**Paradigmas de Programación (UNPAZ**) Plan de la asignatura - 2025

Duración: 16 semanas – 4 horas cada clase

Modalidad: Teórico-práctica (20% teoría – 80% práctica)

Evaluación: 2 exámenes parciales + Proyecto Final

Docentes: Alvarez Leandro.

## Distribución del Cuatrimestre por semanas

## Semana 1 – Introducción + Expresiones, Funciones y Arrays

Contenidos:

* - Instalación/configuración Node.js y VS Code.
* - Creación y buenas prácticas en la creación de un nuevo proyecto.
* - Introducción al paradigma de programación, imperativo vs declarativo.
* - Expresiones y evaluación.
* - Funciones: de primera clase, funciones de orden superior (ciudadanas), anónimas, funciones como parámetros, funciones que retornan funciones. Concepto de recursividad.
* - Introducción a las estructuras de datos Arrays: Inmutabilidad, includes, some, every, map, reduce, sort. Creación de una estructura del tipo objeto.

Prácticas:

* - Buscador de productos con filtros múltiples.
* - Cálculo de estadísticas de un arreglo.

## Semana 2 – Arrays avanzados

Contenidos:

* - Uso combinado de map, filter, reduce, sort.
* - Arrays de objetos y procesamiento de datos.
* - Funciones flecha y encadenamiento de métodos.

Prácticas:

* - Listado de estudiantes con promedios y ranking.
* - Filtrado y ordenamiento de inventario.

## Semana 3 – Objetos y manipulación de datos

Contenidos:

* - Creación y modificación de objetos.
* - Acceso y desestructuración.
* - Objetos anidados y arrays de objetos.
* - Operador spread/rest.

Prácticas:

* - Sistema de gestión de clientes.
* - Procesar pedidos con información anidada.

## Semana 4 – Manejo de archivos (Node.js)

Contenidos:

* - Módulo fs: lectura y escritura síncrona y asíncrona.
* - JSON: parseo y serialización.
* - Validaciones y manejo de errores.

Prácticas:

* - Guardar y cargar listado de tareas desde archivo.
* - Exportar datos procesados a JSON.

## Semana 5 – Funcional avanzada

Contenidos:

* - Funciones puras e inmutabilidad.
* - Composición de funciones.
* - Currying y partial application.

Prácticas:

* - Pipeline de procesamiento de datos.
* - Funciones parametrizadas para cálculos.

## Semana 6 – POO I: Clases y herencia

Contenidos:

* - Clases, constructores, propiedades y métodos.
* - Herencia con extends y super().
* - Métodos estáticos.

Prácticas:

* - Jerarquía de clases para gestión de vehículos.
* - Métodos comunes y especializados.

## Semana 7 – POO II: Encapsulamiento y polimorfismo

Contenidos:

* - Encapsulamiento con campos privados (#).
* - Getters y setters.
* - Polimorfismo y sobreescritura.

Prácticas:

* - Sistema de pagos con distintos métodos.
* - Gestión de acceso a datos sensibles.

## Semana 8 – Primer Examen Parcial

Contenidos:

* - Evaluación teórico-práctica.
* - Ejercicios con funciones, arrays, objetos y manejo básico de archivos.

## Semana 9 – Modularidad y organización del código

Contenidos:

* - ES Modules (import, export).
* - CommonJS.
* - Estructura de carpetas y responsabilidad única.

Prácticas:

* - Proyecto dividido en módulos.
* - Implementar utilidades reutilizables.

## Semana 10 – Estructuras de datos avanzadas

Contenidos:

* - Map, Set, WeakMap, WeakSet.
* - Operaciones útiles y casos de uso.

Prácticas:

* - Agenda de contactos con Map.
* - Registro único de elementos con Set.

## Semana 11 – Algoritmos y complejidad

Contenidos:

* - Búsqueda lineal y binaria.
* - Ordenamiento (burbuja, selección, inserción).
* - Introducción a Big O.

Prácticas:

* - Comparar rendimientos entre métodos.
* - Implementar buscadores optimizados.

## Semana 12 – Algoritmos avanzados

Contenidos:

* - QuickSort, MergeSort.
* - Recursividad y memoización.

Prácticas:

* - Implementar QuickSort para ordenar grandes volúmenes.
* - Fibonacci optimizado.

## Semana 13 – Programación asíncrona I

Contenidos:

* - Callbacks y su problema.
* - Promesas (Promise).
* - async/await.

Prácticas:

* - Simular consultas a una API con setTimeout.
* - Procesar datos en paralelo con Promise.all.

## Semana 14 – Programación asíncrona II + Fetch API

Contenidos:

* - fetch() para obtener datos.
* - Manejo de errores y respuestas.
* - JSON en asincronismo.

Prácticas:

* - Consumir API pública.
* - Mostrar y filtrar resultados.

## Semana 15 – Segundo Examen Parcial

Contenidos:

* - Evaluación teórico-práctica.
* - Incluye POO, manejo de archivos, modularidad, asincronismo y algoritmos.

## Semana 16 – Recuperatorio final y cierre de calificaciones

Contenidos:

* - Sistema de gestión integrando todos los temas.
* - Defensa y exposición del proyecto.
* - Retroalimentación final.