



---

**PROGRAMA CURSO: PROGRAMACIÓN II**  
I Semestre, 2014

---

**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:** 8

**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 S 08:00 a 11:50

**Suficiencia:** No

**Tutoría:** No

---

**Datos del Profesor**

**Nombre:** Alvaro Bolaños Campos

**Correo Electrónico:** alvaro.bolanos@ucrso.info

**Horario de Consulta:** L 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. 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. Manejo dinámico de memoria
  - 2.1. Concepto y ejemplos de estructuras de datos dinámicas y punteros.
  - 2.2. Concepto y utilización de las colecciones.
    - 2.2.1. Concepto y aplicación de Collection y Collections, ArrayList, Iterator
    - 2.2.2. Utilización de los métodos para manipular colecciones
3. Patrones de diseño
  - 3.1. Principios de diseño: definición, importancia y utilización.
    - 3.1.1. Principio abierto/cerrado
    - 3.1.2. Principio de inversión de dependencias
    - 3.1.3. Principio de segregación de interfaces
    - 3.1.4. Principio de responsabilidad única
    - 3.1.5. Principio de sustitución de Liskov
  - 3.2. Patrones de diseño: definición, características, importancia y utilización.
  - 3.3. Patrones de creación: concepto y utilización
    - 3.3.1. Abstract Factory (Fabrica abstracta)
    - 3.3.2. Builder (Constructor)
    - 3.3.3. Prototype (Prototipo)
    - 3.3.4. Singleton (Único)
  - 3.4. Patrones de comportamiento: concepto y utilización
    - 3.4.1. Observer (Observador)
    - 3.4.2. Memento (Recuerdo)
    - 3.4.3. Strategy (Estrategia)
  - 3.5. Patrones estructurales: concepto y utilización
    - 3.5.1. Composite (Compuesto)



### 3.5.2. Decorator (Decorador)

4. Pruebas Unitarias
  - 4.1. Definición prueba de software, tipos de pruebas, pruebas unitarias
  - 4.2. Realización de Pruebas Unitarias utilizando JUnit
5. Hilos
  - 5.1. Concepto y características de hilos
  - 5.2. Creación, administración y destrucción de hilos
  - 5.3. Múltiples hilos
  - 5.4. Los hilos en las aplicaciones Cliente-Servidor
  - 5.5. Sincronización
6. Sockets
  - 6.1. Concepto y características de sockets
  - 6.2. Conexión de aplicaciones mediante sockets
  - 6.3. Sockets en aplicaciones Cliente - Servidor
7. Archivos especializados (XML)
  - 7.1. Definición de archivo XML, características y utilización
  - 7.2. Creación, lectura y escritura
  - 7.3. Parseo de sintaxis con archivo XML
8. Introducción al desarrollo de aplicaciones web
  - 8.1. Introducción a las aplicaciones web (definición, características)
  - 8.2. Historia de la Web
  - 8.3. Arquitectura básica web (cliente-servidor)
  - 8.4. Doctype, html, head, body
  - 8.5. Imágenes, listas, tablas, formularios, otros
  - 8.6. Estructura de documentos XHTML
  - 8.7. Estándares y Validación W3C
  - 8.8. 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	20%
Examen Parcial II	20%
Examen Parcial III	10%
Quices, Tareas Cortas , otros	10%
Laboratorios	10%
Tarea Programada 1	10%
Tarea Programada 2	10%
Trabajo de investigación	10%
<b>Total:</b>	<b>100%</b>

### 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(a).
- ✓ Para las tareas programadas se realizará una comprobación individual para determinar la participación de los integrantes, de la cual dependerá el total de la nota.
- ✓ 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(desde un minuto hasta 24 horas exactas), 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).
- ✓ Para cualquier ejercicio programado(independientemente del rubro de evaluación al que pertenezca), es responsabilidad del estudiante entregar una versión de la aplicación que compile sin errores, si el resultado entregado no cumple con esta característica el ejercicio será calificado con nota 0.
- ✓ 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 contempla la reglamentación universitaria.
- ✓ Para comunicación con el profesor(a) se utilizará únicamente el correo de la carrera (@ucurso.info) por lo cual el estudiante debe tenerlo activo y revisarlo continuamente, ya que es el medio que se utilizará para avisarle sobre actualizaciones en el sitio digital del curso, avisos, asignación de tareas, etc.



- ✓ Se utilizará la plataforma de Moodle de Informática Empresarial como apoyo del curso, entrega de material del curso por parte del profesor(a) y para recibo y calificación de algunas de las evaluaciones.

## 7. Cronograma.

	SEMANA	TEMA
1	09 – 15 Marzo	Presentación curso y entrega programa del curso Repaso POO Tema 1: Modelado de sistemas orientados a objetos con UML
2	16– 22 Marzo	Tema 2: Manejo dinámico de memoria Tema 3: Patrones de diseño
3	23 – 29 Marzo	Tema 3: Patrones de diseño
4	30 – 05 Abril	<b>Semana Santa</b>
5	06 – 12 Abril	Tema 3: Patrones de diseño
6	13 – 19 Abril	<b>I Examen</b> Tema 3: Patrones de diseño
7	20 – 26 Abril	Tema 4: Pruebas Unitarias <b>Semana Universitaria</b>
8	27 – 03 Mayo	Tema 4: Pruebas Unitarias <b>Feriado 1 de mayo</b>
9	04 – 10 Mayo	Tema 4: Hilos <b>Entrega y Comprobación I Tarea Programada</b>
1 0	11 – 17 Mayo	Tema 4: Hilos Tema 5: Sockets
1 1	18 – 24 Mayo	Tema 5: Sockets
1 2	25 – 31 Mayo	<b>II Examen</b> Tema 6: Archivos especializados (XML)



1 3	01 – 07 Junio	Tema 7: Archivos especializados (XML)
1 4	08 – 14 Junio	Tema 8: Introducción al desarrollo de aplicaciones web
1 5	15 – 21 Junio	Tema 8 Introducción al desarrollo de aplicaciones web
1 6	22 – 28 Junio	Tema 8: Introducción al desarrollo de aplicaciones web
1 7	29 – 05 Julio	<b>III Examen</b>
1 8	06 – 12 Julio	<b>Entrega y Comprobación II Tarea Programada</b> <b>Entrega de promedios</b>
1 9	13 – 19 Julio	<b>Ampliación</b>

## 9. 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: McGraw Hill.
- Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. México: Pearson Education.
- Freeman Eric, Freeman Elisabeth (2004). *Headfirst Design Patterns*. O´really
- 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/>
-