



Departamento de Informática  
Facultad de Ingeniería  
Universidad Nacional de la Patagonia San Juan Bosco

PLANIFICACIÓN ANUAL  
PROGRAMACIÓN ORIENTADA A OBJETOS

AÑO 2024

## OBJETIVOS

- Dominar el Paradigma Orientado a Objetos independientemente del lenguaje de programación a utilizar.
- Dominar al menos un lenguaje de programación orientado a objetos de una forma eficiente para el desarrollo de aplicaciones.
- Disminuir la brecha existente entre el dominio del problema y el dominio de la aplicación.
- Desarrollar prácticas profesionales diseñando aplicaciones de mediana complejidad, teniendo en cuenta las etapas de la Ingeniería del Software en un nivel básico, como parte de la Instancia Supervisada de Formación Práctica Profesional (ISFPP), fortaleciendo sus competencias profesionales y trabajo en equipo.
- Conocer patrones de diseño fundamentales para garantizar la calidad en el desarrollo de sistemas bajo este paradigma.
- Introducir al alumno en los conceptos de concurrencia y paralelismo.

## CONTENIDOS MÍNIMOS

- Paradigma de programación orientada a objetos.
- Abstracción y encapsulamiento en programación orientada a objetos.
- Objetos. Clases e instancias.
- Herencia. Jerarquía de clases.
- Polimorfismo.
- Lenguajes y entornos de desarrollo.
- Interfaces gráficas. Eventos.
- Nociones de concurrencia y paralelismo.

## PROGRAMA ANALITICO

### **Unidad 1: Introducción al paradigma orientado a objetos.**

Que es un paradigma, comparación de paradigmas imperativos y procedural y la necesidad del paradigma orientado a objetos. Calidad del software, reducción de la brecha entre el mundo de los problemas y el mundo de los modelos. Aumento del nivel de complejidad de los sistemas. Reutilización y extensión del código. Uso de prototipos. Programación por eventos e interfaz gráfica del usuario. Lenguajes de programación orientados a objetos. Entornos de desarrollo.

### **Unidad 2: Paradigma orientado a objetos, clases y objetos.**

Evolución del concepto de tipo de datos abstracto. Abstracción. Objetos y clases. Estructura interna de un objeto. Responsabilidad de un objeto: métodos y mensajes. Encapsulamiento. Ocultamiento de la información. Protocolo. Encapsular atributos y comportamiento. Diseño de clases e Instanciación de objetos, constructores y destructores. Visibilidad de atributos. Lenguajes puros e híbridos.

### **Unidad 3: Herencia y Polimorfismo.**

Reutilización de código. Concepto de herencia. Herencia simple y múltiple. Clases abstractas y concretas. Generalización y Especialización. Interfaces. Métodos abstractos. Acoplamiento y cohesión.

Polimorfismo. Redefinición y reescritura de métodos. Información de tipos en tiempo de ejecución. Binding dinámico y estático.

#### **Unidad 4: Interfaz Gráfica de Usuario.**

Programación orientada a eventos, Interfaz gráfica del usuario (IGU) ventanas y componentes, Frames y Paneles rótulos y botones, componentes radio, combo y check. Listas y áreas de texto. Gestores de apariencia. Menús. Paletas. Manejo de eventos de mouse. Clases adaptadoras. Manejo de eventos de teclado. Diseño de Aplicaciones simples con IGU. Aplicaciones complejas con IGU, MVC y Bases de datos.

#### **Unidad 5: Uso de Patrones de Diseño en el POO.**

Patrones de diseño, clasificación, motivaciones y usos para los siguientes patrones: builder, factory method, singleton, Data Access Object, composite, decorator, facade, observer, state y strategy. Principios de diseño SOLID.

#### **Unidad 6: Concurrencia y Paralelismo.**

Introducción a la concurrencia, manejo de hilos, ejecución concurrente. Threads e Interfaz Runnable, Aplicaciones en animación y en planificación de procesos. Objetos activos y pasivos. Estado de un thread. Exclusión mutua y sincronización de hilos. Introducción al paralelismo, paralelismo de tareas y de datos. Métricas Speed-Up y Eficiencia, descomposición, asignación y comunicación de tareas. Aplicaciones del paralelismo.

## **PLANIFICACIÓN TEMPORAL**

Basados en el segundo cuatrimestre del año actual la planificación sería la siguiente:

Nº semana	Día	Fecha	Tema	
			Teoría	Práctica
1	L	05/08	Unidad 1	Practico 1
	M	07/08	Unidad 2	Practico 1
2	L	12/08	Unidad 2	Practico 2
	M	14/08	Unidad 2	Practico 2
3	L	19/08	Unidad 3	Practico 2
	M	21/08	Unidad 3	Practico 3
4	L	26/08	Unidad 3	Practico 3
	M	28/08	Unidad 4	Practico 3
5	L	02/09	Unidad 4	Consulta
	M	04/09	Primer parcial	
6	L	09/09	Unidad 4	Unidad 4
	M	11/09	Unidad 5	Unidad 4
7	L	16/09	Semana del Estudiante	
	M	18/09		

8	L	23/09	Unidad 5	Unidad 4
	M	25/09	Recuperatorio de primer parcial	
9	L	30/09	Unidad 5	Unidad 5
	M	02/10	Unidad 6	Unidad 5
10	L	07/10	Unidad 6	Unidad 5
	M	09/10	Unidad 6	Unidad 5
11	L	14/10	Consulta / practica	Unidad 6
	M	16/10	Consulta / practica	Unidad 6
12	L	21/10	Consulta / practica	Unidad 6
	M	23/10	Consulta / practica	Unidad 6
13	L	28/10	Consulta / practica	Consulta
	M	30/10	Segundo parcial	
14	L	04/11	Consulta / practica	Consulta / practica
	M	06/11	Recuperatorio segundo parcial / Promoción	
15	L	11/11	Consulta / practica	Consulta / practica
	M	13/11	Presentación trabajo final	
16	L	18/11	Feriado	
	M	20/11	Recuperatorio final	

**Teorías:** lunes 14 a 16 hs y miércoles de 17 a 19 hs.

**Prácticas:** lunes 19 a 21 hs y miércoles 19 a 21 hs.

**Aula:** Laboratorio Ardenghi.

## BIBLIOGRAFÍA

Título	Autor	Editorial	Año	Básica	Comp.
Python 3 Object-oriented Programming	Phillips	Packt	2015	1,2,3,5,6	
Python Object-Oriented Programming	Lott	Packt	2021	1,2,3,5,6	
An introduction to Object-Oriented Programming	Budd	Addison Wesley	2002	1,2,3,5	
Introducción a la programación orientada a objetos	Barraza	Pearson	2006	1,2,3	
Guide to Java A Concise Introduction	Streib	Springer	2023	1,2,3,6	
Fundamentals of OOP and Data Structures in Java	Wiener	Cambridge University Press	2000	1,2,3,4	
Beginning Java Objects From Concepts to Code	Barker	Apress	2023	1,2,3	
Learning Object-Oriented Programming	Hillar	Packt	2015	1,2,3	
Programación Orientada a Objetos	Aguilar	McGraw-Hill	1996	1,2,3	
OOP Concepts Booster	Singh	Notion Press	2019	1,2,3	

Oop - Learn Object Oriented Thinking and Programming	Pecinovský	Tomáš Bruckner	2013	1,2,3,5	
Python GUI Programming Cookbook Develop functional and responsive user interfaces with tkinter and PyQt5	Meier	Packt	2019	4	6
Mastering Design Patterns in Java	Norex	Independently publish	2024	5	
The Unified Modeling Language User Guide	Booch	Addison Wesley	1998	5	
Object Oriented Programming and Data Structures	Balagurusamy	McGraw-Hill	2014		1,2,3
Object-Oriented Python_ Master OOP by Building Games and GUIs	Kalb	No Starch	2022		1,2,3,4
Concepts and Semantics of Programming Languages 2 Modular and Object-oriented Constructs with OCaml, Python, C++, Ada and Java	Hardin	Wiley	2021		1,2,3
Learning Python	Lutz	O'Reilly	2013		1,2,3
Mastering Object-oriented Python	Lott	Packt	2014		2,3

## METODOLOGÍA DE ENSEÑANZA

Cada comienzo de nuevo práctico se iniciará con una revisión oral de los temas vistos previamente en teoría y como estos se relacionan con los ejercicios realizados en la práctica de las clases pasadas, se resolverán dudas del grupo en general en base a sus respuestas. Los nuevos temas serán introducidos con un ejemplo claro de cómo se realiza y cuál es el objetivo del mismo, se les facilitará videos explicativos con ejemplos prácticos y explicaciones de conceptos complementarios.

Durante la cursada se desarrollarán 6 trabajos prácticos, uno por cada unidad, cubriendo todos los contenidos mínimos. Además, se desarrollará un sistema completo de mediana complejidad, como actividad de Instancia Supervisada de Formación Práctica Profesional (ISFPP), en el cual se formarán grupos en función a la cantidad de alumnos que disponga la clase y se les asignará una aplicación que deberán ir resolviendo a medida que los trabajos prácticos y los contenidos teóricos avancen. El objetivo de la práctica profesional es lograr que los alumnos adquieran herramientas de trabajo en grupo y prepararlos para su desenvolvimiento en un ámbito profesional real.

## FORMAS DE EVALUACIÓN

**Condiciones para obtener el cursado de la Asignatura:**

Se tomarán dos parciales prácticos con su respectivo recuperatorio y recuperatorio final, en donde el alumno deberá demostrar que ha comprendido y asimilado los conceptos impartidos durante la cursada. El primer parcial consta de los contenidos vistos en las unidades 1,2 y 3. El segundo parcial abarcará las unidades 4, 5 y 6.

Los exámenes se deberán aprobar con una nota de 6 (seis) o superior y solo se podrá acceder al recuperatorio final si en los recuperatorios anteriores se obtuvo un 4 o 5. También deberán tener aprobada la Instancia Supervisada de Formación Práctica Profesional (ISFPP), en la cual evalúan todas las unidades

de la materia sobre un desarrollo de mediana complejidad. Al igual que los exámenes, esta instancia se aprueba con una nota mayor o igual a 6. La nota del concepto es un promedio entre los dos parciales aprobados y la ISFPP.

### Condiciones para obtener la aprobación de la Asignatura:

Se tomará un examen final teórico en las mesas finales propuestas por la Facultad de Ingeniería.

Los alumnos libres deberán contactar con el profesor responsable de la asignatura para que le asigne un trabajo final como parte de la actividad de la ISFPP. Una vez aprobada dicha práctica, podrá anotarse para rendir en una mesa de examen como alumno libre. En dicha mesa se evaluarán conceptos prácticos y teóricos de la asignatura.

Para obtener el régimen de promoción directa, el alumno deberá aprobar los dos parciales prácticos, en su primera instancia de evaluación, con una nota de 7 o superior. Aprobar la ISFPP y luego aprobar un único examen teórico integrador de promoción con nota 7 o superior.