

**PROGRAMA CURSO: IF-3000**

I Semestre, 2017

**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:** IF3000 Programación I

**Correquisitos:** Algoritmos y Estructuras de Datos

**Ubicación en el plan de estudio:** III Ciclo

**Horario del curso:** M 17:00 a 20:50 y S 8:00 a 11:50

**Suficiencia:** No.

**Tutoría:** No.

**Datos del Profesor**

---

**Nombre:** Mario Alberto Barrantes Quesada

**Correo electrónico:** mario.barrantesquesada@ucr.ac.cr/mario.barrantes@ucrsi.info

**Horas consulta presencial:** M 16:00 a 17:00

**Horas consulta remota:** L 17:00 a 19:50s

**Medio oficial para consulta remota:** Google Hangouts

---

**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.

Además, se estudian algunas interfaces de programación de aplicaciones (API) y se introduce al estudiante 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:

- 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- Manejo dinámico de memoria
  - ArrayList
  - Punteros
  - Collection
  - Iterator
  - Vector
- 2- Creación, lectura, escritura y actualización de archivos
  - Archivos de acceso secuencial.
  - Archivos de acceso aleatorio.
  - Manejo de excepciones en java.
  - Archivos especializados (XML, JSON).
3. Hilos
  - Creación, administración y destrucción de hilos
  - Múltiples hilos
  - Sincronización
4. Patrones de diseño
  - Principios de diseño: definición, importancia y utilización.
    - Principio abierto/cerrado
    - Principio de inversión de dependencias
    - Principio de segregación de interfaces

- Principio de responsabilidad única
  - Principio de sustitución de Liskov
  - Patrones de diseño: definición, características, importancia y utilización.
  - Patrones de creación: concepto y utilización
    - Abstract Factory (Fabrica abstracta)
    - Builder (Constructor)
    - Prototype (Prototipo)
    - Singleton (Único)
  - Patrones de comportamiento: concepto y utilización
    - Observer (Observador)
    - Memento (Recuerdo)
    - Strategy (Estrategia)
  - Patrones estructurales: concepto y utilización
    - Composite (Compuesto)
    - Decorator (Decorador)
5. Pruebas Unitarias
- a. Definición prueba de software, tipos de pruebas, pruebas unitarias
  - b. Realización de Pruebas Unitarias utilizando JUnit
6. Modelado de sistemas orientados a objetos con UML
- a. El análisis/diseño orientado a objetos (ADOO)
  - b. El lenguaje de modelado unificado (UML)
  - c. Diagramas de Clase: Identificación de Clases (atributos, operaciones), herencia e interfaces.
  - d. Asociaciones de clases (uni-direccional, bi-direccional, clase asociación, agregación, reflexiva), multiplicidad y roles.
7. Introducción al desarrollo de aplicaciones web en la plataforma J2EE
- a. Las aplicaciones web
  - b. Introducción al HTML
  - c. Formularios HTML como recolectores de información
  - d. Javaser Pages (JSP)
  - e. Javabeans
  - f. Strut
  - g. Java Server Face (JSF)
8. Sockets
- a. Conexión de aplicaciones mediante sockets
  - b. Sockets en aplicaciones Cliente - Servidor.

## 5. Metodología

El curso presenta un eje de desarrollo teórico-práctico. El profesor desarrolla clases magistrales y realiza asignaciones (quices y tareas cortas) 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>
I Parcial	22%
II Parcial	22%
Quices y Tareas	8%
Laboratorios	10%
I Tarea Programada	15%
II Tarea Programada	18%
Trabajo de Investigación	5%
<b>Total:</b>	<b>100%</b>

## 7. Cronograma

INCLUIR ACÁ.

<b>Semana 1 (13/03-19/03)</b>	<b>Actividades</b>
	Introducción al curso
	Entrega de la carta al estudiante
	Manejo dinámico de memoria
<b>Semana 2 (20/03-26/03)</b>	<b>Actividades</b>
	Manejo dinámico de memoria
	Manejo dinámico de memoria
<b>Semana 3 (27/03-02/04)</b>	<b>Actividades</b>
	Manejo dinámico de memoria
	Manejo de archivos
<b>Semana 4 (03/04-09/04)</b>	<b>Actividades</b>
	Manejo de archivos
	Manejo de archivos
<b>Semana 5 (10/04-16/04)</b>	<b>Actividades</b>
No hay clases	Semana Santa

	Semana Santa
<b>Semana 6 (17/04-23/04)</b>	<b>Actividades</b>
	Hilos
	Hilos
<b>Semana 7 (24/04-30/04)</b>	<b>Actividades</b>
Semana Universitaria	Hilos / Sockets
	Hilos / Sockets
<b>Semana 8 (01/05-07/05)</b>	<b>Actividades</b>
	Patrones de diseño
	<b>I Examen</b>
<b>Semana 9 (08/05-14/05)</b>	<b>Actividades</b>
	Patrones de diseño
	Patrones de diseño
<b>Semana 10 (15/05-21/05)</b>	<b>Actividades</b>
	Patrones de diseño
	<b>Entrega I Tarea Programada</b>
<b>Semana 11 (22/05-28/05)</b>	<b>Actividades</b>
	Patrones de diseño / Pruebas Unitarias
	Patrones de diseño / Pruebas Unitarias
<b>Semana 12 (29/05-04/06)</b>	<b>Actividades</b>
	Introducción al desarrollo Web
	Introducción al desarrollo Web
<b>Semana 13 (05/06-11/06)</b>	<b>Actividades</b>
	Introducción al desarrollo Web
	<b>II Examen</b>
<b>Semana 14 (12/06-18/06)</b>	<b>Actividades</b>
	Introducción al desarrollo Web
	Introducción al desarrollo Web
<b>Semana 15 (19/06-25/06)</b>	<b>Actividades</b>
	Introducción al desarrollo Web
	<b>Exposición Proyecto Investigación</b>
<b>Semana 16 (26/06-02/07)</b>	<b>Actividades</b>
	Introducción al desarrollo Web
	Introducción al desarrollo Web
<b>Semana 17 (03/06-09/07)</b>	<b>Actividades</b>
	Introducción al desarrollo Web
	<b>Entrega II Tarea Programada</b>
<b>Semana 18 (10/06-16/07)</b>	<b>Actividades</b>
	Entrega de Promedios
<b>Semana 19 (17/06-23/07)</b>	<b>Actividades</b>
	Ampliación

## 8. Normativa del Curso

- ✓ 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 (@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 de la UCR como apoyo del curso, entrega de material del curso por parte del profesor(a) y para recibo y calificación de algunas de las evaluaciones.
- ✓ La comprobación de que alguna tarea individual, proyecto o examen es una copia aplicará 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)

## 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: 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). Patones 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. (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/>