

Proyecto Práctico: Sistema de Gestión de Datos con Estructuras Avanzadas

Objetivo:

Desarrollar un sistema de gestión de datos que aplique los modelos de programación y estructuras de datos aprendidos durante el ciclo, resolviendo un problema real o simulado. El proyecto integrará conceptos de programación estructurada, algoritmos de ordenación/búsqueda, estructuras lineales y no lineales, y evaluación de complejidad computacional.

Descripción del Proyecto

Cada grupo diseñará e implementará un sistema que maneje datos complejos, como:

- **Base de datos de estudiantes** (registros, búsquedas eficientes, estadísticas).
- **Simulador de redes sociales** (grafos para relaciones, árboles para jerarquías).
- **Sistema de reservas** (colas para prioridades, pilas para historial).

Requisitos:

1. Fase 1 (Unidad 1-2):

- Diseñar el esquema modular del sistema (funciones, procedimientos, paso de parámetros).
- Implementar algoritmos de ordenación (burbuja) y búsqueda (binaria/secuencial) para procesar datos.

2. Fase 2 (Unidad 3):

- Usar estructuras lineales (listas ligadas, pilas, colas) para almacenar y gestionar datos dinámicos.
- Ejemplo: Cola para tickets de soporte, lista doblemente ligada para historial de acciones.

3. Fase 3 (Unidad 4):

- Aplicar estructuras no lineales (árboles para categorización, grafos para relaciones complejas).
- Ejemplo: Árbol binario para búsqueda rápida, grafo para conexiones entre usuarios.

4. Evaluación de eficiencia:

- Analizar la complejidad computacional de las operaciones clave (ej: búsqueda en árbol vs. lista).

Entregables

1. **Informes parciales:** Documentación técnica por cada fase (diagramas, explicación de estructuras usadas).
2. **Código fuente:** Implementación en lenguaje elegido (C++, Java, Python, etc.).
3. **Presentación final:** Demostración del sistema y defensa de las decisiones técnicas.

Cronograma

Semana	Actividad
1-4	Diseño modular y algoritmos básicos (Fase 1).
5-8	Implementación de estructuras lineales (Fase 2).
9-12	Integración de estructuras no lineales (Fase 3).
13-16	Optimización, pruebas y presentación final.

Evaluación

- **30%** Funcionalidad (correcta implementación de estructuras).
- **25%** Documentación (claridad, justificación técnica).
- **25%** Creatividad (solución innovadora al problema).
- **20%** Presentación (exposición y defensa del proyecto).

Recursos Sugeridos

- Lenguajes: Python (para prototipado rápido) o C++ (para manejo de memoria).
- Bibliografía: "*Fundamentos de programación*" (Joyanes) para algoritmos; "*Programación orientada a objetos con C++*" (Ceballos) para diseño modular.
- Herramientas: Git para control de versiones, Trello para gestión de tareas.

Nota: Los grupos pueden elegir su propio problema a resolver, sujeto a aprobación del docente. El proyecto fomentará el trabajo colaborativo y la aplicación integral de los contenidos del módulo.