

Programa del curso CE1101

Introducción a la Programación

Área Académica de Ingeniería en Computadores

Licenciatura en Ingeniería en Computadores.

[Última revisión del programa: 24 de Julio de 2022]

I parte: Aspectos relativos al plan de estudios

1 Datos generales

Nombre del curso:	Introducción a la Programación
Código:	CE1101
Tipo de curso:	Teórico - Práctico
Electivo o no:	Obligatorio
Nº de créditos:	3
Nº horas de clase por semana:	4
Nº horas extraclase por semana:	5
% de las áreas curriculares:	Ciencias de Ingeniería (100%)
Ubicación en el plan de estudios:	Curso del 1er semestre del Programa de Licenciatura en Ingeniería en Computadores.
Requisitos:	No hay
Correquisitos:	Taller de Programación (CE1102)
El curso es requisito de:	Algoritmos y Estructuras de Datos I (CE1103)
Asistencia:	Obligatoria
Suficiencia:	No
Posibilidad de reconocimiento:	Sí
Vigencia del programa:	II Semestre de 2022

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 sesión de lecciones.

Adicionalmente se estimula la participación en las sesiones 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:

- Análisis de problemas (AP) en nivel inicial.
- Herramientas de ingeniería (HI) en nivel inicial.
- Aprendizaje continuo (AC) en nivel inicial.

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.	Aprendizaje Continuo (AC)	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 para el diseño de algoritmos con un determinado fin.	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 bibliotecas
- 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 (0.5 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 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 las tareas, tanto las programadas como las escritas, y prácticas que se asignen en el curso, en clase o fuera de la misma. Adicionalmente con el fin de evaluar los conocimientos adquiridos, los estudiantes elaborarán pruebas cortas, evaluaciones, tareas programadas y escritas, así como participación en clase, tanto de manera individual como grupal.

Para el desarrollo del curso se necesitará acceso a un computador personal y software de desarrollo específico (Python IDLE). Estos recursos son de acceso libre, por lo tanto, los estudiantes podrán consultarlos de acuerdo con sus necesidades.

6 Evaluación

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

Rubro	Valor (%)	Tiempo estimado de realización
I Examen Parcial	25	Semana 7
II Examen Parcial	25	Semana 11
III Examen Parcial	25	Semana 16
Tareas y Quices	25	1 por semana
Total	100	

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
- Aula F2-07 (Caso presencial)
- Consultas: grupo de Telegram o al correo electrónico del profesor.
- Clases sincrónicas vía Grupo de Teams
- Consulta Teams: Miércoles de 1:30 pm a 3:30 pm.
- Correo electrónico: jschmidtcr@gmail.com

Milton Villegas Lemus

- Grupo: 02.
- Aula F2-10 (Caso presencial)
- Clases Sincrónicas vía Grupo de Microsoft Teams
- Muro de Consultas: Google Classroom Intro G2S122
- Consulta Teams Equipo Intro/Taller, Martes de 4:30 pm a 6:00 pm.
- Correo electrónico: mvilem@gmail.com
- Asistente: Oscar Acuña (arturo.acuna.duran@gmail.com)

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 Área. 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 presenciales han de entregarse en cuadernillos, debidamente engrapados y presentados. Para los exámenes sincrónicos, éstos son a mano alzada, se deben digitalizar (scan), enviarse a la hora y fecha indicada. En caso de solicitarse la versión digital del examen, esta debe enviarse en la fecha y hora indicada a la asignación abierta en el Classroom para ello o en plataforma equivalente especificada por el profesor, se recalca que en estos casos es obligatorio y requisito para la evaluación. El contenido académico de las actividades, ya sean estas tareas, exposiciones, quices, exámenes parciales o proyectos, son acumulativos.
6. Los fraudes en cualquier actividad o asignación llevada a cabo durante el semestre, implicará que se reportará la nota mínima y se aplicará la reglamentación establecida. Además, si aplica se reportará una carta al expediente del estudiante.
7. El profesor se reserva el derecho de calificar forma y fondo de las actividades tomando como referencia la mejor actividad presentada.