



Cronograma por etapas

✓ Etapa 1: Planificación y estructura base

Objetivo: Armar la base del proyecto.

1. Crear el repositorio en GitHub (público).
2. Definir qué structs vas a usar:
 - `Producto` (nombre, precio, stock).
 - `Venta` (nombre producto, cantidad, total).
3. Crear un archivo `.c` principal con `main()` y estructura básica del menú.
4. Crear el README inicial con:
 - Nombre del proyecto.
 - Miembros del grupo.
 - Breve descripción.

👉 **Resultado esperado:** Proyecto creado y menu inicial esqueleto en consola.

✓ Etapa 2: Gestión de productos

Objetivo: Crear el sistema con arreglo de productos.

1. Declarar arreglo de productos.
2. Función para cargar productos (manual o desde archivo).
3. Función para mostrar productos disponibles.
4. Función para editar y actualizar stock.

👉 **Resultado esperado:** Menú funcional que muestra y actualiza productos.

✓ Etapa 3: Registro de ventas con lista dinámica

Objetivo: Implementar lista dinámica para ventas del día.

1. Definir struct nodo de venta y punteros (lista enlazada).
2. Función para registrar una venta (buscar producto, actualizar stock, agregar nodo a lista).
3. Función para mostrar ventas del día.

👉 **Resultado esperado:** Se puede registrar una venta y se lista correctamente.

✓ Etapa 4: Cálculos y operaciones

Objetivo: Cálculo del total vendido y búsqueda.

1. Función para calcular el total vendido del día.
2. Función de búsqueda de producto por nombre (puede usarse también para editar).
3. Agregar ordenamiento de productos por nombre o precio.

👉 **Resultado esperado:** Funcionalidades extra que enriquecen el sistema.

✓ **Etapa 5: Archivos**

Objetivo: Lectura y escritura de archivos.

1. Guardar ventas del día en archivo (`ventas.txt`).
2. Cargar productos desde archivo (`productos.txt`) al iniciar.
3. Guardar productos actualizados al cerrar.

👉 **Resultado esperado:** Datos persistentes entre ejecuciones.

✓ **Etapa 6: Modularización y documentación**

Objetivo: Dejar el código ordenado y documentado.

1. Separar funciones en archivos `.h` y `.c` si es necesario.
2. Agregar comentarios claros y en español.
3. Completar el README con:
 - Cómo compilar y correr.
 - Roles de cada integrante.
 - Explicación de las funciones principales.

👉 **Resultado esperado:** Proyecto presentable, claro y funcional.