

Plan de Estudios - Fundamentos de Programación

Sección C

Profesor: Diego Herbin Stalder Diaz

Profesora: Viviana Ortellado

Período: 6 de febrero al 22 de mayo, 2026

Información General

Campo	Información
Horario Teoría	07:30 - 09:30
Horario Práctica	09:30 - 11:30
Modalidad	Presencial - Viernes
Créditos	4
Duración	2h teoría + 2h práctica por clase

Calendario de Clases

Fecha	Semana	Tema Principal	Contenido Teórico	Contenido Práctico
6 febrero	1	Introducción y Fundamentos	<ul style="list-style-type: none">• Presentación del curso• Algoritmos y pseudocódigo• Python básico: instalación, IDE• Variables y operadores aritméticos	<ul style="list-style-type: none">• Primer programa en Python• Operaciones matemáticas básicas• VPL: ejercicios con variables• Conversión de tipos implícita
13 febrero	2	Tipos de Datos y Cadenas	<ul style="list-style-type: none">• Tipos de datos: int, float, str, bool• Métodos básicos de cadenas• Slicing de strings• Conversión explícita de tipos	<ul style="list-style-type: none">• Manipulación de strings• VPL: ejercicios con cadenas• Validación de formatos• Extracción de subcadenas
20 febrero	3	Control de Flujo	<ul style="list-style-type: none">• Operadores relacionales y lógicos• Estructuras if/elif/else	<ul style="list-style-type: none">• VPL: problemas condicionales• Validación de entradas• Ejercicios de decisión

Fecha	Semana	Tema Principal	Contenido Teórico	Contenido Práctico
			<ul style="list-style-type: none"> • Condicionales anidadas • Expresiones booleanas complejas 	<ul style="list-style-type: none"> • Depuración de código condicional
27 febrero	4	Listas y Bucles For	<ul style="list-style-type: none"> • Creación y acceso a listas • Índices y slicing • Métodos básicos de listas • Estructura for con range() • Iteración sobre listas 	<ul style="list-style-type: none"> • Manipulación de listas • VPL: ejercicios con for • Procesamiento de secuencias • Cálculos con listas numéricas
6 marzo	5	Bucles While y Diccionarios	<ul style="list-style-type: none"> • Estructura while • Contadores y acumuladores • break/continue • Validación con while • Diccionarios básicos 	<ul style="list-style-type: none"> • VPL: ejercicios con while • Menús interactivos • Validación robusta • Manipulación de diccionarios • Comparación for vs while
13 marzo	6	Funciones y Módulos	<ul style="list-style-type: none"> • Definición de funciones (def) • Parámetros y valores de retorno • Ámbito de variables • Módulos: import/from • math, random, datetime 	<ul style="list-style-type: none"> • VPL: creación de funciones • Funciones matemáticas • Creación de módulo propio • Uso de librerías estándar • Ejercicios integradores
20 marzo	--	PRIMER EXAMEN PARCIAL	Cobertura: Semanas 1-6 (hasta funciones y módulos)	Evaluación: Ejercicios de programación aplicando todos los conceptos vistos
27 marzo	--			
3 abril	--	SEMANA SANTA NO HAY CLASES	--	--
10 abril	7	Recursividad	<ul style="list-style-type: none"> • Concepto de recursión • Caso base vs caso recursivo • Factorial y Fibonacci 	<ul style="list-style-type: none"> • VPL: ejercicios recursivos • Implementación funciones recursivas • Análisis de pila de llamadas • Optimización básica

Fecha	Semana	Tema Principal	Contenido Teórico	Contenido Práctico
			<ul style="list-style-type: none"> • Recursión vs iteración • Aplicaciones prácticas 	<ul style="list-style-type: none"> • Presentación proyecto desafío
17 abril	8	Búsqueda y Ordenamiento Básico	<ul style="list-style-type: none"> • Búsqueda lineal $O(n)$ • Búsqueda binaria $O(\log n)$ • Ordenamiento por selección • Comparación de eficiencia • Análisis de complejidad 	<ul style="list-style-type: none"> • Implementación de búsquedas • Implementación selection sort • VPL: comparación de algoritmos • Análisis de tiempos de ejecución • Avance del proyecto
24 abril	9	Ordenamiento Avanzado	<ul style="list-style-type: none"> • Ordenamiento por inserción • Quicksort (divide y vencerás) • Mergesort (concepto) • Comparación de métodos • Estabilidad y casos óptimos 	<ul style="list-style-type: none"> • Implementación insertion sort • Implementación quicksort básico • VPL: algoritmos avanzados • Optimización de código • Trabajo aplicado al proyecto
1 mayo	--	FERIADO Día del Trabajador	--	--
8 mayo	10	NumPy para Ingeniería	<ul style="list-style-type: none"> • Arrays vs listas • Creación de arrays • Operaciones vectorizadas • Álgebra lineal básica • Procesamiento eficiente 	<ul style="list-style-type: none"> • VPL: ejercicios con NumPy • Operaciones matriciales • Cálculos científicos • Integración con Python puro • Optimización en el proyecto
15 mayo	--	FERIADO Independencia Nacional	--	--
22 mayo	11	Visualización y Cierre	<ul style="list-style-type: none"> • Matplotlib básico • Gráficos 2D: líneas, barras, scatter • Visualización de datos • Personalización de 	<ul style="list-style-type: none"> • VPL: creación de visualizaciones • Gráficos de datos del proyecto • Presentación de resultados

Fecha	Semana	Tema Principal	Contenido Teórico	Contenido Práctico
			gráficos • Exportación para informes	• Cierre y presentación proyecto

Desarrollo del Proyecto

10 abril	Presentación y Planificación	<ul style="list-style-type: none"> Definición del problema Propuesta de solución Formación de grupos (3-4 personas) Documento de requisitos 	5%
17 abril	Fase 1: Modelado de Datos	<ul style="list-style-type: none"> Estructura de datos definida Algoritmos de búsqueda implementados Datos de prueba creados Validación básica 	15%
24 abril	Fase 2: Implementación de Algoritmos	<ul style="list-style-type: none"> Funciones modularizadas Algoritmos de ordenamiento Pruebas unitarias Documentación de código 	20%
8 mayo	Fase 3: Visualización y Optimización	<ul style="list-style-type: none"> Visualización con Matplotlib Interfaz de usuario (CLI o básica GUI) Optimización con NumPy Reporte de resultados 	25%
22 mayo	Presentación Final	<ul style="list-style-type: none"> Video demostrativo (3-5 min) Código fuente completo Presentación oral Documentación final 	35%

Observaciones

1. **VPL** = Virtual Programming Lab (plataforma en línea para ejercicios)
2. **Proyecto** se desarrolla en grupos de 3-4 estudiantes
3. **Temas de proyecto** adaptados a cada especialidad de ingeniería
4. **Distribución horaria:** Cada clase incluye 2 horas de teoría (07:30-09:30) y 2 horas de práctica (09:30-11:30)
5. **Metodología:** Enfoque práctico con ejercicios aplicados a problemas de ingeniería

Recursos y Materiales

- **Plataforma principal:** Virtual Programming Lab (VPL) para ejercicios
- **Bibliotecas Python:** NumPy, Matplotlib
- **Entornos recomendados:** VSCode, Jupyter Notebook, Google Colab
- **Material de apoyo:** Diapositivas, códigos de ejemplo, guías de laboratorio