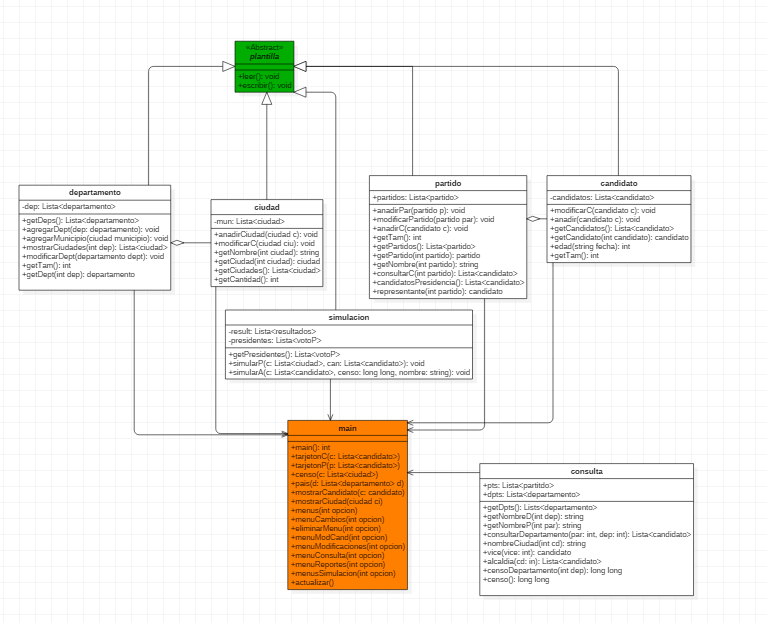
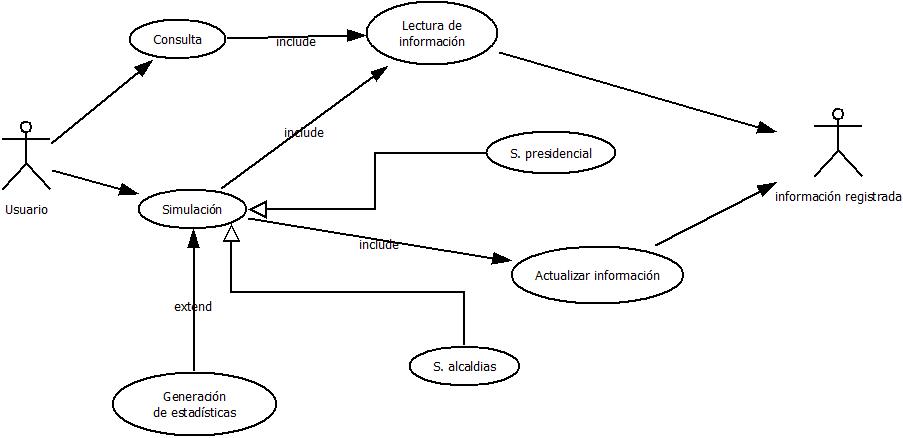
**DOCUMENTO DE SOPORTE**

1. **DISEÑO DE LA SOLUCIÓN**

* **Diagrama de clases**



* **Diagrama de casos de uso**

****

1. **Contenido y estructura de archivos planos.**

* **Archivo candidatos:**

En este archivo se guardan todos los datos correspondientes a un candidato, cada candidato tiene asignado una clave, la cual se utilizará, en tiempo de ejecución para buscarlo.

Cada candidato tiene asignado un partido, una ciudad de residencia, una ciudad de nacimiento, un tipo de candidato (alcalde, presidente, vicepresidente o representante) y una formula vicepresidencia; los cuales están representados como enteros.

Estructura:

[Clave]- nombre- apellido – cedula – sexo – estado civil – fecha de nacimiento- ciudad de nacimiento – ciudad de residencia – partido político – tipo candidato – formula vicepresidencia.

* **Archivo ciudades:**

Este archivo contiene toda la información referente a una ciudad, en el cual se tiene el censo que dicha ciudad posee, el departamento al cual pertenece (representado con enteros), una clave única por ciudad y el estado de la ciudad, es decir, si está habilitada o no para la simulación.

Estructura:

[Clave] – Nombre de ciudad – departamento – censo – estado.

* **Archivo departamentos:**

Este archivo contiene la información de los departamentos habilitados para la simulación, cuenta con una clave única por departamento y el nombre del departamento.

Estructura:

[Clave] – Nombre del departamento.

* **Archivo partidos:**

Este archivo contiene la información de los partidos legalmente registrados para la simulación, cuenta con una clave única por partido, el nombre del partido y el estado actual del partido.

Estructura:

[Clave] – Nombre del partido – estado.

1. **Estructuras definidas en memoria principal.**

* **Estructura candidato:**

Esta estructura cuenta con los string nombre, apellido, id (cedula), sexo, estadoCivil y fechaNac, en los cuales se guardarán los datos homónimos que se han leído del archivo plano; además, cuenta con los enteros ciudadNac, ciudadRes, partido, tipoCandidato, clave y vicepresidente (el cual contiene la clave de la formula vicepresidencial, si la tiene).

struct candidato{

string nombre;

string apellido;

string id;

string sexo;

string estadoCivil;

string fechaNac;

int ciudadNac;

int ciudadRes;

int partido;

int tipoCandidato;

int vicepresidente;

int clave;

};

* **Estructura ciudad:**

Esta estructura cuenta con el string nombre, en el cual se guardará el nombre de la ciudad, además cuenta con los enteros clave, departamento, habilitada, en los cuales se almacenará los datos correspondientes a la clave de la ciudad, el departamento perteneciente y el estado actual de la ciudad, respectivamente. Finalmente cuenta con un long long el cual permitirá manejar el censo de cada ciudad.

struct ciudad{

int clave;

string nombre;

int departamento;

long long censo;

int habilitada;

};

* **Estructura departamento:**

Esta estructura está compuesta por un entero, en el cual se almacena la clave única por departamento, un string nombre, en el que estará el nombre del departamento y finalmente una Lista<ciudad> en la cual estarán todas las ciudades que pertenecen a dicho departamento, además de un long long, en el que estará el censo total del departamento (suma de los censos de todas las ciudades).

struct departamento{

//El identificador del dept es la posicion en el array

int clave;

string nombre;

Lista<ciudad> municipios; //Tiene una lista de ciudades

long long censoPersonas; //Cantidad de personas que pueden votar en el departamento

};

* **Estructura partido:**

Esta estructura cuenta con la clave, de tipo entero, que representa la clave única del partido, un string nombre, que almacena el nombre del partido y finalmente una Lista<candidato> que contiene la lista de todos los candidatos pertenecientes a dicho partido. Ade,ás posee un int que indica el estado del partido.

struct partido{

int clave;

string nombre;

Lista<candidato> candidatos;

int habilitado;

};

* **Estructura votoPC:**

Esta estructura es una estructura auxiliar, la cual ayuda a controlar los votos que recibe un candidato presidencial por ciudad.  
Cuenta con un long long que almacena los voto que obtuvo el candidato en la ciudad, un float que representa el porcentaje del candidato en la ciudad, un entero que tiene el índice de la ciudad en la cual se realiza dicho estudio y otro entero con el número del departamento en cuestión.

struct votoPC{

long long votos;

float porcentaje;

int ciudad;

int departamento;

};

* **Estructura votoEst:**

Esta estructura guarda el reportaje general de un departamento, ya sea de presidencia o alcaldía, guardando la cantidad de votos, votos en blanco, votos nulos y abstención que se tuvo, con sus respectivos porcentajes.

struct votoEst{

long long votos;

long long votosBlanco;

long long votosAbstencion;

long long votosNulos;

float porcentajeBlanco;

float porcentajeAbst;

float porcentajeNulos;

int departamento;

};

* **Estructura estadística:**

La estructura estadística, guarda el reportaje detallado en una votación, por partido, contando la cantidad hombres, mujeres y candidatos que ganaron en las votaciones de alcaldía.

struct estadisticas{

int partido;

int cantidadHombres;

int cantidadMujeres;

int totalCandidatos;

float porcentajeHombres;

float porcentajeMujeres;

float porcentajeTotal;

};

* **Estructura estadisticasP:**

La estructura estadisticasP guarda el reportaje detallado por partido en las votaciones de presidencia. Tiene la cantidad de votos que se hicieron hacia hombres del partido, votos que se hicieron hacia mujeres del partido, con sus porcentajes, respecto al total de votos de presidencia. Tambien almacena el porcentaje total de votos por el partido.

struct estadisticasP{

int partido;

long long votosPorHombres;

long long votosPorMujeres;

long long totalVotos;

float porcentajeTotal;

float porcentajeHombres;

float porcentajeMujeres;

};

* **Estructura voto:**

Esta estructura se emplea para manejar los votos recibidos por parte de un alcalde en tiempo de ejecución.  
Cuenta con un string que contiene el nombre del candidato, un entero que representa el partido político al cual pertenece, un long long que representa la cantidad de votos obtenidos por el candidato y un float que es el porcentaje que dichos votos significan.

struct voto{

string nombreCand;

int partido;

long long votos;

float porcentaje;

};

* **Estructura votoP:**

Esta estructura sirve para manejar los resultados de los candidatos presidenciales, relaciona un candidato, con su vicepresidente y con los votos que recibió en cada ciudad.

El string nombreCand guarda el nombre del candidato, tiene un int que guarda el nuero del partido y otro en el cual representa la clave de su formula vicepresidencial además de un long long que es la suma total de todos los votos obtenidos a nivel nacional, con un float que es el porcentaje que representan dichos votos, y finalmente con un apuntador de tipo votoPC que contiene los resultados por ciudad.

struct votoP{

string nombreCand;

int partido;

int vice;

long long votos;

float porcentaje;

votoPC \*votoP;

};

* **Estructura resultados:**

En esta estructura se guardan los resultados por ciudad de todos los candidatos, cuenta con un string nombreCiudad que tiene el nombre de la ciudad en cuestión, un int que tiene el número del departamento al que pertenece, un int con la clave del candidato ganador a la alcaldía de dicha ciudad y una Lista<voto> que contiene los resultados de todos los candidatos en esa ciudad.

struct resultados{

string nombreCiudad;

int departamento;

int ganador;

Lista<voto> resultado;

};