



---

**PROGRAMA CURSO: PROGRAMACIÓN 1**  
II Semestre, 2011

---

### Datos Generales

**Sigla:** IF-2000

**Nombre del curso:** Programación I

**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:** IF-1300 Introducción a la computación e informática

**Correquisitos:** No tiene

**Ubicación en el plan de estudio:** II ciclo

**Horario del curso:** L 08:00 - 11:50 y V 08:00 - 11:50

**Suficiencia:**

**Tutoría:**

---

### Datos del Profesor

**Nombre:** Mci. Wendy María Ramírez González

**Correo Electrónico:** wendy.ramirez@gmail.com

**Horario de Consulta:** K 14:00 A 17:00

---

### 1. Descripción del curso

Este curso pretende introducir al estudiante 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 estudiante asimile una serie de conceptos básicos relacionados con la calidad del software. Una característica importante del curso es que se introduce al estudiante de manera temprana a la programación orientada a objetos.

---

### 2. Objetivo General

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

---



### 3. Objetivos específicos

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

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

---

### 4. Contenidos

1. Caracterización de un buen software
  - 1.1. Conceptos y ejemplos de
    - 1.1.1. Confiabilidad
    - 1.1.2. Claridad
    - 1.1.3. Relación costo-beneficio
    - 1.1.4. Adaptabilidad del producto
    - 1.1.5. Reutilización
    - 1.1.6. Estándares de documentación (interna y externa)
2. Principios de la ingeniería de software
  - 2.1. Concepto, funcionamiento, ejemplo de
    - 2.1.1. Uso de la abstracción
    - 2.1.2. Modularidad
    - 2.1.3. Encapsulamiento
    - 2.1.4. Ocultamiento de la información
    - 2.1.5. Introducción a la programación
3. Objetos y tipos
  - 3.1. Concepto de objetos simples (variables)
    - 3.1.1. Tipos de datos básicos (enteros, caracteres, números reales)
    - 3.1.2. Características de un buen identificador para el objeto.
      - 3.1.2.1. Convenciones y estándares
    - 3.1.3. Operaciones lógicas, aritméticos y relacionales
  - 3.2. Palabras reservadas



4. Programación orientada a objetos (POO)
  - 4.1. Concepto e Importancia de la POO
  - 4.2. Clases y objetos
    - 4.2.1. Concepto y ejemplos.
    - 4.2.2. Creación de clases y objetos
    - 4.2.3. Métodos, Constructores y destructores
    - 4.2.4. Envío de mensajes entre clases
  - 4.3. Compilación separada
  - 4.4. Sobrecarga de métodos
  - 4.5. Herencia y plantillas.
  - 4.6. Polimorfismos
  - 4.7. Uso de bibliotecas
    - 4.7.1. Math
    - 4.7.2. String
    - 4.7.3. Interfaz gráfica
5. Arreglos y vectores
  - 5.1. Concepto, importancia, aplicación
  - 5.2. Análisis de vectores y operaciones sobre ellos
    - 5.2.1. Crear, modificar, insertar y eliminar un vector
    - 5.2.2. Operaciones sobre vectores (multiplicación, suma, promedio, mínimo, máximo, etc.)
  - 5.3. Algoritmos para clasificación y búsqueda básica
    - 5.3.1. Ordenamiento
      - 5.3.1.1. Insertar ordenado (corrimiento)
    - 5.3.2. Búsqueda
      - 5.3.2.1. Secuencial
      - 5.3.2.2. Binaria
6. Manejo dinámico de memoria
  - 6.1. Concepto de puntero, importancia y aplicación
    - 6.1.1. Crear un puntero y operaciones
  - 6.2. Ventajas y desventajas de las estructuras dinámicas vrs las estáticas.
7. Archivos y registros
  - 7.1. Concepto archivo y registro, relación entre archivo y registro
  - 7.2. Importancia y aplicación de los archivos
  - 7.3. Tipos de archivos



- 7.3.1. Secuenciales (texto y binarios)
- 7.3.2. Aleatorios
  
- 7.4. Operaciones básicas sobre archivos.
  - 7.4.1. Crear, abrir, modificar, cerrar un archivo.
  - 7.4.2. Búsqueda de datos en archivos.
  
- 8. Recursividad.
  - 8.1. Concepto e importancia
  - 8.2. Diferencia de recursividad vr. ciclos.
  - 8.3. Ejemplos básicos de recursivos.

---

## 5. Metodología

Se debe fomentar el trabajo individual y algunos en grupo, enfrentar al estudiante con problemas reales para que les encuentre una solución programada. Con lo que se pretende que el estudiante desarrolle más su iniciativa e inventiva al resolver problemas. Se abordarán los temas mediante clases magistrales, ejercicios prácticos, presentaciones, desarrollo de programas en laboratorio y tareas de investigación.

---

## 6. Evaluación

<i>Descripción</i>	<i>Porcentaje</i>
Examen Parcial I	10%
Examen Parcial II	15%
Examen Parcial III	15%
Laboratorios	15%
Quices, Tareas Cortas o Trabajos Clase	10%
Tarea Programada 1	15%
Tarea Programada 2	20%

**Total: 100%**

### Consideraciones sobre la evaluación

- ✓ Los Quices se harán sin previo aviso en cualquier momento de la lección y no se harán reposiciones.
- ✓ Las tareas programadas se realizarán en grupos establecidos por el profesor.



- ✓ Las tareas programadas deben ir acompañadas de la respectiva documentación. No se recibirá la misma posterior a la entrega.
  - ✓ Los rubros de calificación de cada una de las evaluaciones serán entregadas junto con la especificación de la misma.
  - ✓ 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, perderán el 40% del valor del trabajo.
  - ✓ Para los demás aspectos de evaluación no se aceptarán entregas después de la fecha y hora solicitada.
  - ✓ Para todas las entregas sólo se recibirán aquellas cuyo contenido sea exclusivamente de la asignación respectiva.
  - ✓ 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 o examen es una copia hará que se apliquen 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).
  - ✓ Se utilizará la plataforma de Mediación Virtual como apoyo del curso.
-



## 7. Cronograma.

	SEMANA	TEMA
1	08 – 12 Agosto	Tema 1: Caracterización de un buen software. Tema 2: Principios de la ingeniería de software
2	15 – 19 Agosto <b>(L 15 Agosto Feriado)</b>	Continuación Tema 2. Tema 3: Objetos y tipos
3	22 – 26 Agosto	Continuación Tema 3. Tema 4: Programación orientada a objetos
4	29 – 02 Septiembre	Continuación Tema 4.
5	05 – 09 Septiembre	<b>I Examen*</b> Continuación Tema 4.
6	12– 16 Septiembre	Continuación Tema 4.
7	19 – 23 Septiembre	Tema 5: Arreglos y vectores
8	26 – 30 Octubre	Continuación Tema 5.
9	03 – 07 Octubre	Continuación Tema 5.
10	10 – 14 Octubre	Tema 6: Manejo dinámico de memoria <b>II Examen*</b>
11	17 – 21 Octubre <b>(L 17 Octubre Feriado)</b>	Continuación Tema 6.
12	24 – 28 Octubre	Tema 7: Archivos y registros
13	31 – 04 Noviembre	Continuación Tema 7.
14	07 – 11 Noviembre	Continuación Tema 7.
15	14 – 18 Noviembre	Tema 8: Recursividad
16	21 – 25 noviembre	<b>III Examen*</b>
17	28 – 02 Diciembre	<b>Entrega de Notas</b>
18	05 – 09 Diciembre	<b>Ampliación</b>

\* Las fechas de examen están sujetas a cambio dependiendo del avance de los temas vistos.

## 8. Bibliografía

- ✓ Deitel, P. J., & Deitel, H. M. (2008). *Cómo programar en Java* (Séptima ed.). México: Pearson Education.
- ✓ Wu, C. T. (2008). *Programación en Java*. México: McGraw-Hill.
- ✓ Joyanes Aguilar, L. (2008). *Fundamentos de programación* (Cuarta ed.). México: MacGraw-Hill.
- ✓ Carballo, F. J. (2008). *Java 2, Curso de programación* (Tercera ed.). México: Alfaomega.
- ✓ Ceballos, F. J. (2008). *Java 2, Interfaces gráficas y aplicaciones para Internet* (Tercera ed.). México: Alfaomega.
- ✓ Jamsa, K. (1996). *Aprenda C++ Paso a Paso*. AlfaOmega.
- ✓ Sánchez Allende, J., & otros. (2001). *Java 2*. McGraw-Hil.