



## PROGRAMACIÓN II

### 1. OBJETIVOS

La asignatura tiene por objetivo brindar al alumno los conocimientos teóricos y prácticos necesarios para que pueda implementar (codificar) en un lenguaje orientado a objetos, lo cual incluye:

- Implementación de los conceptos fundamentales de la programación orientada a objetos (POO): clases, objetos, abstracción, encapsulamiento, relaciones entre clases, polimorfismo, interfaces, colecciones.
- Diseño e implementación básicos de una interfaz gráfica de usuario.
- Aplicación de los conceptos anteriores para la construcción de sistemas pequeños y medianos.

### 2. CONTENIDOS

#### 1) INTRODUCCIÓN

Historia de Java - Tipos de datos y métodos - Operadores - Sentencias de control - Paquetes - Funcionamiento - Herramientas - Diferencia entre POO y POP.

#### 2) CLASES Y OBJETOS

Clases: definición, métodos, atributos - Objetos - Vida de un objeto.

#### 3) ABSTRACCIÓN Y ENCAPSULAMIENTO

Abstracción y encapsulamiento – Sobrecarga.

#### 4) RELACIONES ENTRE CLASES

Relaciones entre clases: asociación, agregación y herencia.

#### 5) HERENCIA

Relación de herencia - Características - Redefinición de métodos - Polimorfismo - Comparación de objetos – Clases abstractas.

#### 6) MÉTODOS Y VARIABLES DE CLASE

Métodos de clase - Variables de clase.

#### 7) INTERFACES Y TIPOS GENÉRICOS

Interfaces - Características - Interfaz funcional – Interfaz vs. Clase abstracta - Introducción a los tipos genéricos.

#### 8) COLECCIONES

Colecciones - Instanciación de colecciones – Ordenamiento.

#### 9) MANEJO DE EXCEPCIONES

Manejo de excepciones - Múltiples excepciones - Propagación de excepciones.



10) PERSISTENCIA

Persistencia - Archivos de texto - Uso de buffers.

11) INTERFAZ GRÁFICA

Contenedores: marco, ventana de diálogo, panel - Componentes: etiqueta, campo de texto, botón, casilla de verificación, botón de opciones, lista, combo, tabla - Manejo de eventos - Patrón MVC.

3. **BIBLIOGRAFÍA**

- C. M. Fontela. Orientación a objetos con Java y UML. Nueva Librería. 2011.
- P. Deitel. Cómo programar en Java. Pearson Education. 2004.
- M. A. Weiss. Estructuras de datos en Java. Pearson Education. 2013.
- Z. J. Hernández Figueroa et al. Fundamentos de estructuras de datos, soluciones en Ada, Java y C++. Thomson. 2005.

4. **METODOLOGÍA DE ENSEÑANZA**

La asignatura se desarrolla mediante clases teóricas con ejemplos prácticos en aula, y trabajos prácticos en laboratorio. En los temas que correspondan se muestran distintas alternativas de soluciones, se analizan ventajas y desventajas de las mismas. Se aplican distintas metodologías (lectura de artículos, videos sobre alguna temática en particular, trabajos grupales, etc.) con la finalidad de fortalecer el aprendizaje por parte de los alumnos.

5. **EVALUACIÓN**

La signatura cuenta actualmente con los siguientes métodos de evaluación:

- Exámenes parciales prácticos (3).
- Evaluativos teóricos (10).
- Evaluación de trabajos prácticos.

Para poder regularizar la asignatura se debe:

- Asistir al 75% de las clases prácticas.
- Tener de promedio entre los 3 parciales nota mayor o igual a 4/10 (los 2 últimos con nota mayor o igual a 4/10). Para poder rendir el primer parcial se debe presentar y aprobar el trabajo práctico 4 y para el parcial 2 el práctico 7. El parcial 3 se trabaja y se rinde en forma grupal. Los prácticos no se recuperan.
- Si el promedio de los 3 parciales es menor a 4, o si el segundo o tercer parcial tiene por nota menos de 4, al finalizar la materia se puede recuperar integralmente (un único examen integral).

Para poder promocionar la asignatura se debe:

- Asistir al 75% de las clases prácticas.
- Tener de promedio entre los 3 parciales nota mayor o igual a 7/10 (los 2 últimos con nota mayor o igual a 4/10). Para poder rendir el primer parcial se debe presentar y aprobar el trabajo práctico 4



y para el parcial 2 el práctico 7. El parcial 3 se trabaja y se rinde en forma grupal. Los prácticos no se recuperan.

## 6. **CARGA HORARIA**

La asignatura pertenece al bloque “Tecnologías Básicas”, correspondiente al módulo IV del segundo año de la carrera de Ingeniería en Computación, con una carga horaria de 96 horas, distribuidas de la siguiente forma:

- Clases teóricas: 64 horas.
- Clases de formación práctica y laboratorios: 32 horas.
- Exámenes: 5 horas.

## 7. **OTRA INFORMACIÓN**

La cátedra tiene su sitio web (<https://catedras.facet.unt.edu.ar/prog2/>) en el cual se publican las pautas de cursado, fechas importantes, trabajos prácticos, clases teóricas y cualquier otro material de estudio que se considere necesario. Se publican las notas de los evaluativos y todas las novedades referidas a la materia.