

**PROGRAMA CURSO: PROGRAMACIÓN II**  
I Semestre, 2017

### Datos Generales

---

**Sigla del curso:** IF3000  
**Nombre del curso:** Programación II  
**Tipo de curso:** Teórico-Práctico  
**Créditos:** 4  
**Horas lectivas semanales:** 8  
**Requisitos:** IF2000 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 V 8:00 a 11:50  
**Tutoría:** NO  
**Suficiencia:** NO  
**Modalidad:** Bajo virtual

### Datos del Profesor

---

**Nombre:** Mci. Wendy María Ramírez González  
**Correo electrónico:** wendymaria.ramirez@ucr.ac.cr, wendy.ramirez@ucrso.info  
**Horas de consulta presencial:** V 13:00 a 17:00  
**Horas de consulta remota:** No  
**Medio oficial para consulta remota:** No

---

### 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.

---

### Objetivos

#### 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:

- Aplicar los conceptos de la orientación a objetos en la construcción de aplicaciones de software.

- Desarrollar aplicaciones cuya implementación y estructura esté separada en capas lógicas.
  - Formular e implementar modelos de clases basados en el lenguaje de modelado UML.
  - 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(plantillas).
    - 2.2.1. Concepto y aplicación de colecciones
    - 2.2.2. Utilización de métodos para manipular colecciones
3. Patrones de diseño
  - 3.1. Principios de diseño: definición, importancia y utilización.
    - 3.1.1. Principios SOLID
  - 3.2. Patrones de diseño: definición, características, importancia y utilización.
  - 3.3. Patrones de creación: concepto y utilización
    - 3.3.1. Factory Method (Método de fábrica)
    - 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)
  - 3.6. Patrones de sistema: concepto y utilización
    - 3.6.1. Model - View – Controller(Modelo-Vista-Controlador, MVC)
4. Pruebas Unitarias
  - 4.1. Definición prueba de software, tipos de pruebas, pruebas unitarias
  - 4.2. Utilización de un framework de pruebas

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, historia)
  - 8.2. Arquitectura básica web (cliente-servidor)
  - 8.3. HTML: Doctype, head, body, imágenes, listas, tablas, formularios, otros
  - 8.4. Estructura de documentos XHTML
  - 8.5. Estándares y Validación W3C
  - 8.6. Lenguajes y tecnologías 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

<i>Descripción</i>	<i>Porcentaje</i>
Examen Parcial I	22%
Examen Parcial II	22%
Quices, Tareas Cortas, otros	8%
Laboratorios	10%
Proyecto Programado I	15%
Proyecto Programado II	18%
Trabajo de Investigación	5%
<b>Total:</b>	<b>100%</b>

### Consideraciones sobre la evaluación

- Las pruebas cortas se harán sin previo aviso en cualquier momento de la lección. Dichas pruebas podrán ser de comprobación de tareas cortas, tareas programadas, laboratorios, exposiciones, comprobaciones de lectura, comprobaciones de la materia vista durante la clase o en clases anteriores.
- En las clases se realizarán diferentes ejercicios de programación y otras prácticas relacionadas con los temas del curso, durante las clases se podrán realizar laboratorios los cuales consisten en prácticas a realizar durante la clase y entregadas al final de la misma, a no ser que el profesor indique una fecha y hora posterior para la entrega. Los laboratorios también pueden ser realizados sin previo aviso.
- El valor de cada prueba corta y laboratorio depende de la cantidad de pruebas que se realicen durante el curso.
- 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.
- Laboratorios, exposiciones y proyectos 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.
- Para ningún tipo de evaluación se aceptarán entregas después de la fecha y hora solicitadas en el sitio del curso o entregados por otros medios. Es responsabilidad de cada estudiante o grupo de trabajo enviar el archivo con suficiente tiempo antes de la hora límite.
- En las evaluaciones en las que se deba crear un programa de software deberán ser entregados con el código fuente completo y en el formato adecuado para poder ser revisado, compilado y ejecutado. Si un programa tiene errores de compilación o algún otro tipo de error que impida su ejecución será calificado con nota CERO.
- Los temas de investigación también podrán ser evaluados dentro del rubro los exámenes.
- Solo se reponen evaluaciones según lo estipulado en el artículo 24 del Reglamento de Régimen Académico Estudiantil y utilizando el procedimiento ahí descrito.
- El uso del laboratorio para fines no académicos está prohibido.
- Mantener celulares en modo silencioso o apagado durante las lecciones.
- El uso de teléfonos celulares o cualquier otro dispositivo de comunicación durante las pruebas está totalmente prohibido dentro y fuera del aula mientras el estudiante no haya entregado la prueba. Cualquier dispositivo deberá permanecer apagado y guardado en el bolso durante la prueba. El uso u aparición de cualquier dispositivo anulará automáticamente la prueba de los involucrados.
- La detección de cualquier copia o plagio, total o parcial, en cualquier evaluación anulará automáticamente la evaluación presentada por los involucrados y se elevará el caso automáticamente a la Dirección de la Sede para que se apliquen las sanciones correspondientes según el Reglamento de Orden y Disciplina Estudiantil.
- Para comunicación con el profesor(a) se utilizará únicamente el correo universitario (@ucr.ac.cr) 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 mediación virtual (<http://mediacionvirtual.ucr.ac.cr>) como apoyo del curso, entrega de material del curso y para recibo y calificación de evaluaciones.

## 7. Cronograma

	SEMANA	TEMA
1	13 – 19 Marzo	Lectura programa del curso Tema 1: Modelado de sistemas orientados a objetos con UML Tema 2: Manejo dinámico de memoria
2	20– 26 Marzo	Tema 2: Manejo dinámico de memoria Tema 3: Patrones de diseño
3	27 – 02 Abril	Tema 3: Patrones de diseño
4	03 – 09 Abril	Tema 3: Patrones de diseño
5	10 – 16 Abril	<b>Semana Santa</b>
6	17 – 23 Abril	Tema 3: Patrones de diseño Tema 4: Pruebas Unitarias
7	24 – 30 Abril	Tema 4: Pruebas Unitarias <b>Semana Universitaria</b>
8	01 – 07 Mayo	<b>I Examen</b> Tema 5: Hilos
9	08 – 14 Mayo	Tema 5: Hilos <b>Entrega y Comprobación I Proyecto Programado</b>
10	15 – 21 Mayo	Tema 6: Sockets
11	22 – 28 Mayo	Tema 6: Sockets Tema 7: Archivos especializados (XML)
12	29 – 04 Junio	Tema 7: Archivos especializados (XML)
13	05 – 11 Junio	Tema 8: Introducción al desarrollo de aplicaciones web <b>II Examen</b>
14	12 – 18 Junio	Tema 8: Introducción al desarrollo de aplicaciones web
15	19 – 25 Junio	Tema 8 Introducción al desarrollo de aplicaciones web Exposiciones trabajo investigación
16	26 – 02 Julio	Tema 8: Introducción al desarrollo de aplicaciones web Exposiciones trabajo investigación
17	03 – 09 Julio	Tema 8: Introducción al desarrollo de aplicaciones web Exposiciones trabajo investigación <b>Entrega y Comprobación II Proyecto Programado</b>
18	10 – 16 Julio	<b>Entrega de promedios</b>
19	17 – 23 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.
- 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.
- 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 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. (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*. (2016). Obtenido de <http://www.w3schools.com/>

