

# Análisis del Escenario de Recolección de Datos y Diseño del Sistema de Gestión de Datos

## Análisis del Escenario

### Descripción

El sistema gestiona la información de candidatos políticos y votos en una elección. Permite ingresar candidatos, registrar votos y mostrar resultados.

### Datos Requeridos

#### 1. Candidatos:

- **ID:** Identificador único.
- **Nombre:** Nombre del candidato.
- **Partido:** Partido político.
- **Plataforma:** Descripción breve.
- **Votos:** Cantidad de votos (inicialmente 0).

#### 2. Votos:

- **ID del Candidato:** Identificador del candidato.
- **Número de Votos:** Cantidad de votos registrados.

### Procedimientos

- **Ingreso de Candidatos:** Introducción de datos del candidato.
- **Registro de Votos:** Asignación de votos a un candidato.
- **Visualización de Resultados:** Cálculo y presentación de votos y porcentajes.

## Diseño del Sistema

### Requisitos

- **Interfaz:** Línea de comandos.
- **Almacenamiento:** Estructuras de datos en memoria.
- **Cálculo:** Total y porcentaje de votos.

### Estructuras de Datos

- **Candidato:**
  - **ID (int)**
  - **Nombre (string)**
  - **Partido (string)**
  - **Plataforma (string)**
  - **Votos (int)**

- **Contenedor:**
  - **Vector** `vector<Candidato>`: Almacena candidatos.

## Funciones

1. **ingresarCandidatos:** Agrega nuevos candidatos.
2. **registrarVotos:** Registra votos para un candidato.
3. **verResultados:** Muestra resultados de la elección.
4. **mostrarMenu:** Presenta opciones al usuario.

## Flujo del Programa

1. **Inicio:** Muestra menú.
2. **Selección:** Usuario elige opción.
3. **Ejecución:** Realiza acción según opción.
4. **Finalización:** Sale del programa si se selecciona "Salir".

## Consideraciones

- **Validación:** Verifica existencia del ID al registrar votos.
- **Errores:** Mensajes claros para entradas inválidas.