

PROGRAMA CURSO: **IF3000 – PROGRAMACIÓN II**
I Semestre, 2018

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: Presencial

Datos del Profesor

Nombre: Wendy María Ramírez González

Correo electrónico: wendymaria.ramirez@ucr.ac.cr, wendy.ramirez@ucrsi.info

Horas consulta presencial: V 13:00 a 17:00

Horas 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.

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

- Manejo dinámico de memoria
 - ArrayList
 - Punteros
 - Collection
 - Iterator
 - Vector
- Creación, lectura, escritura y actualización de archivos
 - Archivos de acceso secuencial.
 - Archivos de acceso aleatorio.
 - Manejo de excepciones en java.
- Programación orientada a objetos
 - Composición de clases.
 - Miembros de clase estáticos.
 - Definición, implementación y uso de Interfaces.
 - Patrones de diseño
- Modelado de sistemas orientados a objetos con UML
 - El análisis/diseño orientado a objetos (ADOO)
 - El lenguaje de modelado unificado (UML)
 - Diagramas de Clase: Identificación de Clases (atributos, operaciones), herencia e interfaces.
 - Asociaciones de clases (uni-direccional, bi-direccional, clase asociación, agregación, reflexiva), multiplicidad y roles.
- Hilos
 - Creación, administración y destrucción de hilos
 - Múltiples hilos
 - Los hilos en las aplicaciones Cliente-Servidor
 - Sincronización

- Sockets
 - Conexión de aplicaciones mediante sockets
 - Sockets en aplicaciones Cliente - Servidor.

- Introducción al desarrollo de aplicaciones web en la plataforma J2EE
 - Las aplicaciones web
 - Introducción al HTML
 - Formularios HTML como recolectores de información
 - Javaser Pages (JSP)
 - Javabeans
 - Strut
 - Java Server Face (JSF)

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

Descripción	Porcentaje
Examen Parcial I	22%
Examen Parcial II	23%
Quices, Tareas Cortas, otros	10%
Laboratorios	10%
Proyecto Programado I	15%
Proyecto Programado II	15%
Trabajo de Investigación	5%
Total:	100%

Consideraciones sobre la evaluación

- La plataforma oficial que se utilizará en el curso es la plataforma institucional de Mediación Virtual (<https://mediacionvirtual.ucr.ac.cr>). La entrega de tareas, proyectos y otros evaluables será mediante el aula virtual del curso, el día y la hora indicados previamente por el docente. No se recibirán por ningún otro medio ni fuera de la fecha y hora establecida.
- Para comunicación entre profesores y estudiantes se utilizarán únicamente las cuentas de correo de la universidad (@ucr.ac.cr) y/o de la carrera (@ucrsi.info) por lo cual el estudiante debe tenerlo activo y revisarlo continuamente, ya que es el medio por el cual se

les informará sobre actualizaciones en el aula virtual, evaluaciones, asignación de tareas, entre otros.

- Durante las lecciones debe mantener los celulares en modo silencioso.
- Durante las evaluaciones el uso de teléfonos celulares, tabletas o cualquier otro dispositivo de comunicación está totalmente prohibido dentro y fuera del aula mientras el estudiante no haya hecho entrega de su evaluación. Todos dispositivos que traiga consigo deberán permanecer apagados y guardados en su bolso o salveque. El uso o aparición de algún dispositivo anulará automáticamente la evaluación de todos los involucrados.
- Los rubros de calificación de cada evaluación serán entregados junto con la especificación de la misma.
- Toda evaluación será comunicada al estudiante al menos 5 días hábiles antes de realizarse, a excepción de las pruebas cortas o “quizes”, según lo establece el Reglamento de Régimen académico.
- Las evaluaciones deben realizarse únicamente dentro de las instalaciones de la sede o recinto donde se imparte el estudiante, y no en otro lugar aunque sea parte de la UCR.
- En caso de ausencia a alguna evaluación se procederá de acuerdo a lo establecido en el Artículo 24 del Reglamento de Régimen Académico Estudiantil de la UCR para su reposición.
- La detección de una posible copia o plagio, total o parcial, en cualquier evaluación se anulará automáticamente a los involucrados y se elevará el caso a la Dirección de la Sede de Occidente para que se aplique el proceso y las sanciones correspondientes según el Reglamento de Orden y Disciplina Estudiantil.
- Todo trabajo escrito deberá entregarse en formato de documento abierto (odf) o en formato de documento portátil (pdf).
- Para todas las tareas o investigaciones solamente se permite el uso de sitios web de corte académico, como por ejemplo artículos o libros de revistas indexadas, enciclopedias Web, sitios de noticias, así como el uso de libros y periódicos. No se permite utilizar material de Internet de fuentes de información que se pueden corroborar o sean poco confiables, de lo contrario no se tomará como bibliografía o referencia válida y la nota de la evaluación será cero.
- Dentro del aula/laboratorio sólo pueden permanecer estudiantes matriculados en el curso y grupo, no se permiten “oyentes”, estudiantes matriculados en otros grupos, administrativos o terceras personas salvo que hayan sido invitados explícitamente por el profesor para fines del curso.
- El uso del aula y/o del laboratorio es exclusivo para fines académicos propios del curso. Cualquier otro uso está prohibido.
- Durante la clase velar por el orden y aseo, principalmente cuando se dan lecciones en los laboratorios, está prohibido consumir alimentos dentro de los laboratorios y no deben dejar basura de ningún tipo al terminar la clase. También asegurar que todos los equipos queden apagados al igual que los aires acondicionados y los pupitres ordenados.
- El examen de ampliación contendrá todos los contenidos del curso.

- Los laboratorios, investigaciones, exposiciones, tareas programadas y cualquier asignación grupal se realizarán en grupos que el profesor definirá, según la cantidad de estudiantes del curso.
- Si algún estudiante es expulsado(a) de su grupo de trabajo debido a incumplimiento de labores justificadas adecuadamente, perderá los puntos respectivos y no podrá continuar con el trabajo. Las pruebas respectivas para expulsar a un compañero(a) deben ser presentadas por escrito y con copia al o los involucrados y firmadas por los miembros, donde se exponga la situación presentada.
- Para las evaluaciones que impliquen programación se realizará una comprobación individual para determinar la participación de los integrantes, de la cual dependerá el total de la nota.
- La no presentación de código fuente en tareas o proyectos programados, se calificará con nota cero. Aquellos programas que no cumplan con los requerimientos mínimos establecidos en el enunciado o que presenten errores de compilación se calificará de igual manera con nota cero.
- El valor de cada prueba corta, tarea y/o laboratorio depende de la cantidad total que se realicen durante el curso, sacando un promedio de las notas obtenidas para obtener el porcentaje correspondiente.
- Las fechas de las evaluaciones del cronograma están sujetas a cambio, previamente comentados en clase, dependiendo del avance de los temas en el cronograma.
- Los laboratorios se deben de realizar en clase, excepto que se indique lo contrario. Si un estudiante no firma la hoja de entrega del laboratorio en clase no se revisará el laboratorio aunque lo haya entregado en el aula virtual.

7. Cronograma

	SEMANA	TEMA
1	12 – 18 Marzo	Lectura programa del curso Tema: Manejo dinámico de memoria
2	19– 25 Marzo	Tema: Manejo dinámico de memoria Tema: Modelado de sistemas orientados a objetos con UML
3	26 – 01 Abril	Semana Santa
4	02 – 08 Abril	Tema: Programación orientada a objetos
5	09 – 15 Abril	Tema: Programación orientada a objetos
6	16 – 22 Abril	Tema: Programación orientada a objetos Tema: Creación, lectura, escritura y actualización de archivos
7	23 – 29 Abril	Tema: Creación, lectura, escritura y actualización de archivos Semana Universitaria
8	30 – 06 Mayo	Feriado 01 de mayo I Examen
9	07 – 13 Mayo	Tema : Hilos

10	14 – 20 Mayo	Tema : Hilos
11	21 – 27 Mayo	Tema: Sockets Entrega y Comprobación I Proyecto Programado
12	28 – 03 Junio	Tema: Sockets
13	04 – 10 Junio	II Examen Tema : Introducción al desarrollo de aplicaciones web
14	11 – 17 Junio	Tema : Introducción al desarrollo de aplicaciones web
15	18 – 24 Junio	Tema : Introducción al desarrollo de aplicaciones web
16	25 – 01 Julio	Exposiciones trabajo investigación
17	02 – 08 Julio	Exposiciones trabajo investigación Entrega y Comprobación II Proyecto Programado
18	09 – 15 Julio	Entrega de promedios
19	16 – 22 Julio	Ampliación

8. Bibliografía

- Deitel, P. J., & Deitel, H. M. (2017). Java how to program, Early objects (11th ed.). NY: Pearson.
- Stelting, S., & Maassen, O. (2003). Patrones de diseño aplicados a Java. Madrid: Pearson Education.
- Booch, G., Jacobson, I., & Rumbaugh, J. (2006). El lenguaje de modelado unificado, UML. (Segunda ed.). Pearson Education.
- Freeman E., & Freeman E. (2004). Headfirst Design Patterns. USA: O´really
- Larman, C. (2003). UML y Patrones : Una introducción al análisis y diseño orientado a objetos y al proceso unificado. (Segunda ed.). España: Prentice Hall.
- Schach, S. (2005). Análisis y Diseño Orientado a Objetos con UML y el proceso unificado. México: Mc Graw Hill.
- 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., y otros. (2008). GoF Design Patterns - with examples using Java and UML2. Creative Commons Attribution.
- Naftalin M, & Wadler, P. (2006). Java Generics and Collections: Speed Up the Java Development Process. USA: O'Reilly.
- O'Docherty, M. (2005). Object-Oriented Analysis and Design Understanding System Development with UML 2.0. Inglaterra: John Wiley & Sons.
- Poo, D., Kiong, D. & Ashok, S. (2008). Object-Oriented Programming and Java. (2nd ed.). Inglaterra: Springer.

