

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

**Descriptor de asignatura**

*Programación Avanzada*

**1. Identificación de la asignatura:**

|                                                |                                              |
|------------------------------------------------|----------------------------------------------|
| Nombre de la Asignatura: Programación Avanzada |                                              |
| Códigos: CIT-1110                              | Créditos: 5                                  |
| Duración: Semestral                            | Ubicación en el plan de estudios: Semestre 2 |
| Requisitos: CIT-1100 Programación              |                                              |
| Sesiones cátedras semanales: 2 cátedras        |                                              |
| Sesiones de Ayudantía: 1                       |                                              |

**2. Descripción de la asignatura:**

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.

**3. Resultados de aprendizaje:**

1. Diseña e implementa clases y objetos aplicando los principios de encapsulamiento y abstracción, con el propósito de desarrollar programas estructurados, mantenibles y orientados a la representación de entidades del mundo real.
2. Reutiliza y extiende código mediante herencia y polimorfismo, para facilitar la escalabilidad del software y la implementación de nuevas funcionalidades sin modificar código existente.
3. Implementa y analiza algoritmos fundamentales —como recursión, métodos clásicos de ordenamiento y búsqueda— sobre listas y colecciones, con el objetivo de resolver problemas computacionales de manera eficiente en términos de tiempo de ejecución y uso de memoria.
4. Modela y resuelve problemas reales simples mediante la selección adecuada de estructuras de datos y estrategias de resolución, desarrollando soluciones contextualizadas y fundamentando las decisiones técnicas en el análisis de i-ciencia.
5. Participa en equipos de trabajo, planificando, coordinando y ejecutando tareas con liderazgo y responsabilidad, comunicándose efectivamente y elaborando informes técnicos que reflejen procedimientos, resultados y análisis del trabajo realizado.

**4. Unidades Temáticas:**

- **Unidad 1: Clases y objetos**
  - Definición, instanciación, atributos y métodos.
  - Encapsulamiento y abstracción.

- **Unidad 2: Herencia y polimorfismo**
  - Reutilización de código y extensión de funcionalidades.
  - Composición vs. herencia.
- **Unidad 3: Algoritmos y estructuras de datos**
  - Recursividad, ordenamientos (ej: Selection, Insertion, Merge Sort).
  - Algoritmos de Búsqueda (ej: secuencial y binaria )
  - Implementación sobre listas y colecciones.
- **Unidad 4: Modelado y eficiencia**
  - Resolución de problemas reales simples.
  - Análisis de eficiencia (tiempo y memoria).
  - Justificación de decisiones técnicas.

## 5. Descripción general del método de enseñanza:

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.

## 6. Descripción general de la modalidad de 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.

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

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.

## **7. Bibliografía Básica Obligatoria:**

1. Curso Intensivo de Python, Eric Matthes
2. Python para análisis de datos, Wes McKinney
3. Apuntes de Cátedra.