

Programa del curso CE-1101

## **Introducción a la Programación**

Área Académica de Ingeniería en Computadores

Programa de Licenciatura en Ingeniería en Computadores.

[Última revisión del programa: 22 de enero de 2018]

## I parte: Aspectos relativos al plan de estudios

### 1 Datos generales

<b>Nombre del curso:</b>	Introducción a la Programación
<b>Código:</b>	CE-1101
<b>Tipo de curso:</b>	Teórico - Práctico
<b>Electivo o no:</b>	Obligatorio
<b>Nº de créditos:</b>	3
<b>Nº horas de clase por semana:</b>	4
<b>Nº horas extraclase por semana:</b>	5
<b>% de las áreas curriculares:</b>	--
<b>Ubicación en el plan de estudios:</b>	Curso del 1er semestre del Programa de Licenciatura en Ingeniería en Computadores.
<b>Requisitos:</b>	No hay
<b>Correquisitos:</b>	Taller de Programación (CE-1102)
<b>El curso es requisito de:</b>	Algoritmos y Estructuras de Datos I (CE-1103)
<b>Asistencia:</b>	Obligatoria
<b>Suficiencia:</b>	No
<b>Posibilidad de reconocimiento:</b>	Sí
<b>Vigencia del programa:</b>	I Semestre de 2018

## 2 Descripción general

*“La mayoría de los buenos programadores programan no porque esperan que les paguen o que el público los adore, sino porque programar es divertido.”*

Linus Torvalds

Este curso comprende una introducción a los conceptos básicos para modelar y solucionar problemas de forma algorítmica, así como las estrategias fundamentales para la construcción de programas de computadoras. Le da una visión general para poder ingresar a tecnologías, lenguajes y temas computacionales más avanzados. Se cubre el proceso de resolución algorítmica de problemas, algoritmos numéricos, técnicas con listas y estructuras de datos avanzadas, con lo que se sientan las bases para el curso “Algoritmos y Estructuras de Datos” (CE-1103).

Adicionalmente se estimula la participación en clase y el trabajo en grupo para la presentación de propuestas de solución a problemas planteados en el aula. Adicionalmente se estimula la participación en clase y el trabajo en grupo para la presentación de propuestas de solución a problemas planteados en el aula, proyectos grupales y exposiciones.

La metodología del curso busca además reforzar ciertas habilidades el uso de diferentes herramientas y plataformas para el desarrollo de programas informáticos.

Los atributos que se ejercitan a lo largo de las diferentes actividades de este y que se proponen que el estudiante pueda alcanzar al final del curso son:

- Conocimiento base de Ingeniería (CB) en nivel inicial.
- Análisis de problemas (AP) en nivel inicial.
- Uso de herramientas de ingeniería (HI).

El curso es adaptable a necesidades educativas especiales que puedan tener los estudiantes.

### 3 Objetivos

#### Objetivo General:

Al finalizar el curso el estudiante deberá ser capaz de:

- Diseñar, analizar y desarrollar algoritmos básicos y realizarlos de manera sistemática en un modelo computacional específico o lenguaje de programación

#### Objetivos Específicos:

Objetivo(s) del curso	Atributo(s) correspondiente(s)	Nivel de desarrollo de cada atributo que se planea alcanzar: Inicial - I, intermedio - M o avanzado - A
1. Conocer algoritmos fundamentales de las ciencias de la computación y utilizarlos para el diseño de soluciones algorítmicas a problemas específicos.	Conocimientos base de ingeniería (CB)	I
2. Diseñar y especificar algoritmos por medio de abstracciones funcionales.	Análisis de problemas (AP)	I

3. Conocer y delimitar las etapas básicas asociadas en la construcción de algoritmos.	Análisis de Problemas (AP)	I
4. Analizar la calidad, correcta escritura, eficiencia y mantenibilidad de una solución algorítmica.	Análisis de Problemas (AP)	I
5. Conocer y utilizar los principales elementos semánticos y sintácticos de un modelo de programación concreto para el diseño de algoritmos con un determinado fin.	Uso de Herramientas de Ingeniería (HI)	I

## 4 Contenidos

### **Introducción a la programación (1 semana).**

- Definiciones básicas
- Lenguajes de programación
- El concepto de algoritmo.
- Ciclo de solución de problemas
- Método de solución de problemas
- Aplicación del método de solución de problemas.

### **El lenguaje de programación (2 semanas)**

- Características del lenguaje
- El entorno de programación
- Expresiones, identificadores, tipos de datos y operadores
- Importación de librerías
- Expresiones relacionales
- Expresiones lógicas

### **Escritura de programas (1 semanas)**

- Funciones escritas en el lenguaje
- Ejecución de funciones
- Errores en un programa
- Depurando la solución
- Comentando el código
- Solución de problemas simples
- Estructuras de control
- Estructura de decisión
- Manejo de argumentos
- Estilo de codificación

### **Introducción a la recursividad con números (3 semanas).**

- Concepto de recursividad
- Funciones recursivas con números
- Verificación de restricciones y casos especiales.
- Cambios de base numérica

### **Programación recursiva (3.5 semanas).**

- Funciones recursivas con listas
- Recursividad de cola
- Funciones recursivas con números
- Funciones recursivas con listas
- Formalización de la recursividad
- Cálculos por aproximación

### **El enfoque iterativo (1 semana).**

- Concepto de iteración.
- Las estructuras while y for
- Comparación de recursividad e iteración.
- Funciones iterativas con números.
- Funciones iterativas con listas

- Estructuras break y continue

**Estructuras de datos (2 semanas).**

- Algoritmos de búsqueda.
- Algoritmos de ordenamiento.
- Operaciones sobre strings, vectores, matrices.

**Programación orientada a objetos (2 semanas).**

- Introducción a la terminología de objetos.
- Acceso a los atributos de una clase
- Constructores y destructores.
- Composición: referencias a objetos como miembros de clases.
- Encapsulado: abstracción y ocultamiento de datos.
- Reutilización de código.
- Herencia
- Sobre-escritura de métodos.
- Composición vs. herencia
- Interfaces e implementación.
- Polimorfismo.

**Introducción a JAVA (1.5 semanas).**

- Introducción a la plataforma JAVA.
- Acceso a los atributos de una clase
- Solución de problemas simples
- Estructuras de control
- Estructura de decisión

## **II parte: Aspectos operativos**

### **5 Metodología de enseñanza y aprendizaje**

Se emplearán técnicas de clases magistrales por parte del profesor, en donde se desarrollarán los aspectos teórico-prácticos más relevantes de los diferentes temas.

Esto se combinará con una alta participación por parte de los estudiantes durante el transcurso de las lecciones, por medio de llamadas orales, respuestas a casos en la pizarra y de trabajos en grupo.

Se presupone que el alumno profundiza los temas abordados en la clase con lecturas y realización de ejercicios recomendados por el profesor.

El estudiante será responsable de desarrollar los proyectos programados y prácticas que se asignen en el curso. Adicionalmente con el fin de evaluar los conocimientos adquiridos, los estudiantes elaborarán pruebas cortas, evaluaciones, tareas, proyectos programados, así como participación en clase.



## 6 Evaluación

Los siguientes son los rubros que se evaluarán en el curso:

Rubro	Valor (%)	Tiempo estimado de realización
I Examen Parcial	25	Semana 8
II Examen Parcial	25	Semana 12
III Examen Parcial	25	Semana 17
Tareas y quices	25	1 por semana
<b>Total</b>	<b>100</b>	

## 7 Bibliografía

### Obligatoria:

- Downey, A., Elkner, J., Meyers, C. How to think like a computer scientist. Green Tea Press. 2002.
- Deitel, H. et all. Python: How to program. Prentice Hall, 2002.

### Complementaria:

- Abelson, H. Sussman, G. Structure and Interpretation of Computer Programs.
- The MIT Press, McGraw-Hil Company, Massachusetts, 1985.

## 8 Profesores

### Jeff Schmidt Peralta

- Grupo 01
- Consultas: 24/7 al grupo de Facebook o al correo electrónico del profesor.
- Consulta presencial: en el CIC miércoles de 1:30 pm a 4:30 pm.
- Correo electrónico: [jschmidtcr@gmail.com](mailto:jschmidtcr@gmail.com)

### Milton Villegas Lemus

- Grupo: 02.
- Consultas: Google Classroom [IntroTaller2018G2](#).
- Consulta presencial: en oficina 12 de la Escuela de Computación, los martes y jueves de 4:30 pm a 6:00 pm.
- Correo electrónico: [mvilem@gmail.com](mailto:mvilem@gmail.com)
- Asistente: Santiago Gamboa ([santigr17@gmail.com](mailto:santigr17@gmail.com))

### Pedro Gutiérrez García

- Grupo: 03.
- Consultas: al grupo de Facebook
- Consulta presencial: en oficina, los martes y jueves de 1:00 pm a 3:00 pm.
- Correo electrónico: [faq.intro@gmail.com](mailto:faq.intro@gmail.com)

### Dr Antonio González Torres

- Grupo: 04.
- Consultas: email.
- Consulta presencial: en oficina 12 de la Escuela de computación, los martes y jueves de 1:00 pm a 5:00 pm.
- Correo electrónico: [antonio.gonzalez@itcr.ac.cr](mailto:antonio.gonzalez@itcr.ac.cr)

### Ing. Fabián Zamora Ramírez

- Grupo: 05.
- Consultas: [fzamora@itcr.ac.cr](mailto:fzamora@itcr.ac.cr) o a convenir con el profesor
- Consulta presencial: en oficina 410 edificio K1, los viernes de 1:00 pm a 3:00 pm.
- Correo electrónico: [fzamora@itcr.ac.cr](mailto:fzamora@itcr.ac.cr)

**9 Normativa  
del curso**

1. El curso se aprueba con nota de 70. No hay examen de reposición.
2. Las actividades o trabajos académicos deben cumplir con las normas dictadas por el Consejo de Escuela. Si alguna actividad implica la entrega de un medio físico, debe estar libre de virus, sino tendrá una nota de cero.
3. No se aceptarán trabajos después de la fecha y hora indicadas. Por lo tanto, trabajos entregados tardíamente tendrán una nota de cero. Los quices no se reponen.
4. Para proyectos, el profesor especificará la penalización por entrega extemporánea según su criterio.
5. Los exámenes escritos han de entregarse en cuadernillos oficiales del ITCR, debidamente engrapados y presentados. En caso de solicitarse el examen en formato digital, debe enviarse en la fecha indicada por el profesor en forma obligatoria como requisito para la evaluación.
6. El contenido académico de las actividades, ya sean estas tareas, exposiciones, quices, exámenes parciales o proyectos, son acumulativos.
7. Los fraudes en cualquier actividad llevada a cabo durante el semestre implicará que se perderá el curso y se reportará la nota mínima. Además se reportará una carta al expediente del estudiante.
8. La asistencia a clases es obligatoria. Si un estudiante falta al 15% o más del total de horas semestrales, pierde automáticamente el curso. Si el estudiante llega 15 minutos después de iniciada la lección se considerará ausente del periodo.
9. El profesor se reserva el derecho de calificar forma y fondo de las actividades tomando como referencia la mejor actividad presentada.
10. Durante la clase se pide apagar o silenciar aparatos celulares, no interrumpir el tema de la lección, para que esta fluya con en un ambiente de respeto e intercambio de palabra ordenado.