



**PROGRAMA DEL CURSO
IF2000 - Programación I
III CICLO 2019**

1 Datos Generales

- Sigla: IF2000
- Nombre: Programación I
- Tipo de curso: Teórico - práctico
- Créditos: 4
- Horas lectivas: 8
- Requisitos: IF1300 Introducción a la computación e informática
- Correquisitos: No
- Ubicación en el plan de estudio: II Ciclo
- Suficiencia: No
- Modalidad: Regular, bajo virtual

2 Descripción

Este curso pretende introducir al estudiantado en el campo de la elaboración de programas computacionales. Se presentan los componentes básicos de todo lenguaje de programación y el manejo dinámico de memoria. Se hace énfasis en que el estudiantado asimile una serie de conceptos básicos relacionados con la calidad del software. Una característica importante del curso es que se introduce al estudiantado de manera temprana a la programación orientada a objetos.

3 Objetivo General

Introducir al estudiantado en el análisis de problemas y la abstracción de soluciones que permitan la construcción de programas a través de un lenguaje de programación.



4 Objetivos Específicos

Al finalizar el curso el o la estudiante estará en capacidad de:

- Identificar los atributos que caracterizan un producto de software de calidad para su aplicación en el desarrollo de programas.
- Describir la estructura de un lenguaje de programación para comprender su funcionalidad.
- Reconocer las ventajas que ofrece el diseño en la creación de programas de software.
- Comprender la conveniencia de separar el código de los programas en módulos independientes.
- Estudiar el concepto de programación orientada a objetos así como mostrar la importancia y aplicación del mismo en la actualidad.

5 Contenidos

1. Caracterización de un buen software

- 1.1 Confiabilidad
- 1.2 Claridad
- 1.3 Relación costo-beneficio
- 1.4 Adaptabilidad del producto
- 1.5 Reutilización
- 1.6 Estándares de documentación (interna y externa)

2. Principios de la ingeniería de software

- 2.1 Uso de la abstracción
- 2.2 Modularidad
- 2.3 Estructuras jerárquicas
- 2.4 Ocultamiento de la información



3. Objetos y tipos

- 3.1 Concepto de objetos simples (variables)
- 3.2 Características de un buen identificador para el objeto
- 3.3 Palabras reservadas
- 3.4 Tipos de datos básicos (enteros, caracteres, números reales)

4. Programación orientada a objetos

- 4.1 Clases y objetos
- 4.2 Uso de librerías
- 4.3 Importancia
- 4.4 Interfaces
- 4.5 Compilación separada
- 4.6 Enlace de los módulos orientada a objetos
- 4.7 Constructores y destructores
- 4.8 Envío de mensajes
- 4.9 Sobrecarga de métodos
- 4.10 Herencia y plantillas
- 4.11 Polimorfismo

5. Arreglos unidimensionales y bidimensionales

- 5.1 Análisis de arreglos y operaciones sobre ellos
- 5.2 Algoritmos para clasificación y búsqueda básica

6. Manejo dinámico de memoria

- 6.1 Concepto de puntero
- 6.2 Ventajas y desventajas de las estructuras dinámicas vrs las estáticas
- 6.3 Tipos definidos por el programador
- 6.4 Funciones que realizan la asignación de memoria y el trabajo con objetos dinámicos



7. Archivos y registros

7.1 Registros

7.2 Archivos, tipos y operaciones básicas

8. Recursividad

6 Metodología

Se fomenta el trabajo individual y algunos en grupo. Se enfrenta al estudiantado con problemas reales para que encuentren una solución programada, con lo que se pretende que desarrollen más su iniciativa e inventiva al resolver problemas.

Se abordarán los temas mediante clases magistrales, ejercicios prácticos, presentaciones, quices o tareas cortas, desarrollo de programas en laboratorio y proyectos programados.

La codificación y documentación de código de las aplicaciones, tanto de los ejemplos entregados por el o la docente, así como los ejercicios realizados en clase y las evaluaciones que impliquen codificación de parte de los y las estudiantes (exceptuando evaluaciones en las cuales no tengan permiso de acceder a un traductor, como los exámenes) se realizarán en idioma inglés (esto no incluye la interfaz gráfica, por ejemplo mensajes al usuario, etiquetas, texto de botones, etc; que puede ser en español)

7 Evaluación

DESCRIPCIÓN	PORCENTAJE
Examen Parcial I	20%
Examen Parcial II	20%
Quices o Tareas Cortas	10%
Laboratorios	10%
Proyecto Programado I	20%
Proyecto Programado II	20%

7.1 Consideraciones sobre la evaluación

- Según lo establecido en las resoluciones VD-R-8458-2009 y VD-R-9374-2016, se utilizará el entorno virtual de aprendizaje institucional Mediación Virtual



(<https://mediacionvirtual.ucr.ac.cr>). El mismo se empleará para la entrega del programa del curso, material, enunciados de evaluaciones, entre otros, por parte del profesorado. En el caso del estudiantado, para el envío de entregables y/o realización de evaluaciones asociadas al curso.

- Según lo establecido en la resolución R-2664-2012, que establece el correo institucional con el dominio @ucr.ac.cr como la herramienta oficial para las comunicaciones de toda la comunidad universitaria. Se utilizará el correo institucional como medio oficial de comunicación entre docentes y estudiantes, por lo cual el estudiantado deberá tenerlo activo y revisarlo continuamente.
- Durante las evaluaciones, el uso de teléfonos celulares, tabletas o cualquier otro dispositivo de comunicación está totalmente prohibido - a excepción que el profesor indique lo contrario - dentro y fuera del aula mientras el estudiante no haya hecho entrega de su evaluación. Dichos dispositivos deberán permanecer apagados y guardados en su bolso o bulto.
- Los criterios de calificación de cada evaluación serán especificados en el enunciado de la misma.
- Toda evaluación será comunicada al estudiantado del curso al menos 5 días hábiles antes de realizarse, a excepción de las pruebas cortas o “quices”, de acuerdo con lo especificado en los artículos 15 y 18 del Reglamento de Régimen Académico Estudiantil.
- En caso de ausencia a alguna evaluación, se procederá según lo establecido en el Artículo 24 del Reglamento de Régimen Académico Estudiantil.
- Ante la detección de una posible copia o plagio, total o parcial, en cualquier evaluación, se procederá de acuerdo con lo establecido en el Reglamento de Orden y Disciplina Estudiantil.
- Como parte de las lecturas de apoyo a los temas que se desarrollarán en clase, se utilizará al menos dos lecturas en idioma inglés. El objetivo principal de este aspecto es impulsar la comprensión de lectura. Debido a que hay estudiantes con diferente nivel lingüístico, los reportes y presentaciones para revisar el material leído se deben realizar en idioma español.
- Las fechas del cronograma están sujetas a cambio dependiendo del avance en los contenidos.



8 Docentes del curso

GRUPO	AULA	DOCENTE	HORARIO	CONSULTA
SEDE DEL ATLÁNTICO, RECINTO DE GUÁPILES				
31	Lab3	Mag Wilber Rodríguez Recinos	K y J 08 a 11:50	M 08 a 11:50
		wilber.rodriguezrecinos@ucr.ac.cr	K y J 13 a 16:50	M 13 a 16:50
SEDE DE OCCIDENTE, RECINTO DE GRECIA				
02	LabA	Mag Mario Barrantes Quesada	K y J 17 a 20:50	L 18 a 21:50 (V)
		mario.barrantesquesada@ucr.ac.cr	S 08 a 11:50 S 13 a 16:50	
SEDE DE OCCIDENTE, RECINTO DE SAN RAMÓN				
01	Bib1	Mag Alejandro Ulate Campos	L y V 08 a 11:50	L 17 a 20:50 (V)
		alejandroulatecampos@ucr.ac.cr	L y V 13 a 16:50	V 17 a 20:50 (V)

9 Cronograma

SEM	FECHA	TEMA O ACTIVIDAD
01	06 - 11 ENE	Lectura del programa del curso
		1: Caracterización de un buen software
		2: Principios de la ingeniería de software
		3: Objetos y tipos
02	13 - 18 ENE	4: POO (hasta 4.9, incluyendo conceptos Cont. 6 y UML)
		4: POO (hasta 4.9, incluyendo conceptos Cont. 6 y UML)
03	20 - 25 ENE	Actividad Lectura en Inglés 1
		5: Arreglos (incluyendo conceptos Cont. 6 y UML)
04	27 - 01 FEB	5: Arreglos (incluyendo conceptos Cont. 6 y UML)
		Examen Parcial I
		4: POO: Herencia y Polimorfismo (4.10 y 4.11)



SEM	FECHA	TEMA O ACTIVIDAD
05	03 - 08 FEB	4: POO: Herencia y Polimorfismo (4.10 y 4.11) Actividad Lectura en Inglés 2 Entrega Proyecto Programado I 6: Manejo dinámico de memoria
06	10 - 15 FEB	8: Recursividad
07	17 - 22 FEB	Examen Parcial II 4: POO: GUI (continuación 4.2)
08	24 - 29 FEB	7: Archivos y registros Entrega Proyecto Programado II
09	02 - 07 MAR	Ampliación

Referencias obligatorias

- Deitel, P. and Deitel, H. (2016). *Java: Cómo programar*. Pearson Education, México, 10 edition.
- Deitel, P. and Deitel, H. (2018). *Java How to Program, Early Objects*. Pearson Education, New York, 11 edition.
- Martin, R. C. (1996). The interface segregation principle: One of the many principles of ood. *C PLUS PLUS REPORT*, 8:30–36.
- Martin, R. C. (2002). The single responsibility principle. *The principles, patterns, and practices of Agile Software Development*, 149:154.

Referencias secundarias

- Barnes, D. and Kölling, M. (2016). *Objects First with Java: A Practical Introduction Using BlueJ, Global Edition*. Always learning. Pearson Education, Limited, Boston, 6 edition.
- Booch, G., Rumbaugh, J., Jacobson, I., Molina, J., and Martínez, J. (2007). *El lenguaje unificado de modelado: guía del usuario*. Addison-Wesley object technology series. Pearson Educación.
- Larman, C. (2003). *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Pearson Educación.



Schach, S. (2005). *Análisis y diseño orientado a objetos con UML y el proceso unificado*. McGraw-Hill Interamericana.

Schildt, H. (2017). *Java: A Beginner's Guide*. McGraw-Hill Education, New York, 7 edition.