



PROGRAMA CURSO: PROGRAMACIÓN II
I Semestre, 2013

Datos Generales

Sigla: IF-3000

Nombre del curso: Programación II

Tipo de curso: Teórico-Práctico

Número de créditos: 4

Número de horas semanales presenciales: 8

Número de horas semanales de trabajo independiente del estudiante: mínimo 6

Requisitos: IF-2000 Programación I

Correquisitos: IF3001 Algoritmos y Estructuras de Datos

Ubicación en el plan de estudio: III Ciclo

Horario del curso: K 8:00 a 11:50 y J 8:00 a 11:50

Suficiencia:

Tutoría:

Datos del Profesor

Nombre: Mci. Wendy María Ramírez González

Correo Electrónico: wendy.ramirez@ucurso.info

Horario de Consulta: M 13:00 a 16:50

1. Descripción del curso

Este curso profundiza en el diseño y desarrollo de programas computacionales haciendo uso de un lenguaje de programación. Se realiza un estudio exhaustivo del paradigma de programación orientado a objetos y se introduce al estudiante en el análisis y diseño orientado a objetos empleando el lenguaje de modelado UML y los patrones de diseño. Además, se estudian algunas interfaces de programación de aplicaciones (API) y se introduce al estudiante en el uso de hilos, sockets y en la construcción de aplicaciones basadas en el Web.

2. Objetivo General

Desarrollar en el estudiante la capacidad de construcción de aplicaciones avanzadas, profundizando en el estudio del paradigma de programación orientada a objetos.



3. Objetivos específicos

Al finalizar el curso el o la estudiante estará en capacidad de:

- a. Aplicar los conceptos de la orientación a objetos en la construcción de aplicaciones de software.
- b. Desarrollar aplicaciones cuya implementación y estructura esté separada en capas lógicas.
- c. Formular e implementar modelos de clases basados en el lenguaje de modelado UML.
- d. Crear aplicaciones utilizando colecciones.
- e. Realizar aplicaciones manejando las tecnologías de hilos y sockets.
- f. Construir aplicaciones Web básicas.

4. Contenidos

1. Modelado de sistemas orientados a objetos con UML
 - 1.1. El lenguaje de modelado unificado (UML), importancia y utilización.
 - 1.2. Diagramas de Clase: Identificación de Clases (atributos, operaciones), herencia e interfaces.
 - 1.3. Relaciones de clases: dependencia, generalización, asociación (nombre, rol, multiplicidad, navegación, agregación y composición).
2. Colecciones
 - 2.1. Concepto y utilización de las colecciones.
 - 2.2. Utilización de la interfaz Collection
 - 2.3. Concepto y aplicación de la clase ArrayList
 - 2.4. Concepto y aplicación de la clase Iterator y el patrón de diseño Iterator (Iterador)
 - 2.5. Utilización de los métodos para manipular colecciones
3. Patrones de diseño
 - 3.1. Definición de patrones de diseño, características, importancia y utilización.
 - 3.2. Patrones de creación: concepto y utilización
 - 3.2.1. Abstract Factory (Fabrica abstracta)
 - 3.2.2. Builder (Constructor)
 - 3.2.3. Prototype (Prototipo)
 - 3.2.4. Singleton (Único)
 - 3.3. Patrones de comportamiento: concepto y utilización
 - 3.3.1. Memento (Recuerdo)
 - 3.3.2. Strategy (Estrategia)
 - 3.4. Patrones estructurales: concepto y utilización
 - 3.4.1. Composite (Compuesto)
 - 3.4.2. Decorator (Decorador)
 - 3.4.3. Facade (Fachada)
 - 3.5. Patrones de sistema: concepto y utilización
 - 3.5.1. Modelo Vista Controlador



4. Hilos
 - 4.1. Concepto y características de hilos
 - 4.2. Creación, administración y destrucción de hilos
 - 4.3. Múltiples hilos
 - 4.4. Los hilos en las aplicaciones Cliente-Servidor
 - 4.5. Sincronización

5. Sockets
 - 5.1. Concepto y características de sockets
 - 5.2. Conexión de aplicaciones mediante sockets
 - 5.3. Sockets en aplicaciones Cliente - Servidor

6. Archivos especializados (XML)
 - 6.1. Definición de archivo XML, características y utilización
 - 6.2. Creación, lectura y escritura
 - 6.3. Parseo de sintaxis con archivo XML

7. Introducción al desarrollo de aplicaciones web
 - 7.1. Introducción a las aplicaciones web (definición, características)
 - 7.2. Historia de la Web
 - 7.3. Arquitectura básica web (cliente-servidor)
 - 7.4. XHTML
 - 7.4.1. Estructura de documentos XHTML
 - 7.4.2. Doctype, html, head, body
 - 7.4.3. Imágenes, listas, tablas, formularios, otros
 - 7.4.4. Estándares y Validación W3C
 - 7.5. Lenguajes del lado del servidor

5. Metodología

El curso presenta un eje de desarrollo teórico-práctico. El profesor desarrolla clases magistrales y realiza asignaciones (quices, tareas cortas, entre otros) para que los estudiantes apliquen los conceptos discutidos. Se realizan laboratorios durante el curso. Los estudiantes desarrollan proyectos programados donde ponen en práctica y amplían los conocimientos adquiridos en el curso. Además, se asignarán temas de investigación a los estudiantes.



6. Evaluación

Descripción	Porcentaje
Examen Parcial I	15%
Examen Parcial II	15%
Examen Parcial III	15%
Quices, Tareas Cortas , otros	5%
Laboratorios	10%
Tarea Programada 1	15%
Tarea Programada 2	15%
Trabajo de investigación	10%
Total:	100%

Consideraciones sobre la evaluación

- ✓ Los Quices se harán sin previo aviso en cualquier momento de la lección y no se harán reposiciones (excepto por las disposiciones establecidas en el reglamento de Régimen Académico Estudiantil).
- ✓ Las fechas de las evaluaciones del cronograma están sujetas a cambio dependiendo del avance de los temas vistos.
- ✓ Los rubros de calificación de cada una de las evaluaciones serán entregadas junto con la especificación de la misma.
- ✓ Las tareas programadas deben ser entregadas a la hora y fecha indicadas en el enunciado del proyecto. En caso de que se atrase un día, perderán el 40% del valor del trabajo.
- ✓ Para los demás aspectos de evaluación no se aceptarán entregas después de la fecha y hora solicitada (excepto por las disposiciones establecidas en el reglamento de Régimen Académico Estudiantil).
- ✓ Los temas de investigación también serán evaluados dentro de rubro los exámenes.
- ✓ Mantener celulares en modo silencioso o apagado durante las lecciones.
- ✓ La aparición de un celular durante un examen o quiz anulará automáticamente el mismo.
- ✓ La comprobación de que alguna tarea, laboratorio, proyecto, examen u otra evaluación es una copia implicará un 0 en la nota de los involucrados y hará que se apliquen las sanciones que contemplan los reglamentos universitarios.
- ✓ Se utilizará la plataforma de Mediación Virtual como apoyo del curso.



7. Cronograma.

	SEMANA	TEMA
1	11 – 17 Marzo	Presentación curso y entrega carta al estudiante Tema 1: Modelado de sistemas orientados a objetos con UML
2	18– 24 Marzo	Tema 1: Modelado de sistemas orientados a objetos con UML Tema 2: Colecciones
3	25 – 31 Marzo	Semana Santa
4	01 – 07 Abril	Tema 2: Colecciones Tema 3: Patrones de diseño
5	08 – 14 Abril	Tema 3: Patrones de diseño 11 de abril feriado
6	15 – 21 Abril	Tema 3: Patrones de diseño I Examen
7	22 – 28 Abril	Semana Universitaria Tema 4: Hilos
8	29 – 05 Mayo	Tema 4: Hilos 01 de mayo feriado
9	06 – 12 Mayo	Tema 4: Hilos Entrega I Tarea Programada
10	13 – 19 Mayo	Tema 5: Sockets Comprobación I Tarea Programada
11	20 – 26 Mayo	Tema 5: Sockets
12	27 – 02 Junio	Tema 6: Archivos especializados (XML) II Examen
13	03 – 09 Junio	Tema 6: Archivos especializados (XML)
14	10 – 16 Junio	Tema 7: Introducción al desarrollo de aplicaciones web
15	17 – 23 Junio	Tema 7: Introducción al desarrollo de aplicaciones web
16	24 – 30 Junio	Tema 7: Introducción al desarrollo de aplicaciones web
17	01 – 07 Junio	III Examen
18	08 – 14 Julio	Entrega II Tarea Programada Comprobación II Tarea Programada Entrega de promedios
19	15 – 21 Julio	Ampliación



8. Bibliografía

- Booch, G., Jacobson, I., & Rumbaugh, J. (2006). *El lenguaje de modelado unificado, UML*. Pearson Education.
- Fowler, M. (1997). *UML Gota a Gota*. México: Pearson Education.
- Whitten, J., Bentley, L., Randolph, G., Rico, M., & Orozco, M. (2008). *Análisis de sistemas: diseño y métodos*. México: McGraw-Hill.
- Schach, S. (2005). *Análisis y Diseño Orientado a Objetos con UML y el proceso unificado*. México: Mc Graw Hill.
- Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. México: Pearson Education.
- Stelting, S., & Maassen, O. (2003). *Patrones de diseño aplicados a Java*. Madrid: Pearson Education.
- Larman, C. (2003). *UML y Patrones*. Madrid: Pearson Education.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2003). *Patrones de Diseño. Elementos de software orientado a objetos reusable*. Madrid: Pearson Education.
- Christiansson, B., Forss, M., Hagen, I., Hansson, K., Jonasson, J., Jonasson, M., y otros. (2008). *GoF Design Patterns - with examples using Java and UML2*. Creative Commons Attribution - ShareAlike 3.0 License.
- Deitel, P. J., & Deitel, H. M. (2008). *Cómo programar en Java* (Séptima ed.). México: Pearson Education.
- Deitel, P. J., & Deitel, H. M. (2012). *Cómo programar en Java* (Novena ed.). México: Pearson Education.
- Schifreen, R. (2010). *The Web Book*. Obtenido de <http://www.the-web-book.com/>
- Tutoriales W3School. (2011). Obtenido de <http://www.w3schools.com/>
-