



Nombre del Cuso: Introducción a la Programación y Computación II			
<b>Código:</b>	771	<b>Créditos:</b>	5
<b>Escuela:</b>	CIENCIAS Y SISTEMAS	<b>Área a la que pertenece:</b>	Programación
<b>Pre requisito:</b>	Introducción a la Programación y Computación I (770) Matemática Intermedia (107) Lógica Matemática (795) Matemática de Computo 1 (960)	<b>Post requisito:</b>	Organización Computacional (964) Estructura de Datos (772) Org. Lenguajes y Compiladores 1 (777)
<b>Categoría:</b>	Obligatorio	<b>Semestre:</b>	2do. Semestre 2019
<b>Docente:</b>	MsC Ing. Estuardo Zapeta	<b>Auxiliar:</b>	Alba Chinchilla
<b>Edificio:</b>	T3	<b>Sección:</b>	E
<b>Salón del curso:</b>	Jueves (313, T-3) y viernes (314, T-3)	<b>Salón de laboratorio:</b>	-
<b>Horas por semana del curso:</b>	4	<b>Horas por semana del laboratorio:</b>	2
<b>Días que se imparte el curso:</b>	Jueves y viernes	<b>Días que se imparte el laboratorio:</b>	-
<b>Horario del curso:</b>	07:10 - 08:50	<b>Horario del laboratorio:</b>	-

### 1. Descripción del curso

Este curso está diseñado para que el estudiante inicie el proceso de modelado de sistemas de software utilizando los conceptos de la programación orientada a objetos y los diagramas que el lenguaje unificado de datos proporciona. También se introducirá en los conceptos de bases de datos relacionales y la importancia de la información en el contexto empresarial.

### 2. Objetivos

#### General

Preparar al estudiante para desarrollar aplicaciones de software utilizando un enfoque orientado a objetos.

#### Específicos

1. Modelar problemas de forma estándar y profesional.
2. Entender metodologías de desarrollo ágil para construir aplicaciones de software.
3. Estructurar datos con base al modelado entidad relación.

### 3. Metodología

1. El curso se impartirá a través de clases magistrales dos días por semana, con duración de dos periodos cada día.
2. El laboratorio se impartirá una vez por semana, con duración de dos períodos cada día.
3. Durante el semestre, se asigna un proyecto de desarrollo de software dividido en tres fases, a realizarse de manera individual; así como tareas, ejercicios y pruebas cortas.

### 4. Competencias terminales

Al finalizar el curso el estudiante desarrolla las siguientes competencias:

- Dominio en el modelado relacional.
- Capacidad para implementar modelo bajo el lenguaje de consulta estructurado -SQL-
- Capacidad para aplicar metodologías de desarrollo ágil como Scrum.
- Dominio en el modelado UML.

### 5. Observaciones

1. Es obligatorio acumular el 80% de asistencia antes de cada parcial (de lo contrario no se tendrá derecho a examen).
2. El laboratorio se calificará sobre 100, y será equivalente a 30 puntos de zona.
3. Las prácticas de laboratorio serán 3. (1 pt. c/u)
4. Cada práctica se hará corresponder con uno de los proyectos.
5. El catedrático revisará las notas obtenidas en el curso y el laboratorio. Podrá decidir si es necesaria una segunda revisión a cada fase de los proyectos o práctica y considerar nuevamente la ponderación obtenida en cada fase del proyecto o práctica.
6. Las notas de cada proyecto serán publicadas por el catedrático del curso en el transcurso del semestre, el estudiante tendrá 8 días como máximo para pedir revisión de proyecto.
7. El laboratorio debe aprobarse con 61 puntos.
8. Es obligatorio ganar el laboratorio para tener derecho a evaluación final del curso.
9. No habrá proyecto de retrasada, ni reposición de nota de laboratorio.
10. El curso se aprueba con 61 puntos.



2

0

1

9

S  
E  
M  
E  
S  
T  
R  
E

2

**6. Contenido temático del curso**

Unidad	Tema
<p><b>1. Bases de datos relacionales</b></p>	<p><b>1. Bases de datos relacionales</b></p> <p><b>1.1. Un lenguaje común</b></p> <p>1.1.1. ¿Qué es un dato?</p> <p>1.1.2. ¿Qué es una base de datos?</p> <p>1.1.3. ¿Qué es un modelo de datos?</p> <p><b>1.2. El Modelo Relacional</b></p> <p>1.2.1. Introducción al álgebra relacional</p> <p>1.2.2. Diagrama Entidad Relación</p> <p>1.2.2.1. ¿Qué es una entidad?</p> <p>1.2.2.2. ¿Qué es una tupla?</p> <p>1.2.2.3. Atributos</p> <p>1.2.3. Relaciones básicas</p> <p>1.2.3.1. De uno a uno</p> <p>1.2.3.2. De uno a muchos</p> <p>1.2.3.3. De muchos a muchos</p> <p>1.2.4. Identificadores</p> <p>1.2.4.1. Llaves primarias (simples y compuestas)</p> <p>1.2.4.2. Llaves foráneas</p> <p><b>1.3. Mapeo físico de una base de datos relacional</b></p> <p>1.3.1. Introducción a SQL</p> <p>1.3.1.1. Data Definition Language (DDL)</p> <p>1.3.1.2. Data Manipulation Language (DML)</p> <p>1.3.1.3. Constraints (para PK y FK)</p> <p><b>1.4. Normalización</b></p> <p>1.4.1. Primera forma normal</p> <p>1.4.2. Segunda forma normal</p> <p>1.4.3. Tercera forma normal</p> <p><b>1.5. Manejadores de base de Datos</b></p> <p><b>1.6. Sentencias básicas</b></p> <p>1.6.1. CREATE</p> <p>1.6.2. INSERT</p> <p>1.6.3. SELECT / WHERE / ORDER BY / GROUP BY</p> <p>1.6.4. UPDATE</p> <p>1.6.5. JOIN</p>



<p><b>2. Metodologías para el desarrollo de sistemas de información</b></p>	<p><b>2. Metodologías para el desarrollo de sistemas de información</b></p> <p><b>2.1. ¿Qué es un sistema de información?</b></p> <p><b>2.2. Arquitectura de un sistema de información</b></p> <p>2.2.1. Presentación</p> <p>2.2.2. Lógica del negocio</p> <p>2.2.3. Datos</p> <p><b>2.3. Modelado de sistemas</b></p> <p>2.3.1. Abstracción</p> <p>2.3.2. Conceptualización</p> <p>2.3.3. Simbolización</p> <p><b>2.4. Metodología para desarrollar sistemas de información</b></p> <p><b>2.5. ¿Cómo se construye un sistema de información?</b></p> <p><b>2.6. ¿Cuál es el ciclo de vida para el desarrollo de sistemas de información?</b></p> <p>2.6.1. Toma de requerimientos</p> <p>2.6.2. Análisis y Diseño</p> <p>2.6.3. Desarrollo del software</p> <p>2.6.4. Implementación</p> <p>2.6.5. Pruebas</p> <p>2.6.6. Documentación</p> <p><b>2.7. Metodología de desarrollo ágil</b></p> <p>2.7.1. Scrum</p> <p>2.7.1.1. ¿Qué es Scrum?</p> <p>2.7.1.2. Roles</p> <p>2.7.1.3. Artefactos</p> <p>2.7.1.3.1. Producto Backlog</p> <p>2.7.1.3.2. Sprint Backlog</p> <p>2.7.1.3.3. Incremento</p> <p>2.7.2. Historias de usuario</p> <p>2.7.3. Eventos</p> <p>2.7.3.1. Planificación de Sprint</p> <p>2.7.3.2. Sprint</p> <p>2.7.3.3. Daily Scrum</p> <p>2.7.3.4. Revisión del Sprint</p> <p>2.7.3.5. Retrospectiva</p>
<p><b>3. Lenguaje de Modelado Unificado (UML)</b></p>	<p><b>3. Lenguaje de Modelado Unificado (UML)</b></p> <p><b>3.1. Lenguaje de modelado</b></p> <p><b>3.2. Introducción a UML</b></p> <p><b>3.3. Herramientas de modelado</b></p> <p><b>3.4. Modelado de Diagrama de Casos de Uso</b></p> <p>3.4.1. ¿Qué es un requisito?</p> <p>3.4.2. Tipos de requisito</p> <p>3.4.2.1. Funcionales</p> <p>3.4.2.2. No Funcionales</p> <p>3.4.3. Casos de Uso</p> <p>3.4.3.1. ¿Qué es un caso de uso?</p>



- 3.4.3.2. Actores
- 3.4.3.3. Tipos de Relación
  - 3.4.3.3.1. Extiende
  - 3.4.3.3.2. Incluye
  - 3.4.3.3.3. Generaliza
- 3.4.3.4. Diagrama de casos de uso
- 3.4.3.5. Especificación de alto nivel
- 3.4.3.6. Especificación expandida
- 3.5. Modelado de Diagrama de Actividades**
  - 3.5.1. Acciones
  - 3.5.2. Actividades
  - 3.5.3. Transiciones
  - 3.5.4. División y unión
  - 3.5.5. Calles/Transiciones
- 3.6. Modelado de Diagrama de Clases**
  - 3.6.1. Clases
  - 3.6.2. Notación de clases
  - 3.6.3. Atributos
  - 3.6.4. Métodos
  - 3.6.5. Tipos de visibilidad (Atributos y Métodos)
  - 3.6.6. Relaciones de Clase
    - 3.6.6.1. Asociación
    - 3.6.6.2. Realización
    - 3.6.6.3. Agregación
    - 3.6.6.4. Composición
- 3.7. Modelado de Diagrama de Secuencia**
  - 3.7.1. Línea de vida
  - 3.7.2. Mensajes
  - 3.7.3. Ejecución
  - 3.7.4. Mensajes Self
  - 3.7.5. Destrucción de un objeto
  - 3.7.6. Comportamiento del sistema
  - 3.7.7. Eventos y operaciones de un sistema
- 3.8. Modelado de Diagrama de Componentes**
  - 3.8.1. Representando Componentes
  - 3.8.2. Interfaces requeridas
  - 3.8.3. Tipos de componentes
  - 3.8.4. Usos comunes
    - 3.8.4.1. Modelado de código fuente
    - 3.8.4.2. Modelado de una versión ejecutable y bibliotecas
    - 3.8.4.3. Modelado de una base de datos física



2

0

1

9

S  
E  
M  
E  
S  
T  
R  
E

2

## 7. Evaluación del rendimiento académico

Según el Reglamento General de Evaluación y Promoción del Estudiante de la Universidad de San Carlos de Guatemala, la zona tiene valor de 75 puntos, la nota mínima de promoción es de 61 puntos y la zona mínima para optar a examen final es de 36 puntos.

Procedimiento de evaluación		Ponderación
Clase	Tareas y/o cortos	06 pts.
	Primer parcial	13 pts.
	Segundo parcial	13 pts.
	Tercer parcial	13 pts.
<b>Total de clase</b>		<b>45 pts.</b>
Laboratorio	Practicas	03 pts.
	Proyecto Fase I	05 pts.
	Proyecto Fase II	10 pts.
	Proyecto Fase III	12 pts.
<b>Total de laboratorio</b>		<b>30 pts.</b>
Zona		75 pts.
Examen Final		25 pts.
Nota de promoción		100 pts.



2

0

1

9

SEMESTRE

2

### 8. Cronograma de actividades

Tema principal	Contenido a desarrollar	Fecha
Bases de datos relacionales	Un lenguaje común	18-julio
El modelo relacional	- Introducción al álgebra relacional - Diagrama entidad relación	19-julio
El modelo relacional	- Relaciones básicas - Identificadores	25-julio
El modelo relacional	Ejemplos prácticos del modelado entidad relación	26-julio
Mapeo físico de una base de datos relacional	Introducción a SQL	1-agosto
Normalización	- Primera forma normal - Segunda forma normal - Tercera forma normal	2-agosto
Manejadores de bases de datos	- Conociendo los distintos motores de bases de datos - Sentencias básicas de SQL	8-agosto
Sentencias básicas SQL	Aplicaciones prácticas de SQL	9-agosto
<b>Primer parcial</b>		<b>16-agosto</b>
Metodologías para el desarrollo de sistemas de información	- ¿Qué es un sistema de información? - Arquitectura de un sistema de información - Modelado de sistemas	22-agosto
Metodologías para el desarrollo de sistemas de información	- Metodología para desarrollar sistemas de información - ¿Cómo se construye un sistema de información? - ¿Cuál es el ciclo de vida para el desarrollo de sistemas de información?	23-agosto
Metodología de desarrollo ágil	- Scrum - Historias de usuario - Eventos	29-agosto
Lenguaje de Modelado Unificado	- Lenguaje de modelado - Introducción a UML - Herramientas de modelado	30-agosto
Modelado de Diagrama de Casos de Uso	- ¿Qué es un requisito? - Tipos de requisito - Casos de uso	05-septiembre
Modelado de Diagrama de Casos de Uso	Aplicaciones prácticas de diagramas de casos de uso	06-septiembre
Modelado de Diagrama de Casos de Uso	Aplicaciones prácticas de diagramas de casos de uso	12-septiembre
Modelado de Diagrama de Actividades	- Acciones - Actividades - Transiciones - División y unión - Calles/Transiciones	13-septiembre



Modelado de Diagrama de Actividades	Aplicaciones prácticas de diagramas de actividades	19-septiembre
<b>Segundo parcial</b>		<b>20-septiembre</b>
Modelado de Diagrama de Clases	- Clases - Notación de clases - Atributos - Métodos	03-octubre
Modelado de Diagrama de Clases	- Tipos de visibilidad - Relaciones de clase	04-octubre
Modelado de Diagrama de Clases	Aplicaciones prácticas de diagramas de clases	10-octubre
Modelado de Diagrama de Secuencias	- Línea de vida - Mensajes - Ejecución - Mensajes Self - Destrucción de un objeto - Comportamiento del sistema - Eventos y operaciones de un sistema	11-octubre
Modelado de Diagrama de Secuencias	Aplicaciones prácticas de diagramas de secuencias	17-octubre
Modelado de Diagrama de Componentes	- Representando Componentes - Interfaces requeridas - Tipos de componentes - Usos comunes	18-octubre
Modelado de Diagrama de Componentes	Aplicaciones prácticas de diagramas de componentes	24-octubre
<b>Tercer parcial</b>		<b>25-octubre</b>

## 9. Bibliografía

- Jiménez, C. Fundamentos de SQL. 3ra. Edición. McGraw Hill. 2009.
- Scrum Manager. Enlace [<http://www.scrummanager.net/ok>]. 2009.
- Craig Larman, UML y Patrones, Introducción al análisis y diseño orientado a objetos. Prentice Hall.