

## Nombre del Proyecto

"Gestión Simplificada de Tareas Corporativas"

## Estructura del Proyecto y Grupos

### Grupo 1: Gestión de Usuarios

#### Tareas:

1. Crear un sistema para registrar, editar y eliminar usuarios utilizando listas de Python.
2. Validar que el nombre de usuario no esté vacío y que el correo tenga un formato correcto (por ejemplo, contenga "@" y un dominio).
3. Implementar una opción para listar a todos los usuarios registrados.

#### Elementos:

- Funciones para agregar, editar y eliminar usuarios.
  - Bucles para recorrer y listar usuarios.
  - Validaciones para la entrada de datos.
- 

### Grupo 2: Gestión de Tareas

#### Tareas:

1. Diseñar un módulo para asignar tareas a usuarios.
2. Implementar la funcionalidad para marcar tareas como "pendiente" o "completada".
3. Permitir que se listen las tareas de un usuario específico.

#### Elementos:

- Listas y diccionarios para almacenar usuarios y sus tareas.
  - Condicionales para cambiar el estado de las tareas.
  - Funciones para asignar, actualizar y listar tareas.
- 

### Grupo 3: Gestión de Eventos

#### Tareas:

1. Crear eventos con atributos como nombre, fecha y capacidad máxima.
2. Permitir registrar usuarios a eventos mientras haya cupo.
3. Implementar la funcionalidad para cancelar registros de usuarios a eventos.

#### Elementos:

- Bucles para verificar disponibilidad y registrar usuarios.
  - Condicionales para validar que no se exceda la capacidad del evento.
  - Funciones para crear eventos, registrar y cancelar inscripciones.
-

## Grupo 4: Estadísticas del Sistema

### Tareas:

1. Contar el número total de usuarios, tareas y eventos registrados.
2. Generar un resumen de estadísticas (por ejemplo, porcentaje de tareas completadas).
3. Mostrar la cantidad de eventos en los que cada usuario está inscrito.

### Elementos:

- Bucles para recorrer listas y contar elementos.
  - Funciones para calcular estadísticas.
  - Validaciones para evitar divisiones por cero u otros errores.
- 

## Grupo 5: Seguridad y Validaciones

### Tareas:

1. Validar que los datos ingresados sean correctos (por ejemplo, que no haya nombres vacíos o fechas en formato incorrecto).
2. Implementar un sistema de autenticación básico con usuario y contraseña.
3. Registrar actividades del sistema (por ejemplo, usuarios que inician sesión o errores detectados) en un archivo de texto (logs.txt).

### Elementos:

- Condicionales para verificar datos.
  - Uso de funciones para autenticar usuarios y registrar actividades.
  - Manejo de archivos (with open) para guardar logs.
- 

## Grupo 6: Integración del Sistema

### Tareas:

1. Coordinar la integración de todos los módulos.
2. Crear un menú principal que permita navegar entre los módulos usando un bucle principal.
3. Validar entradas del usuario para evitar errores (por ejemplo, opción no válida).

### Elementos:

- Bucles para un menú interactivo.
- Funciones para enlazar los módulos.
- Condicionales para verificar la selección del usuario.

---

## Grupo 7: Simulación de Flujo de Trabajo

### Tareas:

1. Diseñar un flujo que incluya:
  - Crear un usuario.
  - Asignarle tareas.
  - Inscribirlo en eventos.
2. Generar un reporte en consola que muestre:
  - Nombre del usuario.
  - Número de tareas completadas/pendientes.
  - Eventos a los que está registrado.
3. Validar que el flujo funcione correctamente y sea robusto ante errores.

### Elementos:

- Condicionales para manejar decisiones y validaciones.
- Bucles para recorrer listas y mostrar datos.
- Funciones para mantener el código modular y organizado.