

Universidad de Costa Rica  
Escuela de Ciencias de la Computación e Informática  
CI0202 - Principios de informática  
Grupo 11: K11-13 103-IF, V11-13 212-IN.

Profesor Jeisson Hidalgo-Céspedes  
Correo: [jeissonh@gmail.com](mailto:jeissonh@gmail.com)  
Casillero: 04 de la Escuela de Computación.

# Carta al estudiante

## Descripción del curso

El curso de *Principios de Informática*, es un curso básico de programación para estudiantes del área de ingeniería y afines. Introduce al estudiante al pensamiento abstracto para la resolución de problemas de ingeniería y científicos, automatizable por medio de herramientas informáticas de desarrollo, utilizando metodologías sistemáticas. El estudiante aprenderá a reconocer la aplicabilidad de flujos de control y modelos de datos básicos para lograr el diseño e implementación de programas y algoritmos.

## Objetivos

Proveer formación básica en programación y construcción de algoritmos y de programas, para la resolución de problemas utilizando técnicas actuales.

Al finalizar este curso el estudiante será capaz de:

- Diseñar, organizar e implementar algoritmos para resolver problemas específicos del área de ingeniería, ciencias y afines.
- Usar un ambiente de programación para la edición, prueba y depuración de programas.
- Reutilizar componentes de software.
- Aplicar buenas prácticas de construcción de software.

## Contenido del curso

1. Fundamentos de la programación
  - a. Lenguajes de programación: concepto de programación, lenguaje máquina, lenguaje ensamblador, lenguaje de alto nivel, máquina virtual, compilador y paradigmas.
  - b. Ciclo de vida de un programa: problema, análisis, diseño, implementación y prueba.
  - c. Algoritmo: concepto, primitivas y ejemplos.
2. Introducción a la programación orientada a objetos
  - a. Paradigma: clases e instancias, atributos y métodos, abstracción y reutilización.
  - b. Análisis y diseño: modelaje de clases e instancias.
  - c. Compilación y ejecución.
3. Sistemas numéricos y representación de datos
  - a. Bases y conversión: decimal, binaria y hexadecimal.
  - b. Sistemas de codificación: ASCII y UNICODE.
4. Tipos de datos
  - a. Tipos de datos: primitivos (enteros, reales, booleano y caracteres) e hileras.
  - b. Precisión: entero (byte, short, int y long), real (float y double), booleano, carácter e hilera (ejemplo: secuencias de escape).
5. Definición de variables

- a. Declaración: tipo, identificador y dirección.
- b. Inicialización: tipo primitivo (valor), instancia (referencia) y estado de memoria.
- c. Asignación y conversión (*type casting*).

#### Utilización de variables

- a. Atributos de clase: declaración, ámbito de vida y ocultamiento (encapsulamiento).
- b. Variables locales: declaración y ámbito de vida.
- c. Estáticas y constantes: declaración y ámbito de vida.

#### 6. Entrada y salida básica

- a. Entrada: parámetros de línea de comandos y diálogo.
- b. Salida: línea de comandos y diálogo.

#### Manejo de excepciones

- a. Concepto, ejemplos y definición (clase), lanzamiento y atrape.

#### 7. Expresiones y operadores

- a. Aritméticos binarios (multiplicativos y aditivos) y unarios (negación y posfijos).
- b. Relacionales (comparación e igualdad), lógicos (binarios y unarios) y asignación.
- c. Evaluación y orden de precedencia.

#### 8. Instrucciones y estructuras de control

- a. Estructuras de secuenciación (`{}`).
- b. Estructuras de selección o bifurcación (`if/else` y `switch`).
- c. Estructuras de repetición o iteración (`while`, `do-while` y `for`).

#### 9. Métodos: fundamentos

- a. Conceptos: modularización y reutilización, declaración e invocación.
- b. Componentes: encabezado (identificador, parámetros y tipo de retorno) y cuerpo.
- c. Métodos estáticos (funciones).
- d. Sobrecarga: declaración, firmas y resolución de llamados.

#### Métodos: funcionamiento

- a. Paso de argumentos: por valor y por referencia.
- b. Estado de memoria estática, memoria dinámica y pila de llamados.
- c. Reglas de alcance o ámbito de identificadores.

#### Constructores

- a. Concepto y utilización: declaración e invocación.

#### 10. Recursividad

- a. Concepto y utilización.
- b. Orden de llamados.

#### 11. Arreglos o vectores: fundamentos

- a. Concepto, estructura y estado de memoria.
- b. Declaración e inicialización.
- c. Acceso a celdas y recorrido.
- d. Parámetros de tipo arreglo y paso de argumentos.

#### Arreglos o vectores: operaciones

- a. Utilidad y operaciones comunes (suma, promedio, mínimo, máximo).
- b. Búsqueda: primitivos e instancias.
- c. Ordenamiento: primitivos e instancias.

#### 12. Matrices

- a. Concepto, estructura y estado de memoria.
- b. Declaración e inicialización.
- c. Acceso a celdas y recorrido.

#### 13. Hileras o cadenas de caracteres

- a. Concepto.
- b. Operaciones: concatenación, obtener tamaño, extraer carácter o fragmento, comparación, búsqueda, reemplazo, conversión a mayúscula o minúscula, conversión a arreglo.

#### 14. Entrada y salida: archivos

- a. Conceptos y organización física de archivos.
- b. Operaciones de archivos: lectura y escritura.
- c. Procesamiento binario/textual: apertura/cierre y lectura/escritura.

15. Programación avanzada desarrollando uno de los siguientes temas:
  - a. Matrices, algoritmos y bibliotecas de álgebra lineal.
  - b. Fundamentos de graficación y de interfaces gráficas.
  - c. Punteros y referencias, copia y clonación de objetos.
  - d. Algoritmos de búsqueda y ordenamiento básicos.
  - e. Herencia y polimorfismo

## Metodología

El curso se basa en 2 lecciones magistrales impartidas por el profesor y 2 lecciones en el laboratorio de cómputo. En las lecciones magistrales se utiliza los recursos audiovisuales y pizarra, para mostrar conceptos e implementaciones. En las lecciones de laboratorio se implementarán los ejercicios cortos de programación.

Se asignan practicas, tareas cortas y proyectos en los cuales el estudiante podrá profundizar individualmente los temas desarrollados en clase. Parte de la evaluación será a través de exámenes cortos, los cuales se aplicarán en tiempo de clase a criterio y organización del profesor. Se realizarán tres exámenes parciales.

## Recursos didácticos

- *Equipo audiovisual* con presentaciones referentes a cada tema, donde se explica gráficamente los conceptos a desarrollar y se ofrecen los elementos teóricos básicos para el conocimiento de las distintas estructuras de datos y algoritmos.
- *Pizarra* para explicar en más detalle el funcionamiento de los algoritmos y características de las estructuras de datos. Es un recurso importante para la clarificación de dudas prácticas y presentación de ejercicios prácticos.
- *Prácticas de ejercicios y problemas* redactados por los profesores de la cátedra o de libros de referencia.
- *Laboratorio* para conocer los conceptos básicos de una herramienta visual.

## Evaluación

- 20%. **Quices**. Ejercicios cortos que deberá el estudiante resolver en forma individual al iniciar la clase. No se repondrán por llegadas tardías.
- 20%. **Tareas cortas**. Ejercicios que deberá resolver el estudiante individualmente de forma extraclase.
- 60%. **Exámenes parciales**. Tres exámenes de igual ponderación, cuyas fechas serán comunicadas oportunamente por el profesor.

## Observaciones

1. En toda asignación, sea en papel o digital, se evaluará la indentación, uso correcto de paréntesis (redondos, cuadrados y llaves), la eficiencia y las buenas prácticas de programación.
2. La nota  $N$  de una tarea o proyecto entregado  $h$  horas tarde se calculará como  $N = x - \frac{25}{18} h$ , donde  $x$  es la nota que habría obtenido si se hubiere entregado a tiempo.
3. Cualquier evaluación donde se detecte plagio, su calificación será anulada por completo.

## Bibliografía

1. Deitel, P.J.; Deitel, H.M. *Java Cómo programar, séptima edición*. ISBN 978-97-0261-190-5. Prentice-Hall Pearson, México, 2008.
2. Barnes, David J.; Kölling Michael. *Programación orientada a objetos con Java*. ISBN 978-84-8322-350-5. Pearson Educación, 2007.
3. Ceballos, Francisco Javier. *Java 2 - Curso de Programación - 3ra edición*. ISBN 970-15-1164-6. Alfaomega Ra-Ma, 2006.

# Cronograma

Semana	Fecha	Tema
1	13-ago	Introducción a la programación
2	20-ago	Variables, tipos de datos, expresiones y operadores
3	27-ago	Estructuras de control condicional: if, if-else, switch
4	03-set	Estructuras de control repetitivo (ciclos): while
5	10-set	Estructuras de control repetitivo (ciclos): for, do-while
6	17-set	Objetos: clases, instancias y referencias
7	24-set	Cadenas de caracteres. <b>Examen 01</b> (27-set semanas 1-5)
8	01-oct	Métodos (parte 1)
9	08-oct	Métodos (parte 2)
10	15-oct	Arreglos/vectores (parte 1)
11	22-oct	Arreglos/vectores (parte 2). <b>Examen 02</b> (25-oct semanas 6-9)
12	29-oct	Matrices
13	05-nov	Entrada y salida: archivos
14	12-nov	Herencia*
15	19-nov	Interfaces gráficas*. <b>Examen 03</b> (22-nov semanas 10-13)
16	26-nov	Manejo de eventos*

\* Temas extra o avanzados.