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

NOMBRE DEL CURSO: Introducción a la Computación y Programación 2

CODIGO:	**771**	**CREDITOS:**	**5**
ESCUELA:	**Ciencias y Sistemas**	**AREA A LA QUE PERTENECE:**	**Programación**
PRE REQUISITO:	**Intr. a la Progr. y Computación 2 (771)** **Matemática Intermedia (107)** **Lógica Matemática (795)** **Matemática de Computo 1 (960)**	**POST REQUISITO:**	**Organización Computacional (964)** **Estructura de Datos (772)** **Org. Lenguajes y Compiladores 1 (777)**
CATEGORIA:	**Obligatorio**	**SEMESTRE:**	**1er. Sem 2015**
CATEDRÁTICO (A):	**Ing. Lenin Rodríguez**	**AUXILIAR:**	**Romulo Martinez**
EDIFICIO:	**T-7**	**SECCIÓN:**	**E**
SALON DEL CURSO:	**103**	**SALON DEL LABORATORIO:**	**T-5 Salón 305**
HORAS POR SEMANA DEL CURSO:	**4**	**HORAS POR SEMANA DEL LABORATORIO:**	**2**
DÍAS QUE SE IMPARTE EL CURSO:	**Jueves y viernes**	**DIAS QUE SE IMPARTE EL LABORATORIO:**	**Sábado**
HORARIO DEL CURSO:	**7:10 – 8:50**	**HORARIO DEL LABORATORIO:**	**9:10 – 10:50**

DESCRIPCIÓN DEL CURSO:

Este curso está diseñado para que el estudiante inicie el proceso de modelaje de sistemas de software utilizando los conceptos de la programación orientada a objetos y los diagramas que el lenguaje unificado de datos proporciona.

OBJETIVOS GENERALES:

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

Objetivos Específicos:

1. Que el estudiante modele los problemas de una forma estándar y profesional.
2. Que el estudiante logre un mayor proceso de abstracción en los problemas que resuelva.
3. Que el estudiante utilice una metodología para desarrollar aplicaciones de software.

METODOLOGIA:
1. El curso se impartirá a través de clases magistrales de 4 períodos semanales impartidos dos días por semana. Cada día 2 periodos.

2. El laboratorio se impartirá una vez por semana, con duración de 2 períodos cada día.

3. Durante el semestre, se asignarán 3 proyectos de programación, a realizarse de manera individual; así como tareas, ejercicios e investigaciones.

EVALUACIÓN DEL RENDIMIENTO ACADEMICO: 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.
De acuerdo con el Normativo de Evaluación y Promoción del estudiante de pregrado de la Facultad de Ingenería, se procederá así:

Procedimiento	**Instrumento de Evaluación**			**Ponderación**
Asignación por tema	3 parciales	1ro.	08pts.	
		2do.	16pts.	
		3ro.	16pts.	40pts.
	Tareas y Examenes Cortos de Clase			05pts.
	Laboratorio (3 proyectos y 3 prácticas)			
		1ro.	06pts.	
		2do.	09pts.	
		3er.	15pts.	30pts
Evaluación Final				25pts.
Nota de Final				100pts.

1er. Parcial – Sábado 14/febrero/2015 Unidad 1.
2do. Parcial – Viernes 13/marzo/2015 Unidad 2 y 3 (Hasta Casos de Uso).
3er. Parcial – Jueves 30/abril /2015 Unidad 3 (desde modelo conceptual) y Unidad 4

Primer Proyecto: 19/Febrero al 06/Marzo
Segundo Proyecto: 06/Marzo al 10/Abril
Tercer Proyecto: 10/Abril al 02/Mayo

Observaciones:

- **Es obligatorio acumular el 80% de asistencia antes de cada parcial (de lo contrario no se tendrá derecho a examen)**
- El laboratorio se calificará sobre 100, y será equivalente a 30 puntos de zona.
- Las prácticas de laboratorio serán 3, cada una con un valor de 1 punto. Los 3 puntos serán parte de la nota de tareas del curso. Cada práctica se hará corresponder con uno de los proyectos.
- El laboratorio debe aprobarse con 61 puntos.
- Es obligatorio ganar el laboratorio para tener derecho a evaluación final del curso.
- No habrá proyecto de retrasada, ni reposición de nota de laboratorio.
- Las notas de cada proyecto serán publicadas por el auxiliar en el transcurso del semestre, el estudiante tendrá 8 días como máximo para pedir revisión de proyecto.
- El curso se aprueba con 61 puntos.

CONTENIDO PROGRAMATICO Y CALENDARIZACIÓN:
Las unidades que el curso comprende, se dividen en cinco, siendo ellas:

Primera Unidad

Introducción a la Base de Datos Relacionales

1.1 Conceptos básicos
- **1.1.1** Base de Datos
- **1.1.2** Sistema de base de datos (DBMS)
- **1.1.3** Base de Datos Relacional
- **1.1.4** Usuarios y esquemas
- **1.1.5** Entidad – tupla – atributos
- **1.1.6** Diagrama Entidad Relación
- **1.1.7** Llaves primarias
- **1.1.8** Llaves foráneas
- **1.1.9** Relaciones básicas
  - **1.1.9.1** De uno a uno
  - **1.1.9.2** De uno a muchos
  - **1.1.9.3** De muchos a muchos
- **1.1.10** Mapeo físico de una DB relacional
  - **1.1.10.1** Introducción a SQL (DDL y DML)
  - **1.1.10.2** Constraints (para PK y FK)

29/Enero – Asueto Ingeniería
30/Enero – Asueto Ingeniería
3 días de clase 05/Febrero al 12/Febrero

Segunda Unidad

Métodología para desarrollo de software

2.1 Conceptos generales
- **2.1.1** Métodos de programación

2.2 Introducción al proceso de desarrollo de sistemas
2.3 Introducción al UML
- **2.3.1** Objetivos
- **2.3.2** Vistas
- **2.3.3** Diagramas

2.4 Análisis y Diseño Orientado a Objetos
2.5 Proceso de Desarrollo de SW, método iterativo-incremental
- **1.5.1** Planeación y elaboración
- **1.5.2** Ciclos de Construcción

2.6 Requerimientos
2.7 Mejores prácticas para el desarrollo de software
2.8 Capas arquitectónicas para el desarrollo de aplicaciones de software
2.9 Relación entre UML y el ciclo de desarrollo
2.10 Proceso de Desarrollo de SW, método iterativo incremental

3 días de clase 19/febrero – 26/febrero

Tercera Unidad

Etapa de análisis del ciclo de construcción

3.1 Casos de uso
- **3.1.1** Introducción
- **3.1.2** Casos de Uso de alto nivel
- **3.1.3** Casos de Uso expandidos
- **3.1.4** Tipos de caso de uso
  - **3.1.4.1** Primarios, secundarios
  - **3.1.4.2** Esencial, Real
- **3.1.5** Actores
- **3.1.6** Identificación de casos de uso
- **3.1.7** Diagrama de casos de uso
- **3.1.8** Los sistemas y sus fronteras
- **3.1.9** Relaciones en un diagrama de casos de uso
  - **3.1.9.1** Comunica
  - **3.1.9.2** Incluir
  - **3.1.9.3** Extiende

3.1.9.4 Generalización
3.1.10 Ejemplos

3 días de clase 27/febrero – 06/marzo

3.2 Modelo conceptual – diagrama de estructura estática
3.2.1 Conceptos generales y notación
3.2.1.1 Clase – atributos
3.2.1.2 Objeto (instancia)
3.2.1.3 Relaciones entre clases
3.2.1.3.1 Asociaciones
3.2.1.3.1.1 Nombre y dirección de la asociación
3.2.1.3.1.2 Multiplicidad
3.2.1.3.1.3 Roles
3.2.1.3.1.4 Agregación
3.2.1.3.1.5 Clases asociación
3.2.1.3.1.6 Asociación N-aria
3.2.1.3.1.7 Navegabilidad
3.2.1.3.1.8 Herencia
3.2.1.3.2 Clase paramétrica
3.2.1.3.3 Paquete
3.2.1.3.4 Dependencia
3.2.2 Construcción de un modelo conceptual
3.2.3 Modelos conceptuales
3.2.4 Estrategias para identificar objetos
3.2.5 Directrices para construir el modelo conceptual
3.2.5.1 Especificación o descripción de conceptos
3.2.5.2 Agregación de las asociaciones
3.2.5.3 Identificación de asociaciones
3.2.5.4 Asignación de atributos

4 días de clase 06/marzo – 20/marzo

3.3 Definición de glosario
3.3.1 Introducción
3.3.2 Reglas y restricciones del dominio
3.3.3 Ejemplos
3.4 Comportamiento inicial del sistema
3.4.1 Introducción a diagramas de secuencia
3.4.1.1 Notación UML
3.4.1.2 Caminos alternativos de ejecución y concurrencia
3.4.1.3 Destrucción de un objeto
3.4.1.4 Métodos recursivos
3.4.1.5 Comportamiento del sistema
3.4.1.6 Eventos y operaciones de un sistema

3.4.1.7 Registro de las operaciones de un sistema
3.4.1.8 Elaboración de un diagrama de secuencia
3.4.1.9 Eventos y fronteras de un sistema
3.4.1.10 Asignación de nombres a los eventos y operaciones de un sistema

3.4.2 Introducción a diagramas de estado
3.4.1.1 Conceptos elementales y notación de un diagrama de estado
3.4.1.1.1 Estado
3.4.1.1.2 Eventos
3.4.1.1.3 Envío de mensajes
3.4.1.2 Utilidad de los diagramas de estado para los casos de uso
3.4.1.3 Diagramas de estado del sistema
3.4.1.4 Tipos independientes y dependientes del estado
3.4.1.5 Tipos y clases comunes dependientes del estado
3.4.1.6 Tipos de eventos
3.4.1.7 Notación complementaria de los diagramas de estado
3.4.1.7.1 Acciones de transición
3.4.1.7.2 Condiciones protectoras de las transiciones
3.4.1.7.3 Estados anidados

3.5 Introducción a los diagramas de actividades
3.5.1 Introducción
3.5.2 Estado de actividad y estados de acción
3.5.3 Transiciones
3.5.4 Bifurcaciones
3.5.5 División y unión
3.5.6 Calles

23/marzo al 28/marzo – receso estudiantil
30/marzo al 4/abril – semana mayor
3 días de clase 9/abril – 16/abril

Cuarta Unidad

Etapa de diseño del ciclo de construcción

4.1 Definición de reportes, interfaz de usuario y secuencia de pantallas
4.2 Diagrama de clases del diseño
4.2.1 Introducción
4.2.2 Relaciones de dependencia para representar visibilidad entre clases
4.2.3 Tipos de visibilidad (parámetro, local, global)
4.2.4 Clase controladora
4.2.5 Elaboración de un diagrama de clases del diseño
4.2.6 Navegabilidad
4.2.7 Visibilidad de atributos y métodos

4.2.8 Comparación entre el modelo conceptual y los diagramas de clases del diseño

4.3 Introducción a los diagramas de colaboración

- **4.3.1** Actividades y dependencias
- **4.3.2** Elaboración de diagramas de colaboración
- **4.3.3** Relación entre artefactos
- **4.3.4** Notación básica de los diagramas de colaboración
  - **4.3.4.1** Representación gráfica de clases e instancias
  - **4.3.4.2** Representación gráfica de los vínculos
  - **4.3.4.3** Representación gráfica de los mensajes
  - **4.3.4.4** Representación gráfica de los parámetros
  - **4.3.4.5** Representación gráfica del mensaje para devolver valor
  - **4.3.4.6** Sintaxis de los mensajes
  - **4.3.4.7** Representación gráfica de los mensajes al “emisor” o a “esto”
  - **4.3.4.8** Representación gráfica de la iteración
  - **4.3.4.9** Representación gráfica de la creación de instancias
  - **4.3.4.10** Representación gráfica de la secuencia del número de mensaje
  - **4.3.4.11** Representación gráfica de los mensajes condicionales
  - **4.3.4.12** Representación gráfica de trayectorias condicionales mutuamente excluyentes
  - **4.3.4.13** Representación gráfica de las colecciones
  - **4.3.4.14** Representación gráfica de los mensajes dirigidos a multiobjetos
  - **4.3.4.15** Representación gráfica de los mensajes dirigidos a una clase

3 días de clase 17/Abril – 24/Abril

Quinta Unidad

<u>Modelado físico de un sistema orientado a objetos</u>

5.1 Componentes
5.2 Interfaces
5.3 Tipos de componentes
5.4 Organización de componentes
5.5 Estereotipos de componentes
5.6 Despliegue – nodos
5.7 Nodos y componentes
5.8 Diagramas de componentes

- **5.8.1** Introducción
- **5.8.2** Usos comunes
  - **5.8.2.1** Modelado de código fuente
  - **5.8.2.2** Modelado de una versión ejecutable y bibliotecas
  - **5.8.2.3** Modelado de una base de datos física

5.9 Diagramas de despliegue
- **5.9.1** Modelado de sistema empotrado
- **5.9.2** Modelado de sistema cliente/servidor

5.10 Arquitectura del sistema
- **5.10.1** Arquitectura de tres niveles (capas)
- **5.10.2** Arquitectura de tres niveles orientada a objetos
- **5.10.3** Arquitectura multinivel

5.11 Paquetes

1 de mayo, asueto nacional
2 días de clase 07/mayo – 08/mayo

BIBLIOGRAFÍA:

Craig Larman, UML y Patrones, Introducción al análisis y diseño orientado a objetos. Prentice Hall.

LISTA DE CATEDRÁTICOS

CURSO	SEC	EDI	SALON	INICIO	FIN	L	M	MI	J	V	CATEDRATICOS
Introducción a la Programación y Computación 2	A	T-3	216	07:10	08:50				X	X	Ing. Miguel Ángel Cancinos
Introducción a la Programación y Computación 2	B	T-3	215	07:10	08:50				X	X	Ing. Claudia Rojas Morales
Introducción a la Programación y Computación 2	C	T-3	213	07:10	08:50				X	X	Ing. José Manuel Ruíz
Introducción a la Programación y Computación 2	D	T-3	312	07:10	08:50				X	X	Ing. William Escobar
Introducción a la Programación y Computación 2	E	T-7	103	07:10	08:50				X	X	Ing. Lenin Rodríguez