**Facultad de Ingeniería y Ciencias  
Escuela de Informática y Telecomunicaciones**

**PROGRAMA DE ASIGNATURA**Programación Avanzada

1. **Identificación**
   * Nombre : Programación Avanzada
   * Código : CIT1010
   * Créditos : 6
   * Duración : Semestral
   * Ubicación en plan de estudio: Semestre 2
   * Requisitos : CIT-1000 Programación
   * Sesiones semanales : 2 cátedras, 1 ayudantía
2. **Descripción del curso**

Este curso introduce los fundamentos de la programación orientada a objetos (POO) como paradigma para el diseño y construcción de software modular, reutilizable y mantenible. A través del uso de clases, objetos, herencia, polimorfismo, encapsulamiento y abstracción, las y los estudiantes aprenderán a modelar soluciones a problemas reales. El curso también aborda algoritmos fundamentales de ordenamiento, búsqueda y recursividad, así como el análisis básico de eficiencia en términos de tiempo y uso de memoria. Al finalizar, las y los estudiantes estarán en condiciones de desarrollar programas orientados a objetos que integren estructuras de datos, algoritmos y buenas prácticas de diseño.

1. **Resultados de aprendizaje**

Al finalizar el curso el alumno será capaz de:

1. Diseñar e instanciar clases y objetos, aplicando principios de encapsulamiento y abstracción, para construir programas estructurados, mantenibles y orientados a representar entidades del mundo real.
2. Reutilizar y extender código mediante herencia y polimorfismo, para facilitar la escalabilidad del software y la implementación de nuevas funcionalidades sin modificar código existente.
3. Implementar y analizar algoritmos clave (recursivos, ordenamientos clásicos, búsquedas) sobre listas y colecciones, para resolver problemas computacionales de forma eficiente en cuanto a tiempo y uso de memoria.
4. Modelar y resolver problemas reales simples, eligiendo estructuras de datos y estrategias adecuadas, para desarrollar soluciones contextualizadas y fundamentar decisiones técnicas en base al análisis de eficiencia.
5. **Unidades Temáticas**

* **Clases y objetos**
  + Definición, instanciación, atributos y métodos.
  + Encapsulamiento y abstracción.
* **Herencia y polimorfismo**
  + Reutilización de código y extensión de funcionalidades.
  + Composición vs. herencia.
* **Algoritmos y estructuras de datos**
  + Recursividad, ordenamientos (ej: Selection, Insertion, Merge Sort).
  + Algoritmos de Búsqueda (ej: secuencial y binaria )
  + Implementación sobre listas y colecciones.
* **Modelado y eficiencia**
  + Resolución de problemas reales simples.
  + Análisis de eficiencia (tiempo y memoria).
  + Justificación de decisiones técnicas.

1. **Metodología**

Se contempla la realización de 2 sesiones de cátedra en aula de clases. Además, se dispone de tutoría en Laboratorio, donde se aplicará la técnica de estudio de casos, en las que las materias del curso se irán desarrollando en base a los ejemplos aplicados que en cada sesión se planteen.

Durante el semestre se realizarán evaluaciones en laboratorios de las unidades temáticas vistas hasta la fecha de cada evaluación. Además, realizará un trabajo relacionado a la unidad de **Modelado utilizando contenedores de información**, el cual contempla la entrega de un informe asociado a la solución diseñada.

El curso será apoyado con apuntes de clases almacenados en un servidor para acceso vía Internet, en el que se describirán las materias y los ejercicios que deberán realizarse clase a clase.

Dado que esta es una asignatura con una orientación práctica, se ha contemplado que la evaluación del curso considere los desarrollos realizados por el alumno sesión a sesión.

1. **Evaluación**

Se contempla la realización de evaluaciones parciales (controles, trabajos, prácticas de laboratorio, entre otros), dos pruebas solemnes de igual ponderación y un examen.

Las notas parciales contemplan un mínimo de 3 laboratorios más el trabajo de archivos descrito en la sección metodología, pudiendo incluirse hasta dos evaluaciones parciales adicionales según el criterio del profesor(a) de la asignatura. En cualquier caso, el promedio de notas parciales corresponde al promedio simple entre las evaluaciones antes descritas.

La nota de presentación a examen se calcula de la siguiente forma:

Nota de Presentación = (25% Solemne 1 + 25% Solemne 2 + 20% Notas Parciales)/0.7

A su vez, la nota final del curso está dada por:

Nota Final= 25% Solemne 1 + 25% Solemne 2 + 20% Notas Parciales + 30% Examen.

Según regla general, para aprobar el curso debe tenerse que Nota Final ≥ 4,0 y para presentarse a Examen se debe cumplir con una Nota de Presentación ≥ 3,5.

La inasistencia a una prueba solemne implicará reemplazo de su nota con la Nota de Examen. El profesor podrá eximir del examen a aquellos alumnos cuyos promedios de notas parciales y de pruebas solemnes sean superiores a 5.0.

1. **Bibliografía Básica**
2. **Curso Intensivo de Python**, Eric Matthes
3. Python para análisis de datos​, Wes McKinney
4. Apuntes de Cátedra.



Elaborado por: Jonathan Frez

Fecha revisión: Marzo de 2025

Fecha vigencia: Marzo de 2026