**Documento de análisis del escenario de recolección de datos y el diseño del**

**1. Análisis del Escenario de Recolección de Datos**

**Requisitos**

1. **Registrar Candidatos**:
   * El programa permite la entrada de datos personales de los candidatos, como su nombre, partido político y plataforma.
   * Asegura que los datos ingresados no están vacíos.
2. **Registrar Votos**:
   * El programa permite a los usuarios votar por los candidatos registrados.
   * Valida votos no válidos (por ejemplo, votar por un candidato inexistente) y nos los permite.
3. **Diseño del Sistema de Gestión de Datos**

**Estructuras de Datos**

Para la creación de programa, se utiliza estructuras de datos sencillos que permiten almacenar y manipular la información necesaria:

1. **Estructura Candidato**:
   * nombre (std::string): Nombre del candidato.
   * partidoPolitico (std::string): Partido político del candidato.
   * votos (int): Contador de votos del candidato.
2. **Contenedor de Candidatos**:
   * Se utilizará un std::vector para almacenar una lista de candidatos. El vector permite un acceso eficiente y dinámico a los elementos.

**Funciones del Sistema**

1. **Registrar Candidato**:
   * Permite agregar nuevos candidatos al vector.
2. **Registrar Voto**:
   * Muestra la lista de candidatos y permite al usuario seleccionar uno para votar.
   * Incrementa el contador de votos del candidato seleccionado.
   * Valida que la opción ingresada sea válida.
3. **Mostrar Resultados**:
   * Calcula el total de votos para determinar el porcentaje de cada candidato.
   * Identifica al candidato con más votos y lo declara ganador.
   * Presenta los resultados de manera clara y concisa.

**Diagrama de Flujo del Sistema**

Diagrama

Descripción generada automáticamente

**3. Creación del programa**

#include <iostream>

#include <vector>

#include <string>

#include <iomanip>

#include <algorithm>

struct Candidato {

std::string nombre;

std::string partidoPolitico;

int votos;

};

std::vector<Candidato> candidatos;

void registrarCandidato() {

Candidato candidato;

std::cin.ignore();

std::cout << "Ingrese el nombre del candidato: ";

std::getline(std::cin, candidato.nombre);

std::cout << "Ingrese el partido politico del candidato: ";

std::getline(std::cin, candidato.partidoPolitico);

candidato.votos = 0;

candidatos.push\_back(candidato);

std::cout << "Candidato registrado exitosamente.\n";

}

void votar() {

if (candidatos.empty()) {

std::cout << "No hay candidatos registrados.\n";

return;

}

std::cout << "Lista de candidatos:\n";

for (size\_t i = 0; i < candidatos.size(); ++i) {

std::cout << i + 1 << ". " << candidatos[i].nombre << " (" << candidatos[i].partidoPolitico << ")\n";

}

std::cout << "Ingrese el numero del candidato para votar: ";

size\_t opcion;

std::cin >> opcion;

if (opcion > 0 && opcion <= candidatos.size()) {

candidatos[opcion - 1].votos++;

std::cout << "Voto registrado exitosamente.\n";

} else {

std::cout << "Opcion invalida.\n";

}

}

void mostrarResultados() {

if (candidatos.empty()) {

std::cout << "No hay candidatos registrados.\n";

return;

}

int totalVotos = 0;

for (const auto& candidato : candidatos) {

totalVotos += candidato.votos;

}

std::cout << "Resultados de la eleccion:\n";

for (const auto& candidato : candidatos) {

double porcentaje = (totalVotos > 0) ? (static\_cast<double>(candidato.votos) / totalVotos) \* 100 : 0.0;

std::cout << candidato.nombre << " (" << candidato.partidoPolitico << ") - "

<< candidato.votos << " votos (" << std::fixed << std::setprecision(2) << porcentaje << "%)\n";

}

auto ganador = std::max\_element(candidatos.begin(), candidatos.end(), [](const Candidato& a, const Candidato& b) {

return a.votos < b.votos;

});

if (ganador != candidatos.end()) {

std::cout << "El ganador es: " << ganador->nombre << " con " << ganador->votos << " votos.\n";

}

}

void mostrarMenu() {

std::cout << "\nSistema de Votaciones Electronico\n";

std::cout << "1. Ingresar Candidatos\n";

std::cout << "2. Votaciones\n";

std::cout << "3. Resultados\n";

std::cout << "4. Salir\n";

std::cout << "Ingrese su opicon: ";

}

int main() {

int opcion;

do {

mostrarMenu();

std::cin >> opcion;

switch (opcion) {

case 1:

registrarCandidato();

break;

case 2:

votar();

break;

case 3:

mostrarResultados();

break;

case 4:

std::cout << "Saliendo...\n";

break;

default:

std::cout << " Intente de nuevo.\n";

}

} while (opcion != 4);

return 0;

}

**Conclusión**

El documento contiene un análisis detallado y un diseño del un programa de votaciones electrónico utilizando C++. El sistema es sencillo y cumple con los requisitos especificados, proporcionando una solución funcional y fácil de usar para gestionar elecciones locales sin la necesidad de una base de datos.