



## 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. Thomas Huertas Sánchez

**Correo Electrónico:** thomhs@gmail.com

**Horario de Consulta:** S 13:00 a 17:00

---

## 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. Principios de diseño: definición, importancia y utilización.
    - 2.1.1. Principio abierto/cerrado
    - 2.1.2. Principio de inversión de dependencias
    - 2.1.3. Principio de segregación de interfaces
    - 2.1.4. Principio de responsabilidad única
    - 2.1.5. Principio de sustitución de Liskov
  - 2.2. Definición de patrones de diseño, características, importancia y utilización.
  - 2.3. Patrones de creación: concepto y utilización
    - 2.3.1. Abstract Factory (Fabrica abstracta)
    - 2.3.2. Builder (Constructor)
    - 2.3.3. Prototype (Prototipo)
    - 2.3.4. Singleton (Único)
  - 2.4. Patrones de comportamiento: concepto y utilización
    - 2.4.1. Observer (Observador)
    - 2.4.2. Strategy (Estrategia)
    - 2.4.3. Visitor (Visitante)
  - 2.5. Patrones estructurales: concepto y utilización
    - 2.5.1. Composite (Compuesto)
    - 2.5.2. Decorator (Decorador)
    - 2.5.3. Facade (Fachada)
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](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	10 – 15 Marzo	Presentación curso y entrega carta al estudiante Tema 1: Modelado de sistemas orientados a objetos con UML
2	17 – 22 Marzo	Tema 1: Modelado de sistemas orientados a objetos con UML Tema 2: Patrones de diseño
3	24 – 29 Marzo	Tema 2: Patrones de diseño
4	31 – 05 Abril	Tema 2: Patrones de diseño Tema 2: Patrones de diseño
5	07 – 12 Abril	Tema 2: Patrones de diseño <b>I Examen</b>
6	14 - 19 Abril	<b>Semana Santa</b>
7	21 - 26 Abril	Tema 3: Colecciones Tema 3: Colecciones <b>Semana Universitaria</b>
8	28 – 03 Mayo	Tema 4: Hilos Tema 4: Hilos
9	05 – 10 Mayo	Tema 5: Sockets <b>Entrega I Tarea Programada</b>
10	12 – 17 Mayo	<b>Comprobación I Tarea Programada</b>
11	19 – 24 Mayo	Tema 5: Sockets Tema 6: Archivos especializados (XML)
12	26 – 31 Mayo	Tema 6: Archivos especializados (XML) <b>II Examen</b>
13	02 - 07 Junio	Tema 7: Introducción al desarrollo de aplicaciones web
14	09 – 14 Junio	Tema 7: Introducción al desarrollo de aplicaciones web
15	16 – 21 Junio	<b>Entrega y exposición I Trabajo Investigación</b>
16	23 – 28 Junio	<b>III Examen</b>
17	30 – 04 Julio	<b>Entrega II Tarea Programada</b>



		<b>Comprobación II Tarea Programada</b> <b>Entrega de promedios</b>
18	07 – 12 Julio	<b>Ampliación</b>

## 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.
- Freeman Eric, Freeman Elisabeth (2004). *Headfirst Design Patterns*. O’really
- 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 reutilizable*. 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/>