11 y apéndice 1.	10/03/16	17/03/16
Tarea práctica 3	Programa utilizando librerías e ingeniería inversa (presencial).	21/04/16	
Práctica 1	Programa en consola utilizando ciclos, sentencias de control y vectores.	11/02/16	25/02/16
Práctica 2	Programa con interfaz utilizando hilos.	28/03/16	11/04/16
Práctica 3	Programa con interfaz utilizando archivos.	14/01/16	28/06/16
Proyecto 1	Desarrollo de un programa similar a una hoja de cálculo con funciones distintas para cada sección.	11/02/16	10/03/16
Proyecto 2	Programa para inteligencia de negocios utilizando el proyecto 1 como fuente de información.	10/03/16	05/05/16
Exámenes cortos	Se realizarán 18 exámenes cortos de 5 preguntas en inglés en el horario que el tutor académico delegue.	Martes y jueves del primer semestre de 2016	
Examen final	Evaluación del contenido de laboratorio con parte teórica y práctica.	06/05/16	

Fuente: elaboración propia.

El cronograma de clases de laboratorio que se sugiere utilizar para lograr los objetivos del laboratorio consta de cuatro puntos importantes que son:

- Número de clase: indica en orden ascendente el número de la clase a impartir.
- Clase de laboratorio: contenido de la clase que se impartirá.
- Actividades: listado de actividades que se realizarán en cada clase.
- Encargado: los tutores académicos son los responsables de impartir las clases de laboratorio.

A continuación se presenta la propuesta del cronograma completo con todos sus componentes.

Tabla VII. **Cronograma de clases (laboratorio)**

Número de clase	Clase de laboratorio	Actividades	Encargado
1	Módulo introducción a la computación, HTML e importancia de diseño.	<ul style="list-style-type: none"> <li>• Al inicio de cada clase se realizará un examen corto.</li> <li>• A partir de la clase 12 se realizarán actividades para avanzar en el proyecto.</li> <li>• Realizar ejemplos prácticas en cada clase.</li> <li>• Se tomará en cuenta la participación de los estudiantes en clase.</li> </ul>	Tutores académicos
2	Unidad 1		
3	Unidad 2		
4	Unidad 3		
5	Unidad 4		
6	Unidad 5		
7	Unidad 6		
8	Unidad 7 y 8		
9	Unidad 9		
10	Unidad 10		
11	Unidad 11		
12 a 16			
Ultimo día de clases según calendario oficial de la Facultad de Ingeniería.	Examen final		
	Entrega de notas de laboratorio		

Fuente: elaboración propia.

### **3.3. Programas**

El programa propuesto para la clase o laboratorio a utilizar puede ser susceptible a cambios en relación con el programa original; dichos cambios siguen la secuencia de pasos descritos a continuación:

- Identificar los cambios pertinentes a realizar.
- Presentar la propuesta de cambio.
- Llegar a común acuerdo con los catedráticos y tutores académicos para realizar el cambio sobre el programa (de clase o laboratorio).
- En caso de no llegar a un acuerdo el programa se mantiene.

#### **3.3.1. Programa de clase**

El programa de clase está conformado por las siguientes partes:

- Información general
- Descripción del curso
- Objetivos
- Metodología
- Evaluación del rendimiento académico
- Contenido
- Cláusulas restrictivas
- Bibliografía
- Horario del curso

El detalle de cada una de las partes descritas se encuentra en los anexos 12 al 21.

### **3.3.2. Programa de laboratorio**

El programa de laboratorio está conformado por las siguientes partes:

- Información general
- Descripción del curso
- Objetivos
- Metodología
- Requisitos
- Evaluación
- Contenido de laboratorio
- Puntos importantes a considerar
- Bibliografía

El detalle de cada una de las partes descritas se encuentra en los anexos 22 al 29.

### **3.4. Material de apoyo actual**

El material está conformado por documentos y ejemplos que facilitan el aprendizaje del estudiantado de Introducción a la Programación y Computación 1, además de brindarles una preparación con material de apoyo de los cursos posteriores que son: Lenguajes Formales y de Programación e Introducción a la Programación y Computación 2.

### **3.4.1. Material de apoyo sugerido**

El material de apoyo sugerido para la clase y el laboratorio es una recopilación de documentos brindados por tutores académicos de semestres anteriores y de los catedráticos del curso de Introducción a la Programación y Computación 1, el cual se encuentra en el link mencionado en el apéndice 2, conformado por:

- Programa de clase
- Programa de laboratorio
- Material didáctico de la clase
- Material didáctico del laboratorio
- Lecturas
- Enunciados de las actividades
- Ejemplos prácticos

### **3.4.2. Material de apoyo para cursos posteriores**

El material de apoyo para cursos posteriores es una recopilación de documentos brindados por tutores académicos de los cursos de Lenguajes Formales y de Programación e Introducción a la Programación y Computación 2, los cuales se encuentra en el *link* de la carpeta de la homologación de Introducción a la Programación y Computación 1, mencionado en el apéndice 2.

### **3.5. Ponderación**

Los elementos que conforman la ponderación del curso y del laboratorio se mencionan en cada uno de los incisos siguientes:



### 3.5.1. Ponderación de clase

La ponderación para cada actividad de la clase que se sugiere utilizar es:

Tabla VIII. **Desglose de ponderación de clase**

<b>Actividad</b>	<b>Puntos</b>
Tarea 1	1
Tarea 2	1
Tarea 3	1
Hoja de trabajo 1	1
Hoja de trabajo 2	1
Hoja de trabajo 3	1
Examen corto 1	1
Examen corto 2	1
Examen corto 3	1
Primer examen parcial	8
Segundo examen parcial	14
Tercer examen parcial	14
Laboratorio	30
Examen final	25
<b>Total</b>	<b>100</b>

Fuente: elaboración propia.

### 3.5.2. Ponderación de laboratorio

La ponderación para cada actividad del laboratorio que se sugiere utilizar es:

Tabla IX. **Desglose de ponderación del laboratorio**

<b>Actividad</b>	<b>Puntos</b>
Tarea 1	2
Tarea 2	2
Tarea 3	2
Tarea práctica 1	3

Continuación de la tabla IX.

Tarea práctica 2	3
Tarea práctica 3	5
Práctica 1	5
Práctica 2	5
Práctica 3	5
Proyecto 1	15
Proyecto 2	25
Exámenes cortos	20
Examen final	8
<b>Total</b>	<b>100</b>

Fuente: elaboración propia.

### **3.6. Clases magistrales**

La clase magistral es impartida por el catedrático y tutores académicos en forma verbal utilizando ayuda audiovisual (pizarrón, proyección, material de apoyo), además del uso de tutorías en el tema de listas enlazadas.

#### **3.6.1. Clases teóricas**

Las clases teóricas se impartirán en el salón asignado para el curso según el horario oficial de la Facultad de Ingeniería, se sugiere utilizar el material didáctico que se encuentra en la carpeta de homologación del curso de Introducción a la Programación y Computación 1, en el *link* mencionado en el apéndice 2.

### **3.6.2. Clases prácticas**

Las clases prácticas se impartirán en el salón asignado para el laboratorio según el horario reservado por el tutor académico, se sugiere utilizar el material didáctico mencionado en el apéndice 2.

### **3.7. Módulo operativo y gestión de reportes**

Esta etapa consiste en diversas actividades que ayudarán a que el contenido del curso pueda ser impartido en forma ágil y efectiva.

#### **3.7.1. Independencia de actividades**

- Tareas: los tutores académicos de cada sección del curso son los encargados de la redacción de cada tarea, debiendo cubrir los puntos mencionados en la sección 3.2.2.
- Tareas prácticas: las tareas prácticas 1 y 2 son la documentación de los proyectos 1 y 2 respectivamente, para la tarea práctica 3, los tutores académicos son los encargados de la redacción, y deben cubrir los puntos mencionados en la sección 3.2.2.
- Prácticas: los tutores académicos de cada sección del curso son los encargados de la redacción de los enunciados y hojas de calificación de cada práctica, cubriendo los puntos antes mencionados en la sección 3.2.2.
- Proyectos (los proyectos son los mismos para todas las secciones del curso): para cada sección del curso se manejarán módulos diferentes

con el mismo nivel de dificultad con la finalidad de evitar copias entre secciones y facilidad de identificación de copias.

- Exámenes cortos: la herramienta de evaluación y los temas a cubrir quedan a discreción de los tutores académicos, es válido utilizar la herramienta sugerida en la sección 3.2.2.

### **3.7.2. Horario**

La clase estará sujeta al horario oficial de cursos establecido por la Facultad de Ingeniería.

El laboratorio va a ser impartido en el horario que defina cada tutor académico, según su disponibilidad de tiempo.

### **3.7.3. Control de notas**

Para el control de notas se sugiere utilizar un sistema al cual tendrán acceso únicamente los tutores académicos y el catedrático de cada sección del curso de Introducción a la Programación y Computación 1.

La plantilla para el control de notas de clase y laboratorio que se sugiere utilizar se presenta en los apéndices 6 al 7, y su vista previa en el apéndice 3.

### **3.7.4. Evaluaciones**

El modelo para las evaluaciones de clase que se sugiere utilizar es:

- Primer examen parcial: el contenido que se sugiere evaluar es la unidad 1: conceptos computacionales.
- Segundo examen parcial: el contenido que se sugiere evaluar corresponde a las unidades 2, 3 y 4.
- Tercer examen parcial: el contenido que se sugiere evaluar debe ser el de las unidades 5, 6, 7 y 8.
- Examen final: el contenido que se sugiere evaluar es la totalidad de las unidades de clase.

El detalle de la propuesta para cada evaluación de clase se presenta en el apéndice 4.

El modelo para las evaluaciones de laboratorio que se sugiere utilizar es:

- Para las evaluaciones cortas se sugiere realizar 5 preguntas con las indicaciones mencionadas en el cronograma de actividades de laboratorio.
- Para el examen final se sugiere evaluar la totalidad de las unidades de laboratorio.

El detalle de la propuesta para las evaluaciones cortas y el examen final de laboratorio se presenta en el apéndice 5.

### **3.8. Recomendaciones**

Para obtener buenos resultados en la implementación de la homologación se sugiere las siguientes recomendaciones.

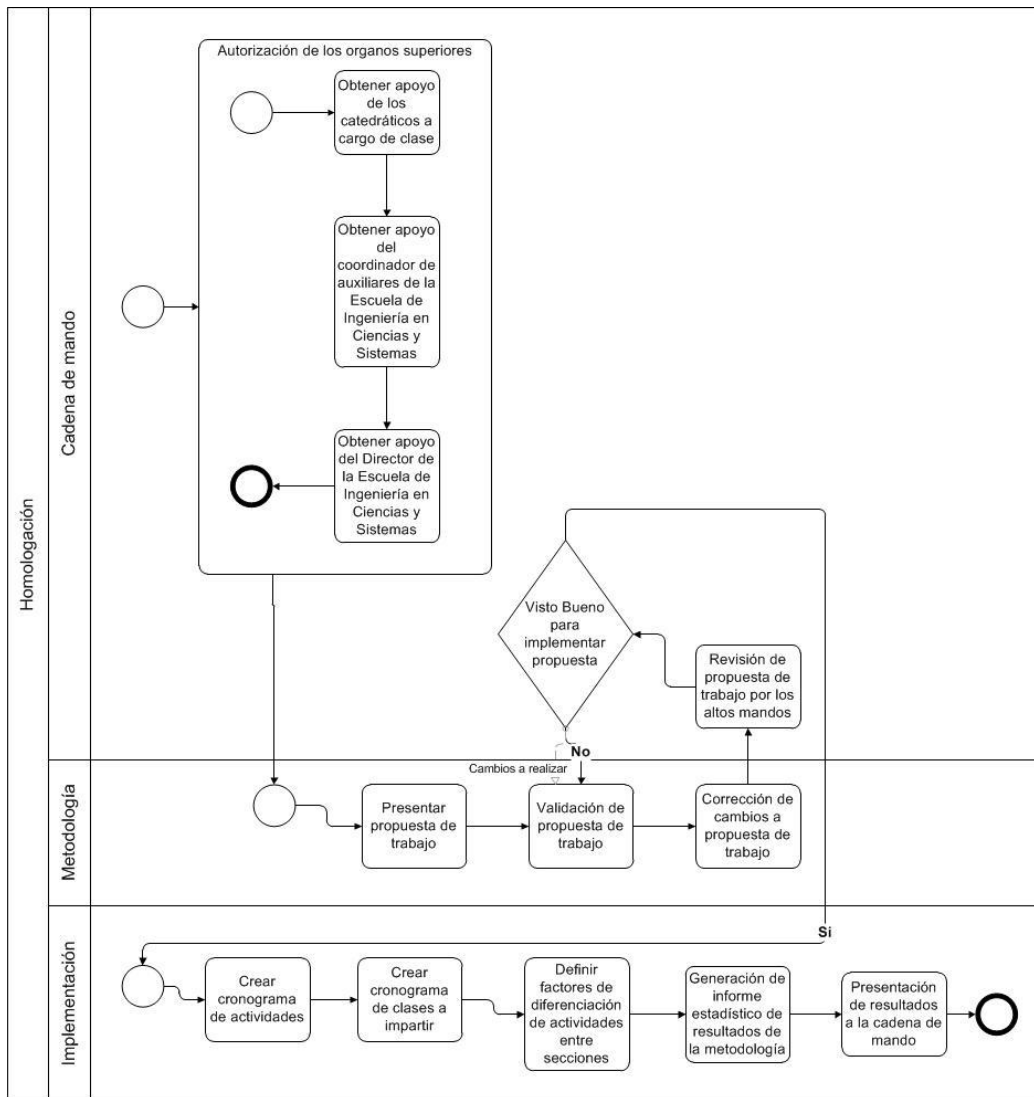
- Utilización de las técnicas de enseñanza descritas en el capítulo 2.
- Seguimiento estricto de las actividades sugeridas a realizar en la clase y laboratorio.

#### **4. IMPLEMENTACIÓN DE LA HOMOLOGACIÓN EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS**

Para lograr la implementación de la homologación en la Escuela de Ingeniería en Ciencias y Sistemas, debe seguirse la secuencia lógica de cada uno de los procesos, tal como se muestra en la figura 1.

Para que la homologación pueda llevarse a cabo en forma efectiva y correcta es necesario designar un coordinador de catedráticos y un coordinador de tutores académicos, quienes tendrán a su cargo la supervisión y orientación en cada etapa del proceso.

Figura 1. Diagrama de la homologación



Fuente: elaboración propia, con programa Visio 2010.

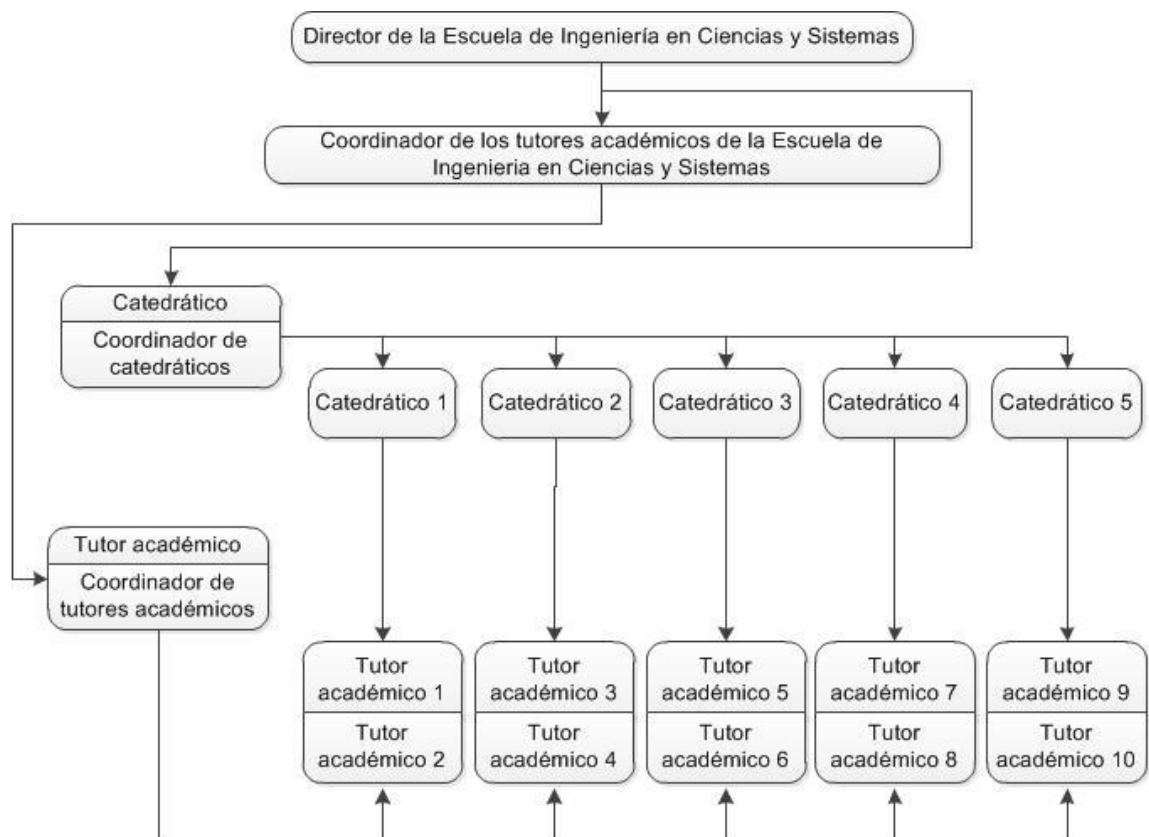
#### 4.1. Cadena de mando

La cadena de mando en una organización es de vital importancia para determinar las funciones que debe realizar cada persona.



Para formar la cadena de mando se requiere una organización jerárquica; en la figura 2 se muestra la estructura jerárquica de la homologación (organización matricial) del curso de Introducción a la Programación y Computación 1.

Figura 2. **Organización jerárquica de la homologación**



Fuente: elaboración propia, con programa Visio 2010.

#### **4.1.1. Autorización del claustro de ingenieros**

El proceso de autorización inicial para la implementación de la homologación debe de ser emitido por el claustro de maestros del curso que conlleva a realizar las siguientes actividades:

- Consultar el horario de clases.
- Extraer el horario en que se imparte el curso de cada catedrático.
- Visitar a cada catedrático para presentar la propuesta de homologación y obtener el visto bueno para realizar la implementación de la homologación.

#### **4.1.2. Autorización del coordinador a cargo de los tutores académicos**

Después de obtener el apoyo y visto bueno de los catedráticos del curso se procede a visitar al coordinador a cargo de los tutores académicos, para realizar los siguientes pasos:

- Visitar la Escuela de Ingeniería en Ciencias y Sistemas.
- Solicitar audiencia a la secretaria de la Escuela de Ingeniería en Ciencias y Sistemas.
- Solicitar el horario del coordinador a cargo de los tutores académicos de la Escuela de Ingeniería en Ciencias y Sistemas.

- Visitar la Escuela de Ingeniería en Ciencias y Sistemas en el horario indicado por la secretaria de la Escuela de Ingeniería en Ciencias y Sistemas.
- Presentar la propuesta de homologación al coordinador a cargo de los tutores académicos de la Escuela de Ingeniería en Ciencias y Sistemas.
- Obtener el visto bueno para poder realizar la implementación de la homologación.

#### **4.1.3. Autorización del director de la Escuela de Ingeniería en Ciencias y Sistemas**

El último paso para obtener el visto bueno de la cadena de mando, es obtener el visto bueno del director de la Escuela de Ingeniería en Ciencias y Sistemas, para lo que se deben seguir los siguientes pasos:

- Visitar la Escuela de Ingeniería en Ciencias y Sistemas.
- Solicitar audiencia a la secretaria de la Escuela de Ingeniería en Ciencias y Sistemas.
- Solicitar el horario del director de la Escuela de Ingeniería en Ciencias y Sistemas.
- Visitar la Escuela de Ingeniería en Ciencias y Sistemas en el horario indicado por la Secretaría de la Escuela de Ingeniería en Ciencias y Sistemas.

- Presentar la propuesta de homologación al director de la Escuela de Ingeniería en Ciencias y Sistemas.
- Obtener el visto bueno para poder realizar la implementación de la homologación.

#### **4.2. Presentación de propuesta de homologación a la cadena de mando**

La presentación de la propuesta homologación a utilizar sigue la estructura, como se indicó en el capítulo 3, pueden presentarse cambios por sugerencia de la cadena de mando; a continuación se presenta la serie de pasos a seguir para realizar los cambios y obtener el visto bueno para la ejecución de la propuesta de homologación.

##### **4.2.1. Correcciones y cambios de la propuesta de homologación**

La propuesta de homologación puede sufrir, modificaciones o correcciones a petición de la cadena de mando.

Para realizar una modificación o corrección es necesario observar los siguientes pasos:

- Recibir la solicitud de modificación o corrección de la propuesta de homologación.
- Realizar toma de requerimientos de la modificación o corrección para elaborar un análisis del impacto, alcance y límite.

- Efectuar la modificación o corrección, utilizando como base el análisis realizado en el paso anterior.
- Presentar nuevamente la propuesta de homologación con la modificación o corrección solicitada por la cadena de mando.
- En el caso que la propuesta de homologación no sufra más modificaciones o correcciones, se procede a realizar el proceso para obtener el visto bueno para la ejecución de la propuesta de homologación.

#### **4.3. Ejecución de la propuesta de homologación**

La ejecución o implementación de la propuesta es muy sencilla; para realizarla se debe nombrar a los coordinadores, presentar la propuesta de homologación a los catedráticos y tutores académicos y por último entregar la propuesta de homologación a los catedráticos y tutores académicos.

##### **4.3.1. Nombramiento de coordinadores**

El coordinador de catedráticos es nombrado democráticamente por los catedráticos del curso, a diferencia del coordinador de los tutores académicos quien es designado por la persona que está a cargo de los tutores académicos de la Escuela de Ingeniería en Ciencias y Sistemas.

#### **4.3.2. Presentación de la propuesta de homologación**

La presentación de la propuesta es de la siguiente forma:

- Convocar a reunión a los catedráticos y tutores académicos el primer día de clases.
- Explicar los detalles y por menores de la metodología de homologación los cuales están en el capítulo 3.
- Resolución de dudas.

#### **4.3.3. Entrega de la propuesta de homologación**

La propuesta de homologación se envía a los catedráticos y tutores académicos por medio de correo electrónico.

#### **4.4. Implementación de la homologación en otros cursos de la Escuela de Ingeniería en Ciencias y Sistemas**

Para la implementación de la homologación en otros cursos de la Escuela de Ingeniería en Ciencias y Sistemas se debe realizar lo siguiente:

- Tomar como punto de partida el capítulo 3 del presente documento.
- Seguir cada inciso y sustituir la información por la información del curso en el que se desea implementar.

- Las tablas para cronograma y actividades pueden ser utilizadas con el mismo formato.
- El diagrama del proceso de la homologación es el mismo para todos los cursos.
- La estructura jerárquica de la homologación para otro curso puede variar en la cantidad de catedráticos y tutores académicos; el resto de la estructura permanece sin cambios.
- El punto de finalización es el capítulo 4 y subinciso 4.3.





## 5. RESULTADOS DE LA APLICACIÓN DE LA HOMOLOGACIÓN CON LOS BENEFICIOS QUE CONLLEVA

El proceso de homologación en el curso de Introducción a la Programación y Computación 1 se aplicó en forma experimental en el primer semestre de 2015, en dos secciones obteniendo los datos que se muestran en la tabla X.

Tabla X. **Datos del plan piloto de la implementación de la homologación**

	Laboratorio		Clase	
	Sección D	Sección C	Sección D	Sección C
Cantidad de alumnos	46	26	46	26
Alumnos aprobados	36	12	32	12
Alumnos reprobados	10	14	14	14
Porcentaje de alumnos aprobados	78,26 %	46,15 %	69,57 %	46,15 %
Porcentaje de alumnos reprobados	21,74 %	53,85 %	30,43 %	53,85 %

Fuente: elaboración propia.

De la tabla X en la sección C, puede observarse que el 100 % de los alumnos que aprobaron el laboratorio, aprobaron la clase, y de la sección D, el 88,89 % de los alumnos que aprobaron el laboratorio también aprobaron la clase.

En el segundo semestre de 2015, el proceso de homologación se lanzó a las cinco secciones del curso de Introducción a la Programación y Computación

1, continuando con la homologación en el primer semestre de 2016, obteniendo los datos que se muestran en la tabla XI y en la tabla XII.

Tabla XI. **Notas de laboratorio segundo semestre de 2015**

	<b>Laboratorio de las cinco secciones de IPC1</b>				
	Sección A	Sección B	Sección C	Sección D	Sección E
Cantidad de alumnos	49	35	40	40	44
Alumnos aprobados	3	5	11	12	17
Alumnos reprobados	46	30	29	28	27
Porcentaje de alumnos aprobados	6,12 %	14,29 %	27,5 %	30 %	38,64 %
Porcentaje de alumnos reprobados	93,88 %	85,72 %	72,5 %	70 %	61,63 %

Fuente: elaboración propia.

De la tabla XI la tendencia es que las secciones con tutores académicos que anteriormente trabajaron con el plan piloto de la homologación; el porcentaje de alumnos es similar (secciones C, D y E) la diferencia radica en la sección E, en la cual se brindaron tutorías académicas los días domingos.

Tabla XII. **Notas de clase segundo semestre de 2015**

	<b>Clase de las cinco secciones de IPC1</b>				
	Sección A	Sección B	Sección C	Sección D	Sección E
Cantidad de alumnos	49	35	40	40	44
Alumnos aprobados	3	5	10	12	17
Alumnos reprobados	46	30	30	28	27
Porcentaje de alumnos aprobados	6,12 %	14,29 %	25 %	30 %	38,64 %
Porcentaje de alumnos reprobados	93,88 %	85,72 %	75 %	70 %	61,63 %

Fuente: elaboración propia.

De la tabla XII la tendencia que se puede abstraer es que el 100 % de los estudiantes que aprueban el laboratorio también aprueban la clase.

Con base en los datos obtenidos de encuestas realizadas a los alumnos (primer semestre de 2015, segundo semestre de 2015, primer semestre de 2016), tutores académicos (primer semestre de 2015) y catedráticos (primer semestre de 2015, segundo semestre de 2015 y primer semestre de 2016), para describir la percepción de la metodología y su tendencia se presentan en las tablas XIII, XIV, XV, XVI y XVII; además en las figuras 3 a la 12, se presenta el promedio de los datos recopilados. La boleta de encuesta que sirvió para recabar los datos se presentan en el apéndice 8.

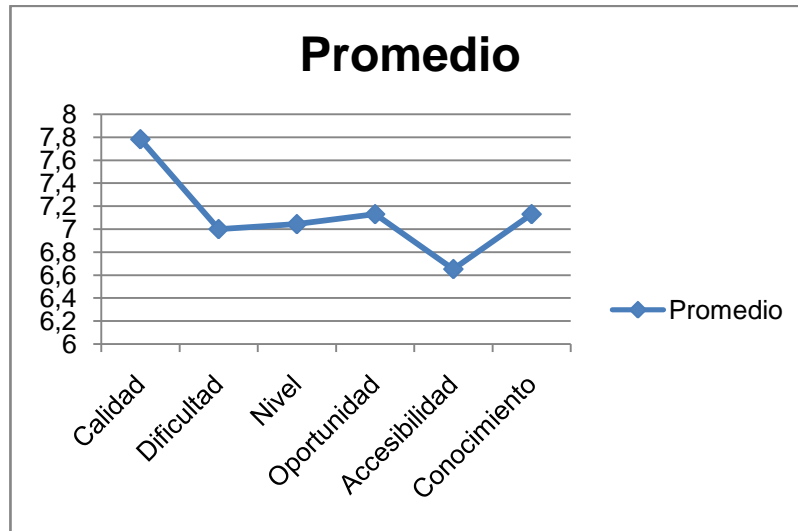
Tabla XIII. Datos del primer semestre de 2015

Estudiante	Calidad	Dificultad	Nivel	Oportunidad	Accesibilidad	Conocimiento
1	9	8	6	7	7	6
2	7	5	6	4	4	8
3	8	8	8	7	8	8
4	8	9	7	7	10	10
5	9	8	8	7	8	8
6	5	9	4	6	4	1
7	9	8	9	8	8	9
8	8	10	10	10	8	10
9	7	7	3	5	1	3
10	6	10	7	6	6	6
11	9	5	9	8	8	9
12	10	7	9	10	10	10
13	9	5	8	8	9	9
14	8	6	8	7	8	8
15	5	7	6	7	8	9
16	8	7	7	8	3	7
17	7	7	6	7	7	6
18	8	5	7	7	6	6
19	7	4	7	8	3	4
20	10	6	7	8	9	10
21	10	8	8	9	9	10
22	4	3	3	5	4	5
23	8	9	9	5	5	2

Fuente: elaboración propia.

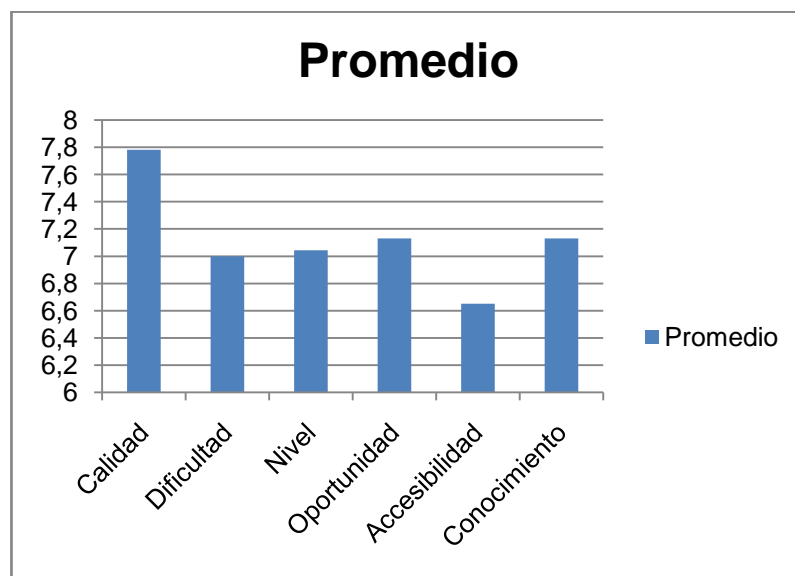
Para una mejor comprensión de los datos de la tabla XIII, el promedio de los datos se muestra de la figura 3 a la 4.

Figura 3. Promedio de datos de la tabla XIII, gráfica de línea



Fuente: elaboración propia, empleando Excel 2007.

Figura 4. Promedio de datos de la tabla XIII, gráfica de barras



Fuente: elaboración propia, empleando Excel 2007.

Las gráficas de las figuras 3 y 4 indican que el plan piloto de homologación resultó tener una dificultad, nivel, oportunidad y conocimiento medio, además de indicar que la calidad de la metodología es de alto nivel con una accesibilidad moderada.

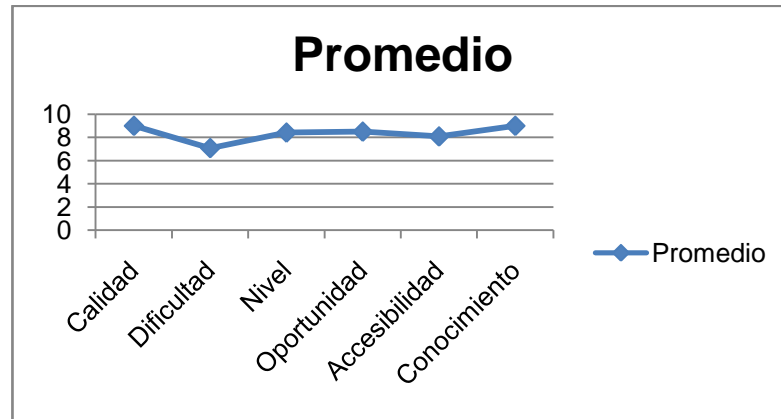
Tabla XIV. **Datos del segundo semestre de 2015**

<b>Estudiante</b>	<b>Calidad</b>	<b>Dificultad</b>	<b>Nivel</b>	<b>Oportunidad</b>	<b>Accesibilidad</b>	<b>Conocimiento</b>
1	9	5	9	9	10	10
2	9	8	8	7	4	8
3	10	2	9	10	9	10
4	9	7	8	8	7	10
5	9	8	9	8	8	9
6	8	9	8	8	9	8
7	9	8	8	8	8	8
8	6	10	7	8	5	7
9	10	8	9	9	10	9
10	9	8	8	8	9	9
11	10	4	10	9	10	10
12	10	8	8	10	8	10

Fuente: elaboración propia.

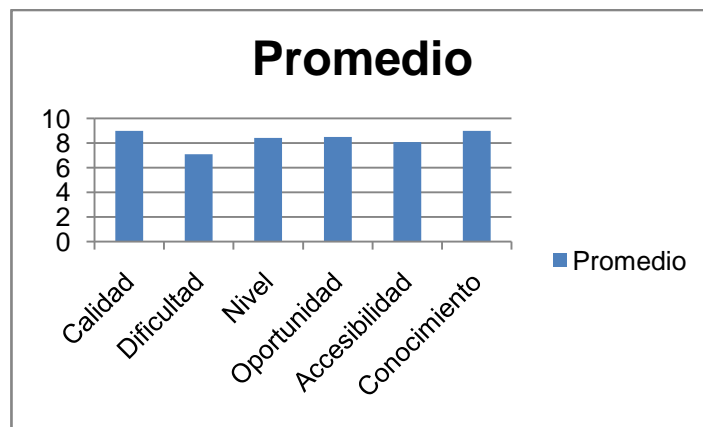
El promedio de los datos de la tabla XIV se muestra en las figuras 5 y 6.

Figura 5. **Promedio de datos de la tabla XIV, gráfica de línea**



Fuente: elaboración propia.

Figura 6. **Promedio de datos de la tabla XIV, gráfica de barras**



Fuente: elaboración propia.

Las gráficas de las figuras 5 y 6 indican que la homologación en el segundo semestre de 2015 resultó tener un incremento en sus aspectos.

Tabla XV. Datos del primer semestre de 2016

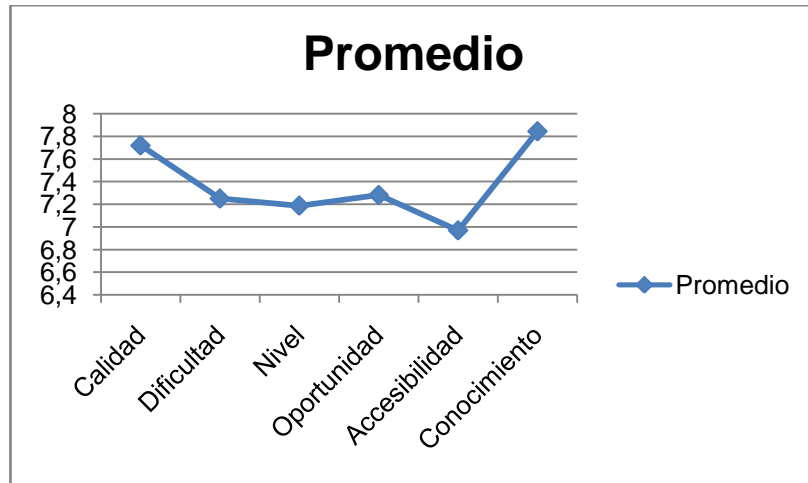
Estudiante	Calidad	Dificultad	Nivel	Oportunidad	Accesibilidad	Conocimiento
1	7	7	9	8	7	8
2	10	9	7	8	8	8
3	7	3	6	7	9	6
4	9	9	9	10	10	10
5	8	3	8	5	3	9
6	9	9	9	8	8	9
7	8	9	8	7	6	8
8	9	9	10	9	9	10
9	8	8	8	8	2	9
10	6	4	5	3	5	6
11	7	8	7	7	5	8
12	10	8	9	7	7	10
13	10	7	5	10	7	7
14	10	10	7	7	6	9
15	9	7	7	8	10	7
16	8	10	7	7	7	9
17	5	5	7	7	9	8
18	7	7	6	8	5	6
19	8	9	6	7	8	9
20	8	9	9	8	6	7
21	10	10	10	10	8	10
22	10	8	7	6	8	9
23	4	6	3	6	5	3
24	8	4	9	7	7	9
25	6	8	6	6	6	7
26	5	5	7	8	8	6
27	8	7	8	7	9	9
28	4	7	4	7	6	7
29	6	8	6	7	5	6
30	6	7	6	6	7	7
31	8	6	8	8	10	8
32	9	6	7	6	7	7

Fuente: elaboración propia.



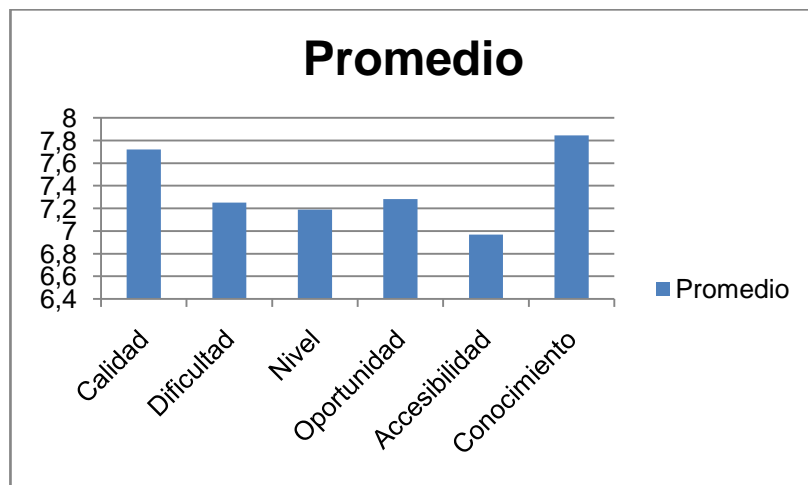
El promedio de los datos de la tabla XV se muestra en las figuras 7 y 8.

Figura 7. **Promedio de datos de la tabla XV, gráfica de línea**



Fuente: elaboración propia.

Figura 8. **Promedio de datos de la tabla XV, gráfica de barras**



Fuente: elaboración propia.

Las graficas de las figuras 7 y 8 indican que la homologación en el primer semestre de 2016 resultó tener un nivel de dificultad mayor al de los dos semestres anteriores; sin embargo, la percepción del alumno debido a su preparación mediante las tutorías académicas ayudó a que su nivel de conocimiento fuera alto, lo que hizo superar con facilidad el nivel de dificultad.

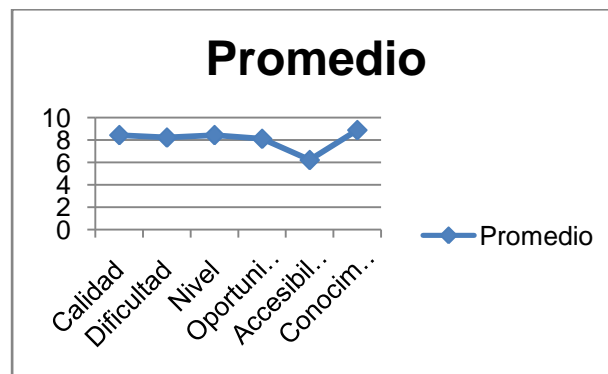
Tabla XVI. **Datos obtenidos en la encuesta a los tutores académicos**

Tutor académico	Calidad	Dificultad	Nivel	Oportunidad	Accesibilidad	Conocimiento
1	9	9	9	9	7	9
2	8	10	7	7	9	10
3	10	8	9	10	3	9
4	9	9	9	10	8	10
5	9	8	9	9	3	9
6	6	7	8	6	5	7
7	9	9	9	6	6	9
8	8	6	8	9	7	9
9	8	8	8	7	8	8

Fuente: elaboración propia.

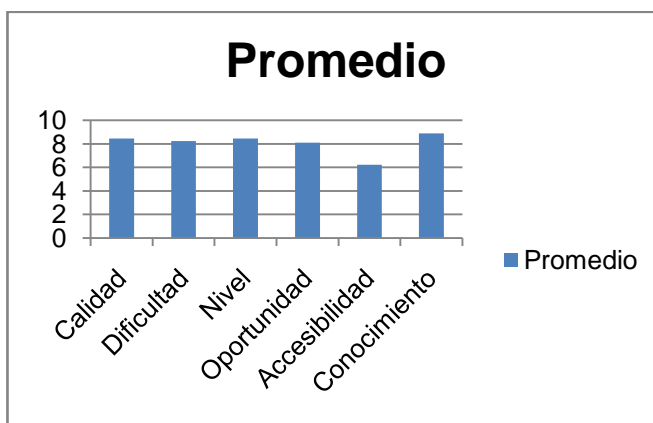
El promedio de los datos de la tabla XVI se muestra en las figuras 9 y 10.

Figura 9. **Promedio de datos de la tabla XVI, gráfica de línea**



Fuente: elaboración propia.

Figura 10. **Promedio de datos de la tabla XVI, gráfica de barras**



Fuente: elaboración propia.

Las graficas de las figuras 9 y 10 indican que la homologación desde el punto de vista de los tutores académicos es de un nivel alto en todos sus aspectos que fueron medibles.

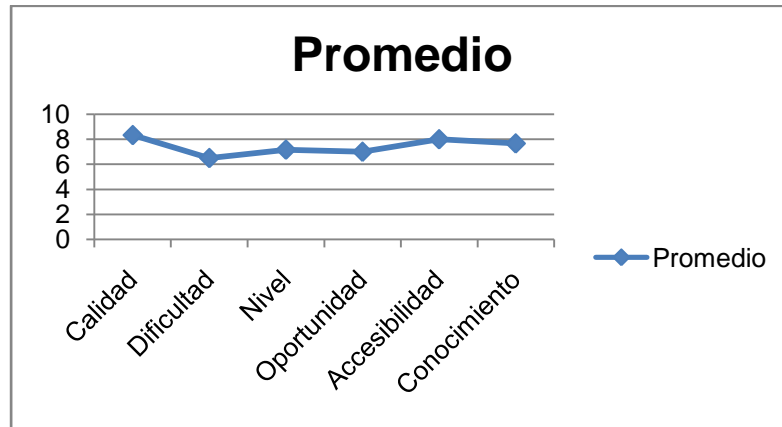
Tabla XVII. **Datos obtenidos en la encuesta a los catedráticos**

Catedrático	Calidad	Dificultad	Nivel	Oportunidad	Accesibilidad	Conocimiento
1	7	7	5	6	6	6
2	9	8	10	9	10	9
3	10	5	10	6	10	10
4	8	9	2	4	6	4
5	9	4	9	10	10	9
6	7	6	7	7	6	8

Fuente: elaboración propia.

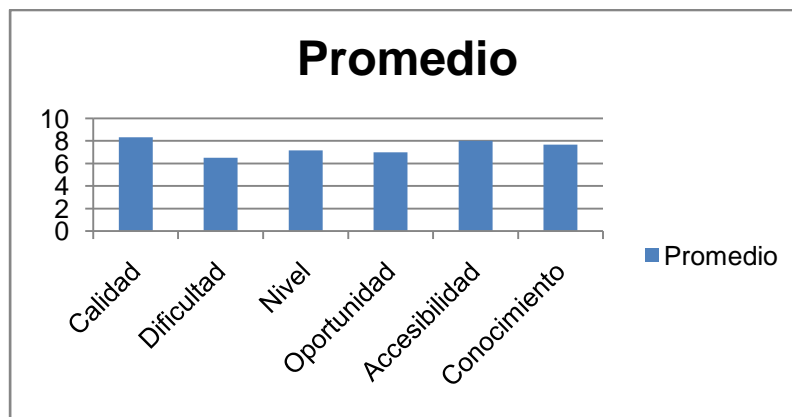
El promedio de los datos de la tabla XVI se muestra en las figuras 11 y 12.

Figura 11. Promedio de datos de la tabla XVII, gráfica de línea



Fuente: elaboración propia.

Figura 12. Promedio de datos de la tabla XVII, gráfica de barras



Fuente: elaboración propia.

Las graficas de las figuras 11 y 12 indican que la homologación desde el punto de vista de los catedráticos tiene una dificultad baja en contraste con el resto de valores ponderados que reflejan una tendencia alta.

## CONCLUSIONES

1. Los estudiantes del curso de Introducción a la Programación y Computación 1 no tienen la preparación necesaria para adquirir los conocimientos adecuados para enfrentar con mayor oportunidad la superación de los cursos siguientes que están íntimamente ligados al curso.
2. Los tutores académicos no han brindado la orientación, asesoría y apoyo a los tutores académicos sustitutos del curso de Introducción a la Programación y Computación 1, debido a la falta de una implementación para fomentar entre los tutores académicos el espíritu de compartir las experiencias adquiridas.
3. El tiempo dedicado para las diferentes actividades del curso no está optimizado, debido a que hay actividades académicas que requieren un mayor número de horas y tienen asignado poco tiempo; otras actividades tienen asignadas muchas horas y se pueden realizar en menor tiempo.
4. Los datos presentados en los cuadros y gráficas anteriores demuestran que el sistema de homologación contribuye al mejoramiento del aprendizaje del alumno elevando sus conocimientos sobre una base sólida, ayuda a mejorar el sistema de docencia mediante la implementación de técnicas para impartir las clases, confirmando la hipótesis planteada.



## RECOMENDACIONES

1. Preparar al estudiantado con conocimientos necesarios para enfrentar con mayor oportunidad de superar las siguientes clases que están íntimamente ligadas al curso mediante la implementación de la homologación del curso de Introducción a la Programación y Computación 1.
2. Fomentar entre los tutores académicos el espíritu de orientación, asesoría y apoyo con los tutores académicos sustitutos, al compartir sus experiencias y conocimientos adquiridos al impartir el curso de Introducción a la Programación y Computación 1.
3. Optimizar y rediseñar la distribución del tiempo para la mejor aplicación de las técnicas de enseñanza, a los estudiantes del curso de Introducción a la Programación y Computación 1.





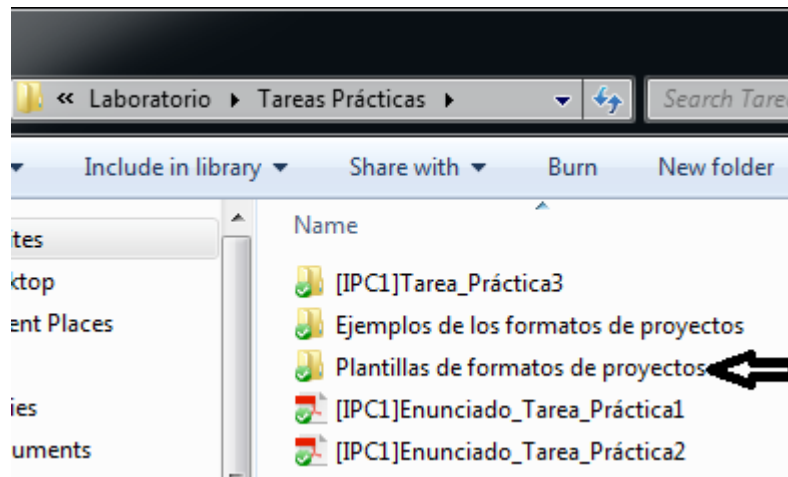
## BIBLIOGRAFÍA

1. AGUILAR, Luis Joyanes; MARTÍNEZ, Ignacio Zahonero. *Programación en C, C++, Java y UML*. México: McGraw, 2014. 767 p.
2. \_\_\_\_\_ . *Fundamentos de programación*. España: McGraw Hill, 2008. 766 p.
3. AHO, Alfred; SETHI Ravi; ULLMAN, Jeffrey. *Compiladores: principios, técnicas y herramientas*. U.S.A.: Addison Wesley Iberoamericana, 1990. 820 p.
4. AMAT, Oriol. *Aprender a enseñar: una visión práctica de la formación de formadores*. Barcelona: GESTIÓN 2000, S. A., 2000. 183 p.
5. BORES, Rosario; ROSALES, Román. *Computación: metodología lógica computacional y programación*. México: McGraw Hill, 1993. 461 p.
6. *Certificación de calidad*. [en línea]. <<https://dl.dropboxusercontent.com/u/88541175/portadas/2009/06-concepcion/index.html>>. [Consulta: 25 de abril de 2016].
7. KOFFMAN, Elliot; WOLFGANG, Paul. *Estructura de datos con C++ objetos, abstracciones y diseño*. México: McGraw Hill, 2008. 806 p.

8. MOORE, Sara; WALSH, Gary; RÍSQUEZ, Angélica. *Estrategias eficaces para enseñar en la universidad: guía para docentes comprometidos*. Madrid: NARCEA, S. A., 2012. 171 p.
9. NAKAMKURA, María Adriana Corona; VALDEZ, María de los Ángeles. *Diseño de algoritmos y su codificación en lenguaje C*. México: McGraw Hill, 2011. 326 p.
10. NECIRI, I.: *Hacia una didáctica general dinámica*. Kapelusz, Buenos Aires, 1979. p. 233-280.
11. NORTON, Petter. *Introducción a la computación*. México: McGraw Hill, 2014. 656 p.
12. PRESSMAN, Roger. *Ingeniería del software: un enfoque práctico*. México: McGraw Hill, 2010. 777 p.
13. SCHACH, Stephen. *Ingeniería de software clásica y orientada a objetos*. México: McGrawHill, 2006. 581 p.
14. UPV/EHU. *Programa SICRE métodos numéricos*. [en línea]. <<http://www.ehu.es/~mepmufov/html/ProgramaSICREMetodosNumericos.pdf>>. [Consulta: 26 de abril de 2016].
15. ZABALA, Miguel. *La enseñanza universitaria: el escenario y sus protagonistas* Madrid: NARCEA, S. A., 2007. 238 p.

## APÉNDICES

### Apéndice 1. Localización de los formatos de proyecto

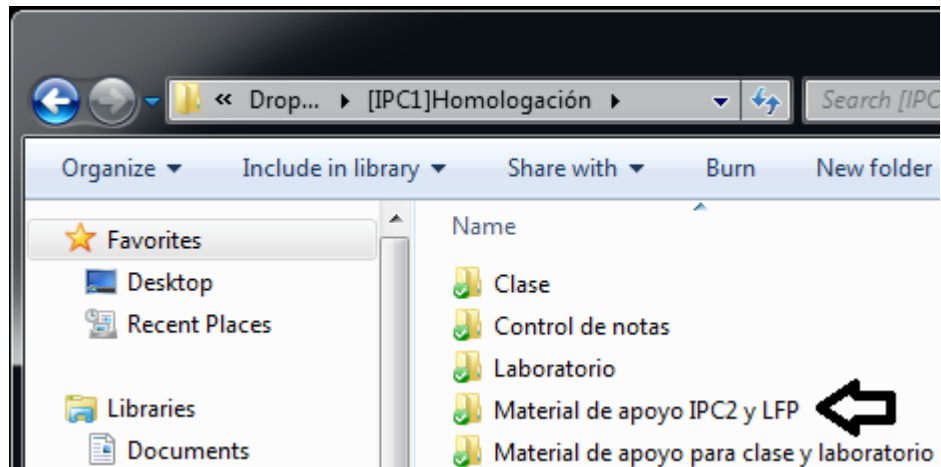


Fuente: elaboración propia, toma de pantalla.

Los formatos de proyecto se encuentran localizados en el siguiente *link*:

<https://www.dropbox.com/sh/rm8yxnjdrxnxt0p/AAArACMFmdyxseXpjLNr2Lxxa?dl=0>

## Apéndice 2. Localización de la carpeta de homologación

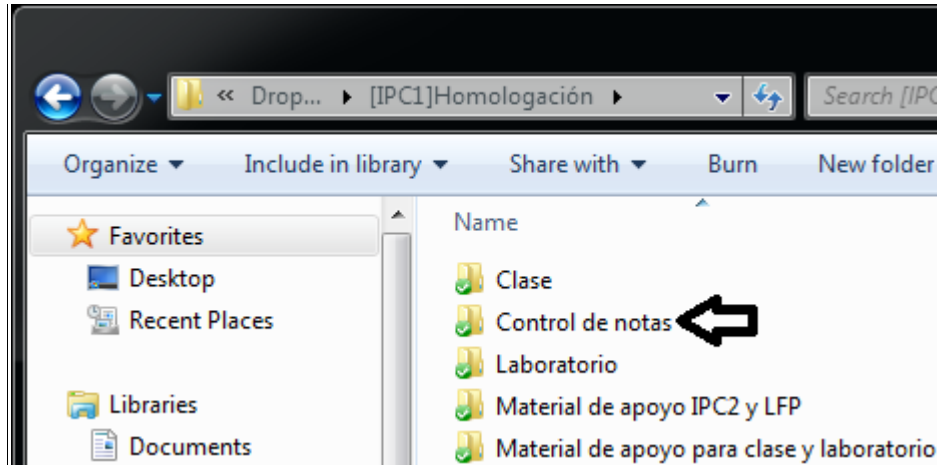


Fuente: elaboración propia, toma de pantalla.

El material de apoyo para la clase, laboratorio y cursos posteriores se encuentran localizados en el siguiente *link*:

<https://www.dropbox.com/sh/kr8r54ap7ipi6cm/AACQMSaSyWIJaDWzC4hcNrnla?dl=0>

### Apéndice 3. Plantillas de control de notas de clase y laboratorio

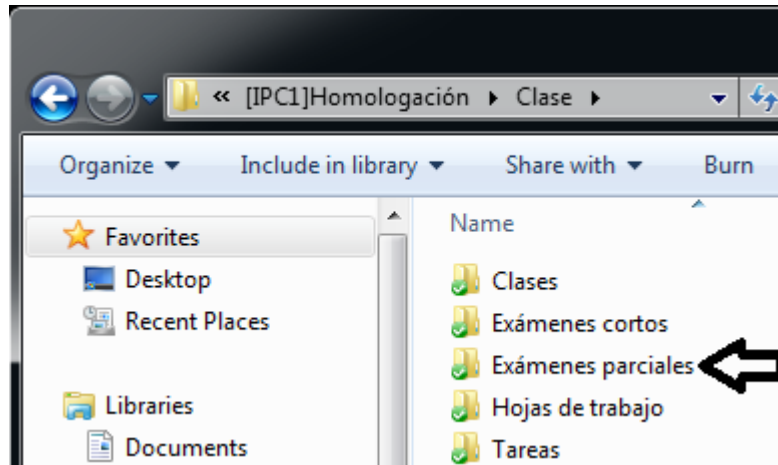


Fuente: elaboración propia, toma de pantalla de plantillas propuestas.

Las plantillas para el control de notas pueden ser descargados en los siguientes *links*:

- <https://docs.google.com/spreadsheets/d/1-QK0BTXkZMRrB29xhTaeXckvDljbuyTJo0DuQIL-m5A/edit?usp=sharing>
- <https://docs.google.com/spreadsheets/d/108qejM51TzSHTDMDIHGjHkytPv1st7dhaKgQ0mYrhjw/edit?usp=sharing>

#### Apéndice 4. **Propuesta de evaluaciones para la clase**



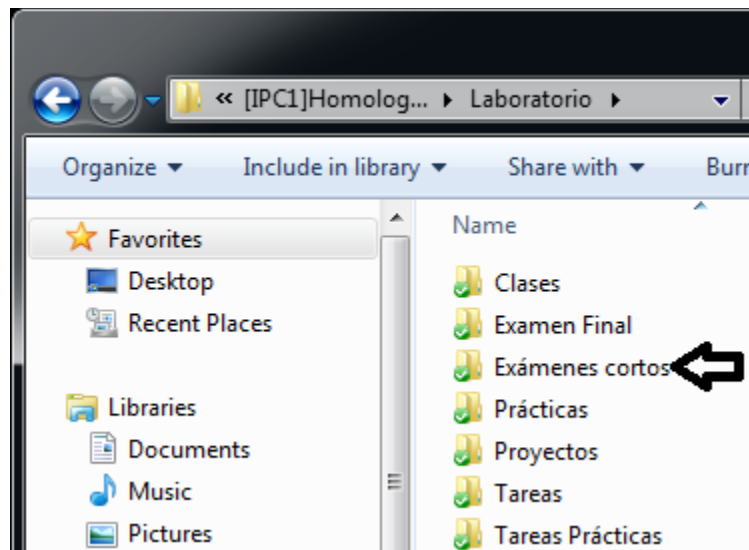
Fuente: elaboración propia, toma de pantalla .

Dentro de la carpeta de homologación y subcarpeta clase, se encuentran las propuestas sugeridas para los exámenes cortos, parciales y final de la clase.

El *link* para ingresar directamente a la carpeta de clase es:

[https://www.dropbox.com/sh/hqksl2ckqxb54bm/AAD4YuH\\_O7W\\_a0mbqR2WJn19a?dl=0](https://www.dropbox.com/sh/hqksl2ckqxb54bm/AAD4YuH_O7W_a0mbqR2WJn19a?dl=0)

## Apéndice 5. Propuesta de evaluaciones para el laboratorio




Fuente: elaboración propia.

Dentro de la carpeta de homologación y subcarpeta laboratorio, se encuentran las propuestas sugeridas para los exámenes cortos y el examen final del laboratorio.

El *link* para ingresar directamente a la carpeta del laboratorio es:


<https://www.dropbox.com/sh/wm9dgeuf45is2gt/AAAYIDAWhh5hQ2mmt8liQ5Hha?dl=0>

Apéndice 6. **Plantilla control de notas de la clase**

 <b>Listado general</b>		Actividades															Nota final
		Tareas			Hojas de trabajo			Exámenes cortos			Exámenes parciales			Laboratorio	Examen Final		
		3 Puntos.			3 Puntos.			3 Puntos.			1er parcial	2do. parcial	3er. parcial	Nota enviada por tutores	25 Puntos.		
		2/2/2016	10/4/2016	19/4/2016	9/2/2016	29/3/2016	21/4/2016	16/2/2016	5/4/2016	28/4/16	18/2/2016	7/4/2016	3/5/2016		25 pts.		
		1 pts.	1 pts.	1 pts.	1 pts.	1 pts.	1 pts.	1 pts.	1 pts.	1 pts.	8pts.	14 pts.	14 pts.	30 pts.	25 pts.		
111111111	Nota Ideal	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100.00	
																0.00	
																0.00	
																0.00	
																0.00	
																0.00	
																0.00	
																0.00	

Fuente: elaboración propia.


Apéndice 7. **Plantilla de control de notas del laboratorio**

 <b>Listado general</b>		Actividades															Nota final
		Tareas			Tareas Prácticas			Prácticas			Cortos		Proyectos.		Examen Final.		
		6 Puntos.			11 Puntos.			15 Puntos.			Corto 1	Corto 2	40 Puntos.		8 Puntos.		
		2 pts.	2 pts.	2 pts.	3 pts.	3 pts.	5 pts.	5 pts.	5 pts.	5 pts.	10 pts.	10 pts.	15 pts.	25 pts.	8 pts.		
111111111	Nota Ideal	100	100	100	100	100	100	100	100	100.00	100.00	0	0	100	60.00		
		0	0	0	0	0	0	0	0	0.00	0.00	0	0	0	0.00		
		0	0	0	0	0	0	0	0	0.00	0.00	0	0	0	0.00		
		0	0	0	0	0	0	0	0	0.00	0.00	0	0	0	0.00		
		0	0	0	0	0	0	0	0	0.00	0.00	0	0	0	0.00		
		0	0	0	0	0	0	0	0	0.00	0.00	0	0	0	0.00		
		0	0	0	0	0	0	0	0	0.00	0.00	0	0	0	0.00		
		0	0	0	0	0	0	0	0	0.00	0.00	0	0	0	0.00		

Fuente: elaboración propia



## Apéndice 8. Boleta de encuesta



*Encuesta a estudiantes primer semestre  
2015*

**\* Required**

*En una escala de 1 a 10 seleccione la calidad de la metodología de IPCI \**

1 2 3 4 5 6 7 8 9 10

*En una escala de 1 a 10 seleccione la dificultad de la metodología de IPCI \**

1 2 3 4 5 6 7 8 9 10

*En una escala de 1 a 10 seleccione el nivel que sus estudiantes alcanzaron haciendo uso de la metodología de IPCI \**

1 2 3 4 5 6 7 8 9 10

*En una escala de 1 a 10 seleccione el nivel de oportunidad de aprobar los cursos posteriores a IPCI que tienen sus estudiantes con el uso de la metodología de estandarización \**

1 2 3 4 5 6 7 8 9 10

*En una escala de 1 a 10 seleccione la accesibilidad de la metodología de IPCI \**  
accesibilidad se refiere a si el estudiante hace esfuerzo y aun así no aprueba.

1 2 3 4 5 6 7 8 9 10

*En una escala de 1 a 10 seleccione el conocimiento que se le otorgo a los estudiantes utilizando la metodología de IPCI \**

1 2 3 4 5 6 7 8 9 10

*Cualquier mensaje que desee dejar este es el espacio. \**

Fuente: elaboración propia



# ANEXOS

## Anexo 1. Contenido del curso, parte 1

1. Introducción
  - 1.1 Conceptos computacionales
    - 1.1.1 Computadora
    - 1.1.2 Hardware
    - 1.1.3 Firmware
    - 1.1.4 Software
  - 1.2 Organización
    - 1.2.1 CPU
    - 1.2.2 Memoria principal
    - 1.2.3 Memoria secundaria
    - 1.2.4 Dispositivos E/S
    - 1.2.5 Periféricos
  - 1.3 Lenguajes de programación
    - 1.3.1 Lenguaje de máquina
    - 1.3.2 Lenguajes de bajo nivel
    - 1.3.3 Lenguajes de alto nivel
  - 1.4 Resolución de problemas computacionales
    - 1.4.1 Toma de requerimientos
      - 1.4.1.1 Plan de requerimientos
    - 1.4.2 Análisis del problema
    - 1.4.3 Diseño del algoritmo
    - 1.4.4 Codificación
    - 1.4.5 Compilación y ejecución
    - 1.4.6 Verificación y depuración
    - 1.4.7 Documentación
      - 1.4.7.1 Plan de proyectos de software
      - 1.4.7.2 Estrategias de Marketing

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 2. Contenido del curso, parte 2

2. Programación modular y estructuras básicas
  - 2.1 Secuencial y procedural: metodología Top-Down.
  - 2.2 Variables: concepto, manipulación y asignación.
  - 2.3 Tipos de datos (primitivos y contruidos por el usuario)
  - 2.4 Operadores aritméticos
  - 2.5 Operadores relacionales y lógicos
  - 2.6 Estructuras de control condicionales
    - 2.6.1 Si – Sino (if – else)
    - 2.6.2 En caso (switch / case)
  - 2.7 Estructuras cíclicas (bucles, loops)
    - 2.7.1 Para (for)
    - 2.7.2 Mientras (while)
    - 2.7.3 Repetir - Hasta (Repeat – Until / do-while)
  - 2.8 Las rutinas
    - 2.8.1 Procedimiento y función
    - 2.8.2 Entorno de las variables (alcance o ámbito)
    - 2.8.3 Los parámetros
      - 2.8.3.1 Por variables
      - 2.8.3.2 Por valor
    - 2.8.4 El valor de retorno
  - 2.9 Modularidad
    - 2.9.1 Segmentos por rutina
    - 2.9.2 Uso adecuado de prefijos
    - 2.9.3 Documentación interna
    - 2.9.4 Legibilidad y entendimiento
  - 2.10 Recursividad

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 3. Contenido del curso, parte 3

- 3. Metodología orientada a objetos
  - 3.1 Concepto de abstracción y clasificación
  - 3.2 Clases y objetos
  - 3.3 Mensajes y métodos
  - 3.4 El principio el encapsulamiento
  - 3.5 Los miembros de una clase
    - 3.5.1 Atributos
    - 3.5.2 Métodos (operaciones)
    - 3.5.3 Constructores y destructores
  - 3.6 Modificadores de visibilidad
    - 3.6.1 Privado
    - 3.6.2 Público
    - 3.6.3 Protegido
  - 3.7 Relaciones entre clases y objetos
    - 3.7.1 Asociación
    - 3.7.2 Agregación y composición
    - 3.7.3 Herencia (simple y múltiple)
  - 3.8 Polimorfismo
    - 3.8.1 Sobrecarga de métodos
    - 3.8.2 Virtualización
  - 3.9 Construcciones abstractas
    - 3.9.1 Clase abstracta
    - 3.9.2 Interface
  - 3.10 Conceptos avanzados
    - 3.10.1 Miembros estáticos (static) y miembros de instancia
    - 3.10.2 Referencia "this"
    - 3.10.3 Clases paramétricas (plantilla de clases).
  - 3.11 Principios básicos de UML (diagrama de clases)
    - 3.11.1 Definición de clases y sus relaciones
    - 3.11.2 Ambito de las propiedades, Métodos
    - 3.11.3 Diseño de programas
    - 3.11.4 Asociaciones y restricciones, clases de asociaciones, Multiplicidad, Dependencia
    - 3.11.5 Relaciones múltiples (asociativas) y reflexivas

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 4. Contenido del curso parte, 4

- 4. Programación orientada a objetos – Laboratorio
  - 4.1 Lenguaje Java (clases, atributos, métodos)
  - 4.2 Constructor y destructor
  - 4.3 Tipos de atributos
  - 4.4 Operaciones (aritméticos, relacionales y lógicos)
  - 4.5 Estructuras de control condicionales (if – else, switch, ?)
  - 4.6 Estructuras cíclicas (for, while, do-while)
  - 4.7 Tipos de accesos (public, private, protected)
  - 4.8 Manejo de variables
  - 4.9 Métodos: funciones/procedimientos y recursividad.
- 5. Estructuras algorítmicas
  - 5.1 Arreglos vectoriales de datos
    - 5.1.1 Conceptos: elementos, longitud, indexación, representación en memoria.
    - 5.1.2 Arreglos bidimensionales (matrices): representación en memoria.
    - 5.1.3 Arreglos n-dimensionales (multidimensionales).
    - 5.1.4 Ejemplos, técnicas de acceso y recomendaciones.
  - 5.2 Las cadenas de caracteres
    - 5.2.1 Concepto: diferencia con arreglos de caracteres.
    - 5.2.2 Cadenas estáticas (ej: String) y dinámicas (ej: StringBuffer).
    - 5.2.3 Operaciones y métodos.
  - 5.3 Búsqueda de datos en arreglos
    - 5.3.1 Secuencial
    - 5.3.2 Binaria
  - 5.4 Ordenamiento de datos en arreglos
    - 5.4.1 Burbuja
    - 5.4.2 Por inserción
    - 5.4.3 Por selección
    - 5.4.4 Quick Sort
  - 5.5 La pila (Stack)
    - 5.5.1 Política de acceso a datos (LIFO) y operaciones.
  - 5.6 La cola (Queue)
    - 5.6.1 Política de acceso a datos (FIFO) y operaciones.
    - 5.6.2 Representaciones: simple y circular.
  - 5.7 El uso de Heap
    - 5.7.1 Asociación a la pila
    - 5.7.2 Tomar y devolver al heap
    - 5.7.3 Usos con las pilas y las colas

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 5. Contenido del curso, parte 5

- 6. Colecciones de datos
  - 6.1 Los índices y el apuntador simple
    - 6.1.1 El apuntador subíndice
    - 6.1.2 Almacenamiento
    - 6.1.3 Ordenamiento
  - 6.2 Los registros
    - 6.2.1 Concepto y definición por campos
- 7. Flujos de bytes y manipulación de archivos
  - 7.1 Concepto: modelo productor-consumidor y flujo (stream).
  - 7.2 Tipos de flujos
  - 7.3 Tipos de archivos
    - 7.3.1 Archivos de texto
    - 7.3.2 Archivos binarios
  - 7.4 Operaciones básicas
    - 7.4.1 Abrir y cerrar
    - 7.4.2 Lectura, escritura y posicionamiento
    - 7.4.3 Localización del final del archivo
- 8. Los tipos de datos abstractos
  - 8.1 Tipos de apuntadores (estáticos y dinámicos)
  - 8.2 Listas simples
  - 8.3 Listas doblemente encadenadas
  - 8.4 Pilas usando listas
  - 8.5 Colas usando listas
  - 8.6 Listas ortogonales
  - 8.7 Listas n-encadenadas

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 6. Contenido del laboratorio, parte 1

**Módulo introducción a la computación.**

**Módulo introducción a HTML.**

**Módulo de importancia de diseño para el desarrollo del software.**

- 1. **Fundamentos de Programación**
  - 1.1 Introducción a Algoritmos.
  - 1.2 Diagramas de Flujo.
  - 1.3 Introducción a la programación.
- 2. **Programación Estructurada**
  - 2.1 Tipos de Variables.
  - 2.2 Estructuras de Datos.
  - 2.3 Estructuras de Control.
  - 2.4 Funciones y Procedimientos.
- 3. **Paradigma de Objetos y UML**
  - 3.1 Introducción al Paradigma Orientado a Objetos.
  - 3.2 Introducción a UML.
  - 3.3 Definiciones.
  - 3.4 Simbología.
- 4. **Introducción a Java**
  - 4.1 Que es Java.
  - 4.2 Versiones y ambiente de Java(JDK, IDE, máquina virtual, etc.).
  - 4.3 Características de Java.
  - 4.4 Tipos de programas en Java.
  - 4.5 Componentes del lenguaje Java(variables, constantes, tipos de datos etc.).
  - 4.6 Estructuras de control.
  - 4.7 Arreglos.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 7. Contenido del laboratorio, parte 2

- 5. Programación Orientada a Objetos en Java(POO)**
  - 5.1 Conceptos de la programación orientada a objetos.
  - 5.2 Creación de clases y objetos.
  - 5.3 Manejo de memoria.
  - 5.4 Casteo de datos.
  - 5.5 Constructores.
  - 5.6 Herencia.
  - 5.7 Polimorfismo.
  
- 6. Clases abstractas, interfaces y paquetes**
  - 6.1 Control de acceso a clases y métodos.
  - 6.2 Clases abstractas y métodos.
  - 6.3 Interfaces.
  - 6.4 Paquetes.
  
- 7. Manejo de Excepciones**
  - 7.1 Comprensión de excepciones y el manejo de estas.
  - 7.2 Sentencia Try Catch.
  - 7.3 Try anidados.
  
- 8. Manejo de hilos en Java**
  - 8.1 Creación de hilos.
  - 8.2 Multi-hilo.
  - 8.3 Animación empleando hilos.
  
- 9. Introducción a AWT y SWING**
  - 9.1 Frames.
  - 9.2 Botones.
  - 9.3 Textbox, etc..
  - 9.4 Ejemplos.
  
- 10. Archivos**
  - 10.1 Introducción a flujo de datos.
  - 10.2 Flujo de entrada y salida de datos.
  - 10.3 Abrir, Cerrar y Modificar Archivos.
  - 10.4 Buffer de lectura y escritura.
  
- 11. Listas y Colas**
  - 11.1 Pilas usando listas.
  - 11.2 Colas usando listas.
  - 11.3 Listas n-encadenadas.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 8. Formato de proyecto 1, parte 1

INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1  
 Cat. Ing. Moisés Velásquez




<b>Formato de inscripción</b>	<b>FP-01</b>
-------------------------------	--------------

1. Seleccione con una X las características que utilizara en su proyecto

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
					
Hardware Sensores	Software Mobile Web	Social Crowd S.	Análisis de datos Estadísticas	GPS Maps	Cámara Mando a distancia

Otro (si utilizara un dispositivo electrónico o interface de software específico)

2. Identificación de los participantes:  = Nombre del Project Manager

	CARNE	APELLIDO	NOMBRE	Rol	Fotografía

3. Escriba el nombre del proyecto:

4. Describa cual es el problema que resuelve o la necesidad que satisface

Necesidad/Problema:

Fuente: formato de Arquitectura de Computadoras y Ensambladores 2 Fiusac.

## Anexo 9. Formato de proyecto 1, parte 2

INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1  
Cat. Ing. Moisés Velásquez



5. Describa la justificación y la solución propuesta

<b>Justificación</b>
----------------------

<b>Solución propuesta</b>
---------------------------

Fuente: formato de Arquitectura de Computadoras y Ensambladores 2 Fiusac.

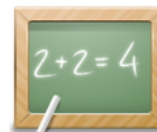
## Anexo 10. Formato de proyecto 2

INTRODUCCIÓN A LA PROGRAMACIÓN 1  
Cat. Ing. Moisés Velásquez



<b>Formato de planos</b>	<b>FP-02</b>
--------------------------	--------------

En este apartado debe describir mediante bocetos, croquis, diagramas UML o cualquier técnica de modelado de forma coherente, el diseño del producto. El objetivo es describir el objeto de forma que cualquier persona pueda ser capaz de entender como las partes que lo forman y como debe funcionar. Tome EN CONSIDERACION EL EJEMPLO EN LA CARPETA DEL ENUNCIADO DE LA TAREA PRÁCTICA.



<b>Nombre del caso:</b>		<b>ID:</b>	
<b>Autor</b>		<b>Fecha Creación:</b>	
Por cada apartado utilice un cuadro diferente y recuérdese de colocarle un ID			

Fuente: formato de Arquitectura de Computadoras y Ensambladores 2 FiUSAC.



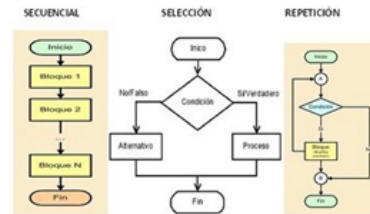
## Anexo 11. Formato de proyecto 4

INTRODUCCIÓN A LA PROGRAMACIÓN 1  
Cat. Ing. Moisés Velásquez



<b>Formato de diagramas</b>	<b>FP-04</b>
-----------------------------	--------------

En este apartado debe describir mediante diagramas UML, el diseño del proyecto 1. El objetivo es llevar en orden los diagramas solicitados en el enunciado del proyecto 1.



<b>Nombre diagrama:</b>	<b>ID:</b>
<b>Autor</b>	<b>Fecha Creación:</b>
Por cada diagrama utilice un cuadro diferente y recuérdese de colocarle un ID	

Fuente: formato de Arquitectura de Computadoras y Ensambladores 2 Fiusac.

## Anexo 12. Programa del curso, parte 1

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA  
ESCUELA DE CIENCIAS



### PROGRAMA DEL CURSO

<b>I. Información General</b>			
<b>NOMBRE DEL CURSO:</b> Introducción a la Programación y Computación 1			
<b>CODIGO:</b>	0770	<b>CREDITOS:</b>	4
<b>ESCUELA:</b>	Ciencias y Sistemas	<b>AREA A LA QUE PERTENECE:</b>	Desarrollo de Software
<b>PRE REQUISITO:</b>	34 créditos y 0103 Matemática Básica 2	<b>POST REQUISITO:</b>	0771 Introducción a la Programación y Computación 2 0796 Lenguajes Formales y de Programación.
<b>CATEGORIA:</b>	Obligatorio	<b>SEMESTRE:</b>	1er. Semestre 2016
<b>CATEDRÁTICO (A):</b>	Ing. Neltali de Jesús Calderón Méndez	<b>AUXILIAR:</b>	Jhonatan Wilfredo Pú Morales Jose Eliezer Pop Estrada
<b>EDIFICIO:</b>	T-7	<b>SECCIÓN:</b>	E
<b>SALÓN DEL CURSO:</b>	212	<b>SALON DEL LABORATORIO:</b>	014
<b>HORAS POR SEMANA DEL CURSO:</b>	4	<b>HORAS POR SEMANA DEL LABORATORIO:</b>	2
<b>DÍAS QUE SE IMPARTE EL CURSO:</b>	Martes y Jueves	<b>DÍAS QUE SE IMPARTE EL LABORATORIO:</b>	Jueves
<b>HORARIO DEL CURSO:</b>	7:10 AM - 8:50 AM	<b>HORARIO DEL LABORATORIO:</b>	9:10 AM - 10:50 AM

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 13. Programa del curso, parte 2

### II. DESCRIPCION DEL CURSO:

El curso es el acercamiento inicial del estudiante de la carrera de sistemas, a la programación mediante el uso de disciplinas y metodologías especializadas. El curso se fundamenta en el concepto de algoritmo para la resolución de problemas de programación, enfatizando el uso del paradigma orientado a objetos. Se introducen conceptos básicos de UML como guía para el diseño de sistemas orientados a objetos. Se acerca al estudiante al conocimiento de los principales algoritmos de búsquedas y ordenamientos. Se cubre una parte importante de las estructuras de datos, los tipos de datos abstractos. Asimismo, el estudiante conocerá el lenguaje Java como el lenguaje oficial de programación del curso.

### III. OBJETIVOS:

#### General

- Lograr que el estudiante adquiera la habilidad de programar y los conocimientos básicos de la programación utilizando el paradigma orientado a objetos.

#### Específico

1. Integrar al estudiante a la tecnología de la computación.
2. Conocer las diferentes metodologías de programación.
3. Organizar soluciones utilizando un lenguaje de programación.
4. Adquirir la habilidad de hacer algoritmos.
5. Aprender a elaborar diseños de clases preliminares en UML.
6. Analizar los problemas con metodología orientada a objetos.
7. Conocer el lenguaje Java como el primer lenguaje de programación para computadoras.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 14. Programa del curso, parte 3

### IV. METODOLOGÍA:

- Clases diarias.
- Elaboración de investigaciones y tareas.
- Práctica de exámenes cortos y parciales.
- Laboratorio taller.
- Elaboración de proyectos de programación.

### V. EVALUACION DEL RENDIMIENTO ACADEMICO:

Clase teórica (70 puntos)		Clase práctica (30 puntos)	
Descripción	Pts.	Descripción	Pts.
Tareas, Cortos y Asistencia	09	Tareas y Tareas Prácticas	17
Primer parcial	08	Prácticas	15
Segundo parcial	14	Proyectos	40
Tercer parcial	14	Exámenes cortos	20
Laboratorio	30		-----
Zona total	75	Zona total	92
Examen Final	25	Examen Final	08
	-----		-----
Total	100	Total	100

El curso se gana con 61 pts. de 100. Y el laboratorio de gana con 61 pts. de 100.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 15. Programa de curso, parte 4

<b>VI. CONTENIDO</b>
1. Introducción
1.1 Conceptos computacionales
1.1.1 Computadora
1.1.2 Hardware
1.1.3 Firmware
1.1.4 Software
1.2 Organización
1.2.1 CPU
1.2.2 Memoria principal
1.2.3 Memoria secundaria
1.2.4 Dispositivos E/S
1.2.5 Periféricos
1.3 Lenguajes de programación
1.3.1 Lenguaje de máquina
1.3.2 Lenguajes de bajo nivel
1.3.3 Lenguajes de alto nivel
1.4 Resolución de problemas computacionales
1.4.1 Toma de requerimientos
1.4.1.1 Plan de requerimientos
1.4.2 Análisis del problema
1.4.3 Diseño del algoritmo
1.4.4 Codificación
1.4.5 Compilación y ejecución
1.4.6 Verificación y depuración
1.4.7 Documentación
1.4.7.1 Plan de proyectos de software
1.4.7.2 Estrategias de Marketing
2. Programación modular y estructuras básicas
2.1 Secuencial y procedural: metodología Top-Down.
2.2 Variables: concepto, manipulación y asignación.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 16. Programa del curso, parte 5

2.3 Tipos de datos (primitivos y construidos por el usuario)
2.4 Operadores aritméticos
2.5 Operadores relacionales y lógicos
2.6 Estructuras de control condicionales
2.6.1 Si – Sino (if – else)
2.6.2 En caso (switch / case)
2.7 Estructuras cíclicas (bucles, loops)
2.7.1 Para (for)
2.7.2 Mientras (while)
2.7.3 Repetir - Hasta (Repeat – Until / do-while)
2.8 Las rutinas
2.8.1 Procedimiento y función
2.8.2 Entorno de las variables (alcance o ámbito)
2.8.3 Los parámetros
2.8.3.1 Por variables
2.8.3.2 Por valor
2.8.4 El valor de retorno
2.9 Modularidad
2.9.1 Segmentos por rutina
2.9.2 Uso adecuado de prefijos
2.9.3 Documentación interna
2.9.4 Legibilidad y entendimiento
2.10 Recursividad

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 17. Programa del curso, parte 6

3. Metodología orientada a objetos
3.1 Concepto de abstracción y clasificación
3.2 Clases y objetos
3.3 Mensajes y métodos
3.4 El principio el encapsulamiento
3.5 Los miembros de una clase
3.5.1 Atributos
3.5.2 Métodos (operaciones)
3.5.3 Constructores y destructores
3.6 Modificadores de visibilidad
3.6.1 Privado
3.6.2 Público
3.6.3 Protegido
3.7 Relaciones entre clases y objetos
3.7.1 Asociación
3.7.2 Agregación y composición
3.7.3 Herencia (simple y múltiple)
3.8 Polimorfismo
3.8.1 Sobrecarga de métodos
3.8.2 Virtualización
3.9 Construcciones abstractas
3.9.1 Clase abstracta
3.9.2 Interfase
3.10 Conceptos avanzados
3.10.1 Miembros estáticos (static) y miembros de instancia
3.10.2 Referencia "this"
3.10.3 Clases paramétricas (plantilla de clases).
3.11 Principios básicos de UML (diagrama de clases)
3.11.1 Definición de clases y sus relaciones
3.11.2 Ámbito de las propiedades, Métodos
3.11.3 Diseño de programas
3.11.4 Asociaciones y restricciones, clases de asociaciones, Multiplicidad, Dependencia
3.11.5 Relaciones múltiples (asociativas) y reflexivas

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 18. Programa del curso, parte 7

4. Programación orientada a objetos – Laboratorio
4.1 Lenguaje Java (clases, atributos, métodos)
4.2 Constructor y destructor
4.3 Tipos de atributos
4.4 Operaciones (aritméticos, relacionales y lógicos)
4.5 Estructuras de control condicionales (if – else, switch, ?:)
4.6 Estructuras cíclicas (for, while, do-while)
4.7 Tipos de accesos (public, private, protected)
4.8 Manejo de variables.
4.9 Métodos: funciones/procedimientos y recursividad.
5. Estructuras algorítmicas
5.1 Arreglos vectoriales de datos
5.1.1 Conceptos: elementos, longitud, indexación, representación en memoria.
5.1.2 Arreglos bidimensionales (matrices): representación en memoria.
5.1.3 Arreglos n-dimensionales (multidimensionales).
5.1.4 Ejemplos, técnicas de acceso y recomendaciones.
5.2 Las cadenas de caracteres
5.2.1 Concepto: diferencia con arreglos de caracteres.
5.2.2 Cadenas estáticas (ej: String) y dinámicas (ej: StringBuffer).
5.2.3 Operaciones y métodos.
5.3 Búsqueda de datos en arreglos
5.3.1 Secuencial
5.3.2 Binaria
5.4 Ordenamiento de datos en arreglos
5.4.1 Burbuja
5.4.2 Por inserción
5.4.3 Por selección

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 19. Programa del curso, parte 8

6. Colecciones de datos
  - 6.1 Los índices y el apuntador simple
    - 6.1.1 El apuntador subíndice
    - 6.1.2 Almacenamiento
    - 6.1.3 Ordenamiento
  - 6.2 Los registros
    - 6.2.1 Concepto y definición por campos
7. Flujos de bytes y manipulación de archivos
  - 7.1 Concepto: modelo productor-consumidor y flujo (stream).
  - 7.2 Tipos de flujos
  - 7.3 Tipos de archivos
    - 7.3.1 Archivos de texto
    - 7.3.2 Archivos binarios
  - 7.4 Operaciones básicas
    - 7.4.1 Abrir y cerrar
    - 7.4.2 Lectura, escritura y posicionamiento
    - 7.4.3 Localización del final del archivo
8. Los tipos de datos abstractos
  - 8.1 Tipos de apuntadores (estáticos y dinámicos)

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 20. Programa del curso, parte 9

- 8.3 Listas doblemente encadenadas
- 8.4 Pilas usando listas
- 8.5 Colas usando listas
- 8.6 Listas ortogonales
- 8.7 Listas n-encadenadas

### VII. CLAUSULAS RESTRICTIVAS:

El perfil del estudiantes de la facultad de Ingeniería de la Universidad de San Carlos de Guatemala, exige una alta calidad en la excelencia académica y ética profesional. Se establecen en este curso los siguientes lineamientos que regulan el comportamiento del estudiante:

- Copias en exámenes, cortos, proyectos, tareas e investigaciones tienen cero de nota.
- La reposición de cualquier parcial se hará tomando la misma nota del examen siguiente, siempre y cuando no tenga más de 4 inasistencias a clase y se justifique, debidamente comprobado, por escrito.
- El examen final NO tiene reposición.
- No hay reposición de proyectos.
- Cualquier proyecto, tarea o investigación que se entregue después de la fecha calendarizada tiene 30 puntos menos cada día de atraso.
- Los exámenes resueltos a lápiz no tienen derecho a revisión.
- Es obligatorio ganar el laboratorio para tener derecho a evaluación total del curso.
- Para optar a sustentar cada uno de los exámenes parciales deberá entregarse completamente resuelta cada una de las tareas especiales pre-examen.
- Para tener derecho a la revisión de la zona final es obligatorio haber asistido a dos exámenes parciales y al examen final.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 21. Programa del curso, parte 10

**VIII. BIBLIOGRAFIA:**

- JOYANES, L. y ZAHONERO, I. "Programación en Java 2 (algoritmos, estructura de datos y programación orientada a objetos)". España, McGraw-Hill / Interamericana de España, S. A. 2002, PP 725
- BUDD, Timothy. "Introducción a la programación orientada a objetos", EUA, Addison-Wesley, Iberoamericana, S. A. 1994, PP. 409
- JOYANES, L. "Programación en Turbo Pascal Versiones 5.5, 6.0, y 7.0", (2da Edición), México, McGraw-Hill / Interamericana de España, S. A. 1995, PP. 914
- Manuales de Referencia de Java, <<http://www.sun.com/java>>.
- Cualquier otro material (escrito o digital) entregado en clase.

Secciones de IPC 1		
Sección A	Ing. Marlon Orellana	T-7 201 07:10 - 08:50 Ma-Ju
Sección B	Ing. William Escobar	T-3 014 07:10 - 08:50Ma- Ju
Sección C	Ing. Moisés Velásquez	T-3 111 07:10 - 08:50Ma- Ju
Sección D	Ing. Herman Véliz	T-3 413 07:10 - 08:50Ma- Ju
Sección E	Ing. Neftalí Calderón	T-7 202 07:10 - 08:50Ma- Ju

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 22. Programa del laboratorio, parte 1

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA  
ESCUELA DE CIENCIAS



NOMBRE DEL CURSO: Introducción a la Programación y computación 1			
CÓDIGO:	0770	CRÉDITOS:	4
ESCUELA:	Ciencias y Sistemas	AREA A LA QUE PERTENECE:	Desarrollo de Software
PRE REQUISITO:	34 créditos y 0130 Matemática Básica 2	POST REQUISITOS:	0771 Introducción a la Programación y Computación 2, 0796 Lenguajes Formales y de Programación.
CATEGORIA:	Obligatorio	SEMESTRE:	1er. Semestre 2016
CATEDRATICO(A):	Ing. Neftalí de Jesús Calderón Méndez	AUXILIAR:	Jhonatan Wilfredo Pú Morales Jose Eliezer Pop Estrada
EDIFICIO:	T - 7	SECCION:	E
SALON DEL CURSO:	202	SALON DEL LABORATORIO:	014
HORAS POR SEMANA DEL CURSO	4	HORAS POR SEMANA DEL LABORATORIO	2
DIAS QUE SE IMPARTE EL CURSO	Martes y Jueves	DIAS QUE SE IMPARTE EL LABORATORIO	Jueves
HORARIO DEL CURSO	7:10 AM -- 8:50 AM	HORARIO DEL LABORATORIO	9:00 AM -- 11:00 AM

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 23. Programa del laboratorio, parte 2

### DESCRIPCIÓN DEL CURSO:

El curso es el acercamiento inicial del estudiante de la carrera de sistemas, a la programación mediante el uso de disciplinas y metodologías especializadas. El curso se fundamenta en el concepto de algoritmo para la resolución de problemas de programación, enfatizando el uso del paradigma orientado a objetos. Se introducen conceptos básicos de UML como guía para el diseño de sistemas orientados a objetos.

Se acerca al estudiante al conocimiento de los principales algoritmos de búsquedas y ordenamientos. Se cubre una parte importante de las estructuras de datos, los tipos de datos abstractos. Asimismo, el estudiante conocerá el lenguaje Java como el lenguaje oficial de programación del curso.

### OBJETIVOS:

#### General

- Lograr que el estudiante adquiera la habilidad de programar y los conocimientos básicos de la programación utilizando el paradigma orientado a objetos.

#### Específico

- Integrar al estudiante a la tecnología de la computación.
- Conocer las diferentes metodologías de programación.
- Organizar soluciones utilizando un lenguaje de programación.
- Adquirir la habilidad de hacer algoritmos.
- Aprender a elaborar diseños de clases preliminares en UML.
- Analizar los problemas con metodología orientada a objetos.
- Conocer el lenguaje Java como el primer lenguaje de programación para computadoras.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 24. Programa del laboratorio, parte 3

### METODOLOGÍA:

- Clases presenciales con contenido audiovisual, clases virtuales, lecturas, etc.
- Capacitación personalizada al estudiantado.
- Elaboración de Actividades.
  - Tareas e Investigaciones.
  - Prácticas.
  - Proyectos.
  - Evaluaciones.
  - Evaluación Final.

### REQUISITOS:

- Es obligatorio aprobar ambos proyecto de laboratorio con una nota de 61 puntos.
- El laboratorio se aprueba con 61 puntos.
- Las actividades a realizar en el laboratorio(tareas, investigaciones, prácticas, proyectos, evaluaciones cortas, examen final, etc.) son dependientes. Consiste en que cada actividad será dependiente de la anterior utilizando como punto de partida la Actividad 0 la cual consiste en enviar un correo electrónico a los auxiliares del curso para registrarse en el laboratorio dando con ello el inicio de la ponderación de punteo.
- **Para tener derecho a su nota final debe de entregar todas las actividades.**
- La forma de entrega de de las actividades será vía dropbox, a cada estudiante se le enviará una carpeta identificada con su número de camé en donde sincronizara sus actividades según la fecha y hora límite de entrega en base al enunciado de cada actividad.
- Para la calificación de las actividades se tomara en cuenta la presentación, calidad y contenido de ellas.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 25. Programa del laboratorio, parte 4

### **EVALUACION:**

La nota de promoción para aprobar el laboratorio es de 61 puntos de 100.

La ponderación para cada actividad de evaluación es la siguiente:

Descripción	Pts.
• Tarea 1	2
• Tarea 2	2
• Tarea 3	2
• Tarea práctica 1	3
• Tarea práctica 2	3
• Tarea práctica 3	5
• Práctica 1	5
• Práctica 2	5
• Práctica 3	5
• Proyecto 1	15
• Proyecto 2	25
• Examen corto 1	20
• Examen final	8
• <b>Total</b>	<b>100</b>

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 26. Programa del laboratorio, parte 5

### **CONTENIDO DE LABORATORIO**

**Módulo introducción a la computación.**

**Módulo introducción a HTML.**

**Módulo de importancia de diseño para el desarrollo del software.**

#### **1. Fundamentos de Programación**

- 1.1 Introducción a Algoritmos.
- 1.2 Diagramas de Flujo.
- 1.3 Introducción a la programación.

#### **2. Programación Estructurada**

- 2.1 Tipos de Variables.
- 2.2 Estructuras de Datos.
- 2.3 Estructuras de Control.
- 2.4 Funciones y Procedimientos.

#### **3. Paradigma de Objetos y UML**

- 3.1 Introducción al Paradigma Orientado a Objetos.
- 3.2 Introducción a UML.
- 3.3 Definiciones.
- 3.4 Simbología.

#### **4. Introducción a Java**

- 4.1 Que es Java.
- 4.2 Versiones y ambiente de Java(JDK, IDE, máquina virtual, etc.).

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.



## Anexo 27. Programa del laboratorio, parte 6

- 4. Introducción a Java**
  - 4.1 Que es Java.
  - 4.2 Versiones y ambiente de Java(JDK, IDE, máquina virtual, etc.).
  - 4.3 Características de Java.
  - 4.4 Tipos de programas en Java.
  - 4.5 Componentes del lenguaje Java(variables, constantes, tipos de datos etc.).
  - 4.6 Estructuras de control.
  - 4.7 Arreglos.
- 5. Programación Orientada a Objetos en Java(POO)**
  - 5.1 Conceptos de la programación orientada a objetos.
  - 5.2 Creación de clases y objetos.
  - 5.3 Manejo de memoria.
  - 5.4 Casteo de datos.
  - 5.5 Constructores.
  - 5.6 Herencia.
  - 5.7 Polimorfismo.
- 6. Clases abstractas, interfaces y paquetes**
  - 6.1 Control de acceso a clases y métodos.
  - 6.2 Clases abstractas y métodos.
  - 6.3 Interfaces.
  - 6.4 Paquetes.
- 7. Manejo de Excepciones**
  - 7.1 Comprensión de excepciones y el manejo de estas.
  - 7.2 Sentencia Try Catch.
  - 7.3 Try anidados.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 28. Programa del laboratorio, parte 7

- 8. Manejo de hilos en Java**
  - 8.1 Creación de hilos.
  - 8.2 Multi-hilo.
  - 8.3 Animación empleando hilos.
- 9. Introducción a AWT y SWING**
  - 9.1 Frames.
  - 9.2 Botones.
  - 9.3 Textbox, etc..
  - 9.4 Ejemplos.
- 10. Archivos**
  - 10.1 Introducción a flujo de datos.
  - 10.2 Flujo de entrada y salida de datos.
  - 10.3 Abrir, Cerrar y Modificar Archivos.
  - 10.4 Buffer de lectura y escritura.
- 11. Listas y Colas**
  - 11.1 Pilas usando listas.
  - 11.2 Colas usando listas.
  - 11.3 Listas n-encadenadas.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.

## Anexo 29. Programa del laboratorio, parte 8

### PUNTOS IMPORTANTES A CONSIDERAR:

- Para tener derecho a nota de laboratorio se debe cumplir con el 80% de asistencia a clase de laboratorio a excepción de presentar carta de trabajo membretada.
- Toda entrega tarde de actividades se sancionará con el 20% de la nota de la actividad a excepción de la primera vez que se falle.
- El único medio de entrega de actividades será la carpeta compartida de dropbox de cada estudiante.

### BIBLIOGRAFIA:

- JOYANES, L. y ZAHONERO, I. "Programación en Java 2 (algoritmos, estructura de datos y Programación Orientada a Objetos)". España, McGraw-Hill / Interamericana de España, S. A. 2002, PP 725
- BUDD, Timothy. "Introducción a la programación orientada a objetos", EUA, Addison-Wesley, Iberoamericana, S. A. 1994, P. 409
- JOYANES, L. "Programación en Turbo Pascal Versiones 5.5, 6.0, y 7.0", (2da Edición), México, McGraw-Hill / Interamericana de España, S. A. 1995, PP. 914.
- Manuales de Referencia de Java.
- Cualquier otro material(escrito o digital) entregado en clase.

Fuente: programa de Introducción a la Programación y Computación 1, Fiusac.