



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: 12

Requisitos: IF-2000 Programación I

Correquisitos: IF3001 Algoritmos y Estructuras de Datos

Ubicación en el plan de estudio: III Ciclo

Horario del curso: L 17:00 a 20:50 y J 17:00 a 20:50

Suficiencia:

Tutoría:

Datos del Profesor

Nombre: Ing. Cristian Moraga Ruiz

Correo Electrónico: cristian.moraga@ucurso.info

Horario de Consulta: M 18:00 a 22:00

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. Asociaciones de clases (uni-direccional, bi-direccional, clase asociación, agregación, reflexiva), multiplicidad y roles.
2. Patrones de diseño
 - 2.1. Definición de patrones de diseño, características, importancia y utilización.
 - 2.2. Patrones de creación: concepto y utilización
 - 2.2.1. Abstract Factory (Fabrica abstracta)
 - 2.2.2. Builder (Constructor)
 - 2.2.3. Prototype (Prototipo)
 - 2.2.4. Singleton (Único)
 - 2.3. Patrones de comportamiento: concepto y utilización
 - 2.3.1. Memento (Recuerdo)
 - 2.3.2. Strategy (Estrategia)
 - 2.3.3. Visitor (Visitante)
 - 2.4. Patrones estructurales: concepto y utilización
 - 2.4.1. Composite (Compuesto)
 - 2.4.2. Decorator (Decorador)
 - 2.4.3. Facade (Fachada)
 - 2.5. Patrones de sistema: concepto y utilización
3. Colecciones
 - 3.1. Concepto y utilización de las colecciones.
 - 3.2. Utilización de la interfaz Collection
 - 3.3. Concepto y aplicación de la clase ArrayList
 - 3.4. Concepto y aplicación de la clase Iterator y el patrón de diseño Iterator (Iterador)



3.5. Utilización de los métodos para manipular colecciones

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
 - 7.4.4. Estándares y Validación W3C

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

<i>Descripción</i>	<i>Porcentaje</i>
Examen Parcial I	15%
Examen Parcial II	20%
Examen Parcial III	15%
Quices, Tareas Cortas , otros	7%
Laboratorios	15%
Tarea Programada 1	10%
Tarea Programada 2	10%
Trabajo de investigación	8%

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 se realizarán en grupos establecidos por el profesor.
- ✓ Las tareas programadas deben ir acompañadas de la respectiva documentación. No se recibirá la misma posterior a la entrega.
- ✓ 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.
- ✓ Para las tareas programadas se realizará una comprobación individual para determinar la participación de los integrantes.
- ✓ El trabajo de investigación incluirá un rubro para exposición del mismo.
- ✓ Para todas las entregas sólo se recibirán aquellas cuyo contenido sea exclusivamente de la asignación respectiva.
- ✓ 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 o examen es una copia hará que se apliquen las sanciones que contemple el reglamento de Régimen Académico Estudiantil. Consultar en: http://cu.ucr.ac.cr/normativ/regimen_academico_estudiantil.pdf.
- ✓ Se utilizará la plataforma de Mediación Virtual como apoyo del curso.



7. Cronograma.

	SEMANA	TEMA
1	05 – 09 Marzo	Presentación curso y entrega carta al estudiante Tema 1: Modelado de sistemas orientados a objetos con UML
2	12 – 16 Marzo	Tema 1: Modelado de sistemas orientados a objetos con UML Tema 2: Patrones de diseño
3	19 – 23 Marzo	Tema 2: Patrones de diseño
4	26 – 30 Marzo	Tema 2: Patrones de diseño Tema 3: Colecciones
5	02 – 06 Abril	Semana Santa
6	09 – 13 Abril	Tema 3: Colecciones I Examen
7	16 – 20 Abril	Tema 4: Hilos Evento de HP DAY (jueves 19 de abril) Evento Microsoft
8	23 – 27 Abril	Tema 4: Hilos Semana Universitaria
9	30 – 04 Mayo	Tema 4: Hilos Entrega I Tarea Programada
10	07 – 11 Mayo	Tema 5: Sockets Comprobación I Tarea Programada
11	14 – 18 Mayo	Tema 5: Sockets
12	21 – 25 Mayo	Tema 6: Archivos especializados (XML) II Examen
13	28 – 01 Junio	Tema 6: Archivos especializados (XML)
14	04 – 08 Junio	Tema 7: Introducción al desarrollo de aplicaciones web
15	11 – 15 Junio	Tema 7: Introducción al desarrollo de aplicaciones web
16	18 – 22 Junio	Entrega y exposición I Trabajo Investigación
17	25 – 29 Junio	III Examen
18	02 – 06 Julio	Entrega II Tarea Programada Comprobación II Tarea Programada Entrega de promedios
19	09 – 13 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.
- Schifreen, R. (2010). *The Web Book*. Obtenido de <http://www.the-web-book.com/>
- Tutoriales W3School*. (2011). Obtenido de <http://www.w3schools.com/>
-