

ESCUELA DE
INGENIERÍA INFORMÁTICA



PONTIFICIA
UNIVERSIDAD
CATÓLICA DE
VALPARAÍSO

EVALUACION MODULAR 3 Y 4 EL PROCESO ELECTORAL PRESIDENCIAL CHILENO

Franco Allendes
Francisco Ceballos
Alonso Diaz

Estructura De Datos INF- 2223

Noviembre 2025

ÍNDICE GENERAL

ÍNDICE GENERAL	i
LISTA DE FIGURAS	ii
1. INTRODUCCION	1
2. DOMINIO DEL PROBLEMA	2
3. ANÁLISIS.....	3
3.1. Estructura sistema electoral	3
3.2. Estructura elección.....	3
3.3. Estructura votante	3
3.4. Estructura mesa.....	4
3.5. Estructura candidato.....	4
3.6. Estructura voto.....	4
4. DISEÑO	6
4.1. Estructura Principal.....	6
4.2. Agregar Candidato	8
4.3. Agregar Votante	10
4.4. Agregar Voto (ultimo nivel de anidación).....	12
4.5. Eliminar Mesa (tercer nivel de anidación).....	13
5. PLANIFICACIÓN	15
5.1. Planificación Grupal.....	15
5.2. Planificación Franco Allendes	15
5.3. Planificación Francisco Ceballos	15
5.4. Planificación Alonso Diaz	15
6. CONCLUSIÓN.....	16
6.1. Dominios elegidos	16
6.2. Solución propuesta.....	16
6.3. Experiencias adquiridas.....	17
7. ANEXOS.....	18

LISTA DE FIGURAS

Figura 4.1 Estructura Principal.....	7
Figura 4.2: Agregar candidato.....	9
Figura 4.3: Agregar Votante	10
Figura 4.4: Agregar Voto.....	12
Figura 4.5: Eliminar Mesa	13
Figura 5.1: Tabla Gantt Grupal.....	15
Figura 5.2: Tabla Gantt Franco Allendes	15
Figura 5.3: Tabla Gantt Francisco Ceballos	15
Figura 5.4: Tabla Gantt Alonso Diaz	15

1. ÍNTRODUCCION

El proceso electoral presidencial en Chile es uno de los mecanismos más importantes de nuestra democracia, caracterizado por su necesidad de transparencia, exactitud y rapidez en el conteo de votos. La gestión de este proceso implica manejar grandes volúmenes de información interconectada: desde el padrón de votantes y la inscripción de candidatos, hasta el funcionamiento de las mesas y el escrutinio final.

El presente informe detalla el diseño y la implementación de un sistema de software desarrollado en lenguaje ANSI C, capaz de simular y gestionar las etapas críticas de una elección presidencial. El objetivo principal de este proyecto es aplicar estructuras de datos eficaces para resolver la complejidad del flujo electoral.

Para lograr esto, se han integrado diversas estructuras de datos dinámicas y estáticas. Se utilizó un Árbol Binario de Búsqueda (ABB) para la gestión eficaz de las mesas de votación, una Lista Dblemente Enlazada para la administración del padrón electoral (permitiendo un manejo flexible de los votantes) y Listas Circulares para el registro de votos, garantizando la integridad del proceso. Además, el sistema contempla la lógica necesaria para gestionar una posible segunda vuelta, asegurando el cumplimiento de la normativa vigente.

2. DOMINIO DEL PROBLEMA

El sistema propuesto tiene como objetivo modelar el flujo de trabajo completo del Proceso Electoral Presidencial de Chile, a partir del registro de candidaturas hasta la proclamación del candidato electo ganador. Este proceso corresponde esencialmente a un mecanismo constitucional el cual busca asegurar transparencia y seguridad para los votantes. Para ello, el sistema busca garantizar el funcionamiento correcto y preciso de las etapas que conforman el Proceso Electoral, esto implica asegurar que los cómputos y datos registrados son inalterables con el fin de cumplir con las exigencias constitucionales.

Es por eso, que como grupo se elaboró un diagrama simple que busca conceptualizar como debiese de estructurarse el programa para llevar a cabo el manejo del registro de candidaturas, la validación de este, votación y mesas, conteo de votos y la declaración de resultados. El funcionamiento del programa permite al usuario buscar, agregar, modificar y eliminar, sin embargo, eliminar votos esta estrictamente prohibido ya que no cumple con el marco legal.

3. ANÁLISIS

En conjunto y tras analizar las estructuras a utilizar, llegamos a la conclusión de que se debería usar lo siguiente:

3.1. Estructura sistema electoral

Esta estructura representa a la plataforma de gestión del proceso electoral. En otras palabras, básicamente es quien contiene y enlaza todas las partes del sistema. Por ello, contiene un puntero a la lista simple de Elecciones la cual a su vez contiene a las mesas de votación y los votos, y otro puntero a la lista doblemente enlazada del Padrón electoral donde se almacenan los datos del votante.

3.2. Estructura elección

Esta estructura de datos representa una instancia de elección, es decir, una vuelta electoral (ya sea primera o segunda), es una lista enlazada simple y a su vez esta contiene los siguientes datos:

- Vuelta: dato de tipo int, identifica si es primera o segunda vuelta electoral.
- Fecha: dato de tipo char, corresponde a la fecha en la que se lleva a cabo la vuelta electoral en cuestión.
- Candidatos: un puntero a un arreglo la cuál referencia a los candidatos activos. Es decir, si fuese segunda vuelta, habría solo dos candidatos.
- Número de Candidatos: dato de tipo int, referencia el total de candidatos en la vuelta.
- Mesas: un puntero a la raíz del árbol binario de búsqueda que contiene a las mesas de sufragio.

3.3. Estructura votante

Esta estructura de datos maneja la información de un ciudadano habilitado para sufragar, es una lista doblemente enlazada. Los datos que contiene esta estructura son:

- RUT: dato de tipo char, que representa el RUT del votante.
- Nombre: dato de tipo char, nombre del votante.
- Edad: dato de tipo int, es importante verificar que el votante cumpla con el requisito legal de mayoría de edad.
- Nacionalidad: dato de tipo char, representa la nacionalidad del votante.
- País, Comuna y Región: datos de tipo char, representa el domicilio del votante, ya sea dentro del territorio o fuera de él.

- Ha votado: dato de tipo int, indica si el votante ya voto o no, esto es clave para asegurar el voto único.
- Mesa asignada: puntero a estructura de datos, es un árbol binario de búsqueda y es la conexión de cada votante a su mesa designada.

3.4. Estructura mesa

Esta estructura de datos representa una mesa receptora de sufragio más conocida popularmente como mesa de votación dentro del sistema electoral. Es un árbol binario de búsqueda, con lo cual se accede con mayor rapidez la mesa mediante el ID. Los datos que contiene la estructura son:

- ID mesa: dato de tipo int, este simboliza a una mesa específica.
- Comuna y Región: datos de tipo char, referencian la ubicación geográfica de la mesa de votación.
- Votos emitidos: dato de tipo int, lleva la cuenta de la cantidad de votos registrados en una mesa específica.
- Conteo de votos: dato de tipo int, almacena los votos válidos obtenidos por candidato en esa mesa.
- Votos Nulos y Blancos: datos de tipo int, llevan registro para los votos no válidos.
- Lista votos: un puntero a una lista circular que compone el registro individual de la totalidad de votos en la mesa.

3.5. Estructura candidato

Esta estructura de datos representa los datos de un candidato a la presidencia, es un arreglo y maneja lo siguiente:

- ID candidato: dato de tipo int, representa a un candidato específico.
- RUT, Nombre y Partido: datos de tipo char, referencian la identificación personal y la pertenencia política del candidato en cuestión.
- Edad: dato de tipo int, es importante verificar que el candidato cumpla con el requisito constitucional (mayor o igual a 35 años).
- Firmas de apoyo: dato de tipo int, permite verificar si el candidato es independiente y cumple con el requisito de las firmas.
- Posición libre: dato de tipo int, simboliza si el espacio del arreglo está disponible u ocupado por un candidato.

3.6. Estructura voto

Esta estructura de datos representa el registro individual de un voto, tal vez sea la estructura más pequeña comparativamente, sin embargo, la eliminación de votos esta estrictamente

prohibida por la ley chilena y de ahí su importancia. Es una lista circular y los datos que almacena son:

- RUT votante: dato de tipo char, permite verificar que el voto emitido fue realizado por un votante habilitado para el sufragio.
- ID candidato votado: dato de tipo int, en pocas palabras corresponde a la opción escogida por el votante. Este dato posteriormente es redirigido al conteo local de votos.

La implementación de las estructuras mostradas permite al Servicio Electoral de Chile (SERVEL), desempeñar un correcto funcionamiento y preservar correctamente la integridad del sistema (votos, candidatos, mesas, votantes). La combinación de las estructuras propuestas soporta los requerimientos necesarios impuestos para el correcto desarrollo y gestión del proceso electoral en nuestro país.

4. DISEÑO

Para poder diseñar el sistema, nos organizamos en pensar como representar un flujo real de una elección presidencial en Chile, tomando en cuenta desde el registro de votantes hasta el escrutinio final. Las estructuras de datos elegidas para este diseño fueron adecuadas para cada parte del proceso, pensando en todo momento que este sistema lo podrían llegar a utilizar millones de chilenos, por lo tanto, la tarea de buscar miles de mesas o datos como el RUT, debe ser eficaz y fáciles de actualizar. El diseño se enfoca en mantener una relación lógica y fluida entre las distintas estructuras de dato, como el padrón electoral, mesas de votación, los candidatos y las urnas de votos.

4.1. Estructura Principal

En la siguiente figura se puede observar como está distribuido nuestro sistema:

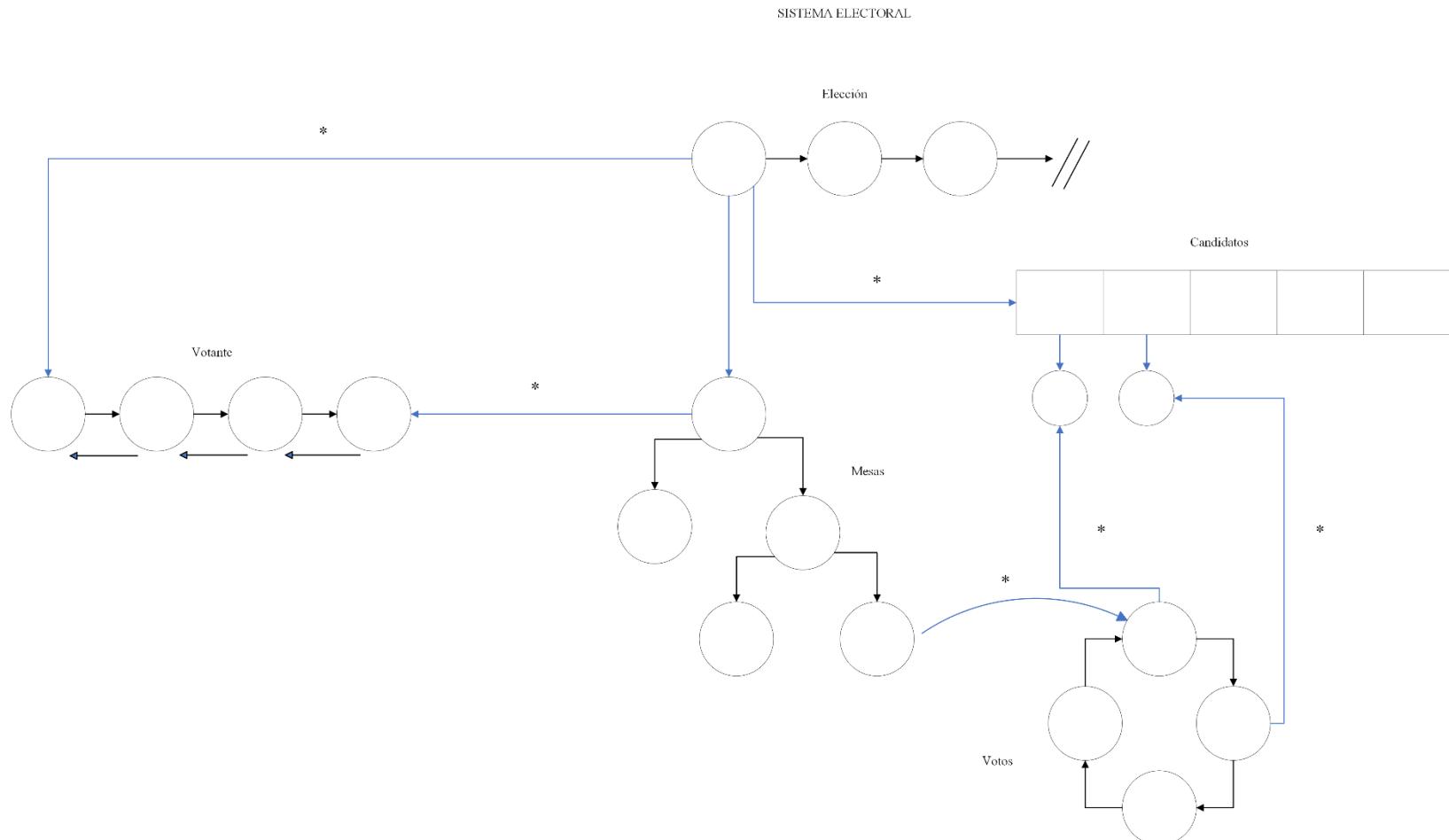


Figura 4.1 Estructura Principal

El sistema central gestiona las distintas vueltas electorales mediante una lista simple simplemente enlazada, El padrón de votantes se guarda en una lista doblemente enlazada, facilitando la administración y permitiendo un recorrido bidireccional de los ciudadanos. Las mesas de votación se guardan en un árbol binario de búsqueda (ABB) dentro de cada vuelta de la elección, gracias a esto encontrar una mesa por su ID es eficaz a la hora de registrar el voto. Por último, los votos se almacenan en una lista circular dentro de cada mesa, permitiendo un registro continuo de los votos, mientras que los candidatos se gestionan en un arreglo estático no compacto, utilizando un puntero libre (Plibre) para asignar eficazmente la memoria.

4.2. Agregar Candidato

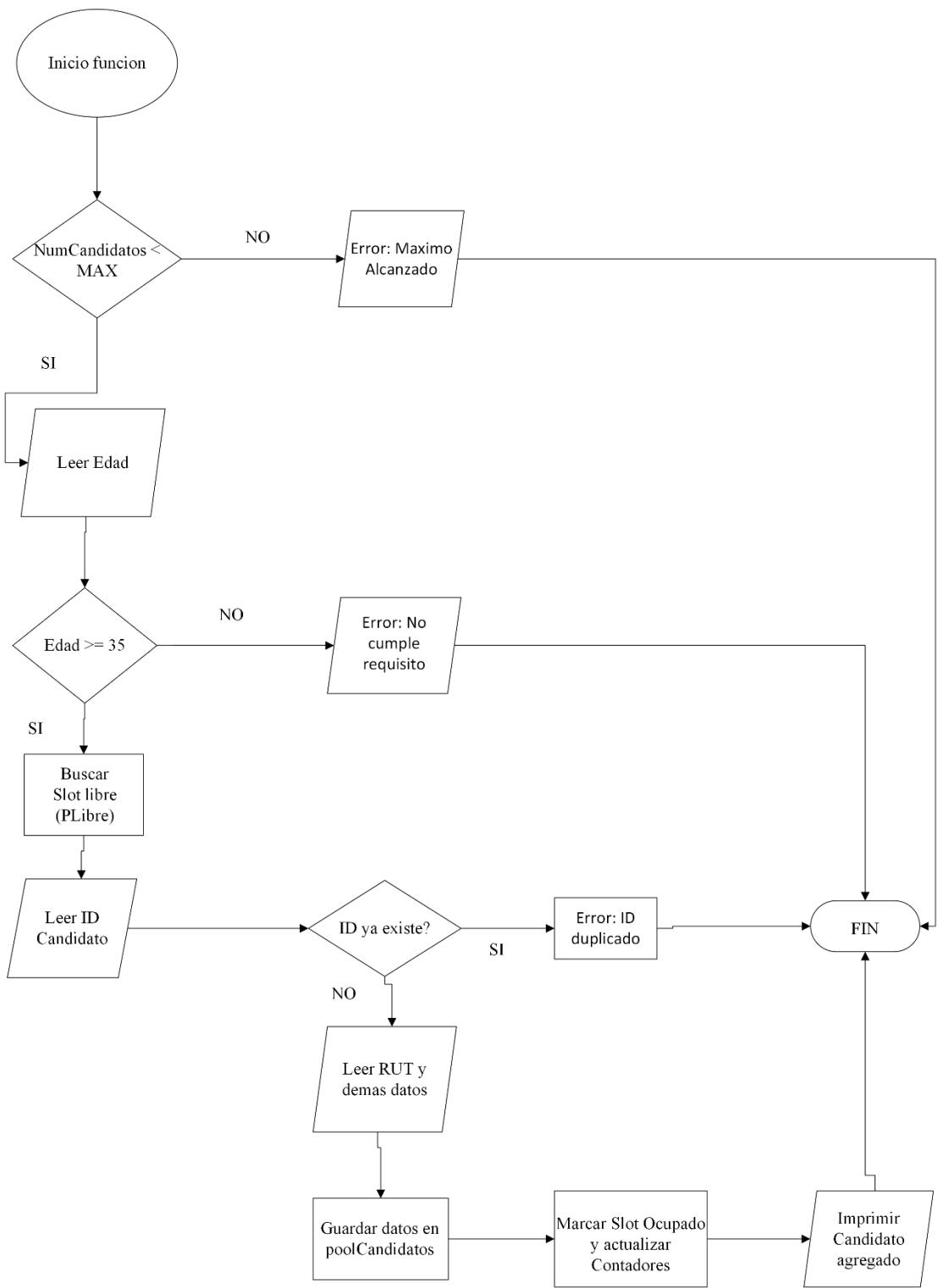


Figura 4.2: Agregar candidato

Este diagrama muestra el algoritmo implementado en la función **agregarCandidato**, que está diseñado para la inscripción de postulantes en un arreglo estático de tamaño fijo (**MAXCANDIDATOS**). El flujo comienza validando que no se haya sobrepasado la cantidad máxima permitida por el sistema y que el candidato cumpla con el requisito legal de edad mínima (35 años). Luego, se ejecuta la lógica de asignación de memoria mediante la estrategia de "Primer Ajuste" (**First Fit**) utilizando la variable **pLibre** o buscando espacios reutilizables. Antes de confirmar el registro, se valida si el ID ya existe, para evitar duplicados. Finalmente se almacenan los datos del candidato (incluyendo RUT y firmas de apoyo independiente) y se actualiza el estado del slot a "ocupado".

4.3. Agregar Votante

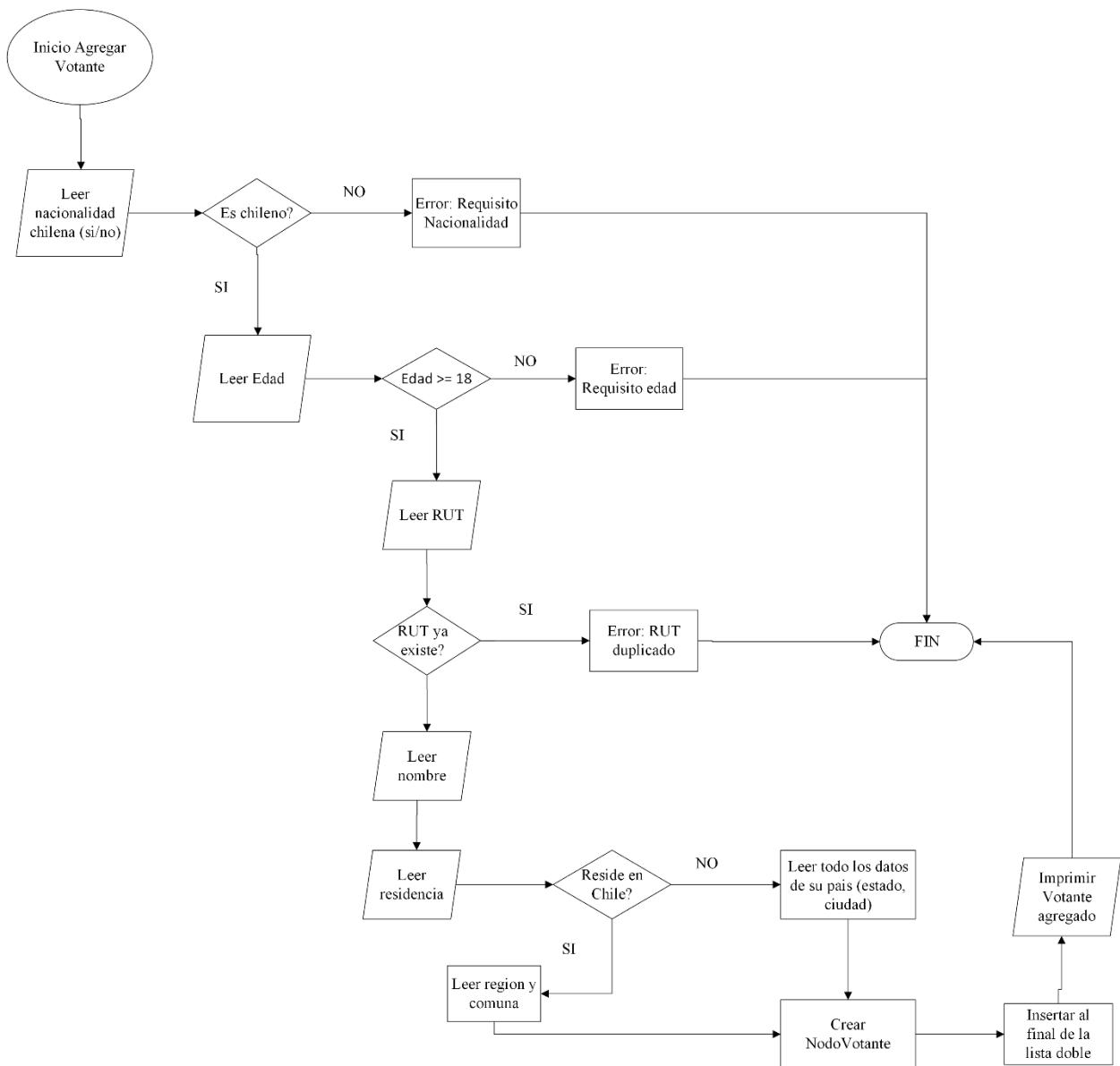


Figura 4.3: Agregar Votante

En este diagrama se puede ver el procedimiento lógico de la función **agregarVotante**, encargada de inscribir nuevos ciudadanos al padrón electoral. El proceso inicia con un bloque de validaciones según los requisitos del sistema actual, validando que su nacionalidad sea chilena y su edad mínima (18 años). Una vez superados estos filtros, el sistema solicita el RUT para verificar que no exista una inscripción previa. Una vez realizado este paso, empieza la bifurcación lógica para determinar la residencia, dependiendo de si el votante reside en Chile o en el extranjero, el sistema solicita los datos geográficos correspondientes (región/comuna o estado/ciudad). Finalmente, se agrega un nuevo nodo dinámico (**nodoVotante**) y se inserta al final de la lista doblemente enlazada, actualizando los punteros **tail** del sistema para mantener la continuidad de la estructura.

4.4. Agregar Voto (ultimo nivel de anidación)

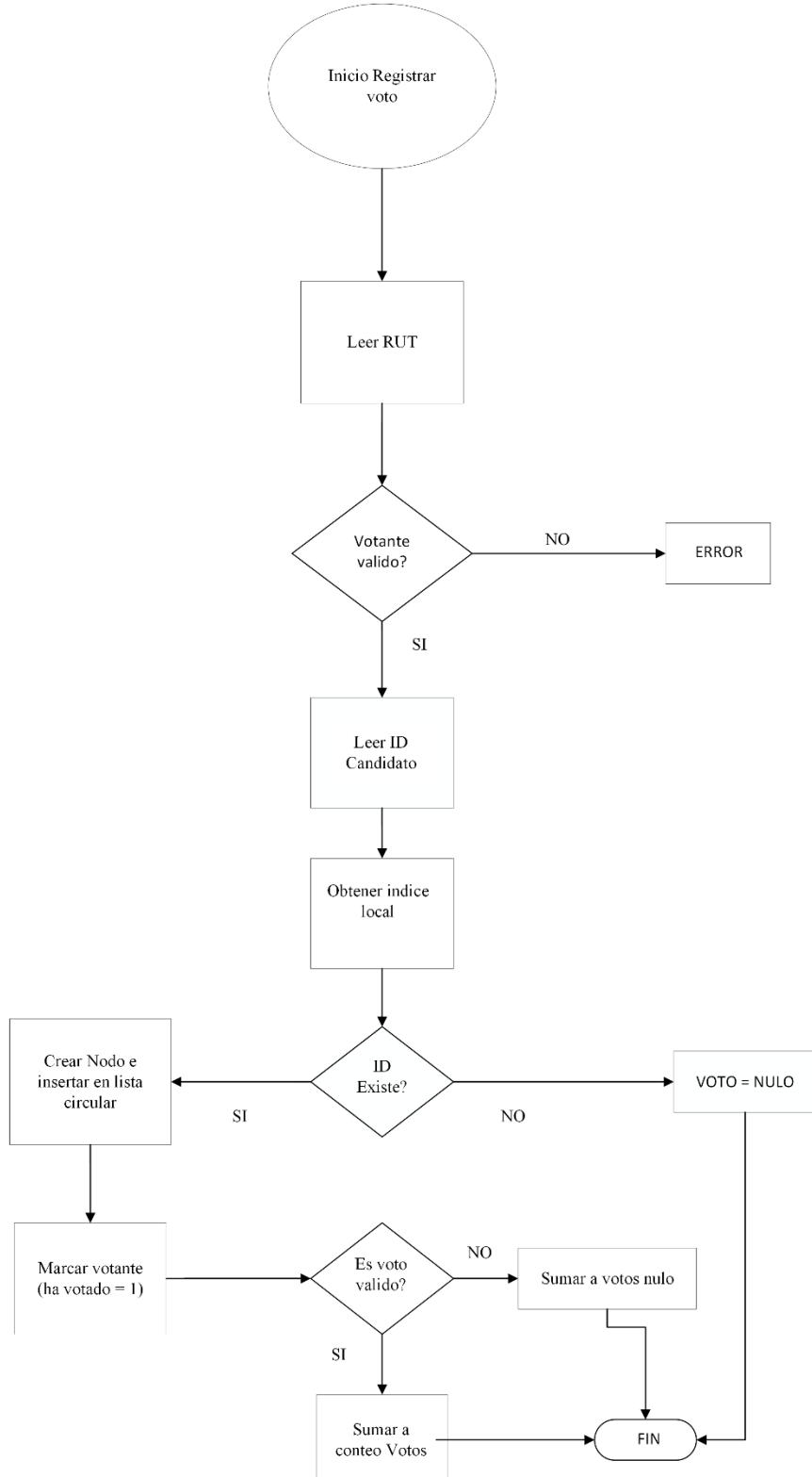


Figura 4.4: Agregar Voto

Este diagrama se puede ver la lógica de la función **registrarVoto**, la cual corresponde al último nivel de anidación en el sistema electoral, que es dentro de los nodos de las mesas. El flujo inicia validando la identidad del votante mediante el RUT y si este habilitado para votar, luego se lee el voto y se valida la existencia de este por la función **obtenerIndiceLocal**. Finalmente, se actualizan las estructuras de datos. Se inserta el nuevo nodo en la lista circular de auditoría y se incrementan los contadores de los votos (validos, nulos o blancos) en el arreglo de escrutinio de la mesa.

4.5. Eliminar Mesa (tercer nivel de anidación)

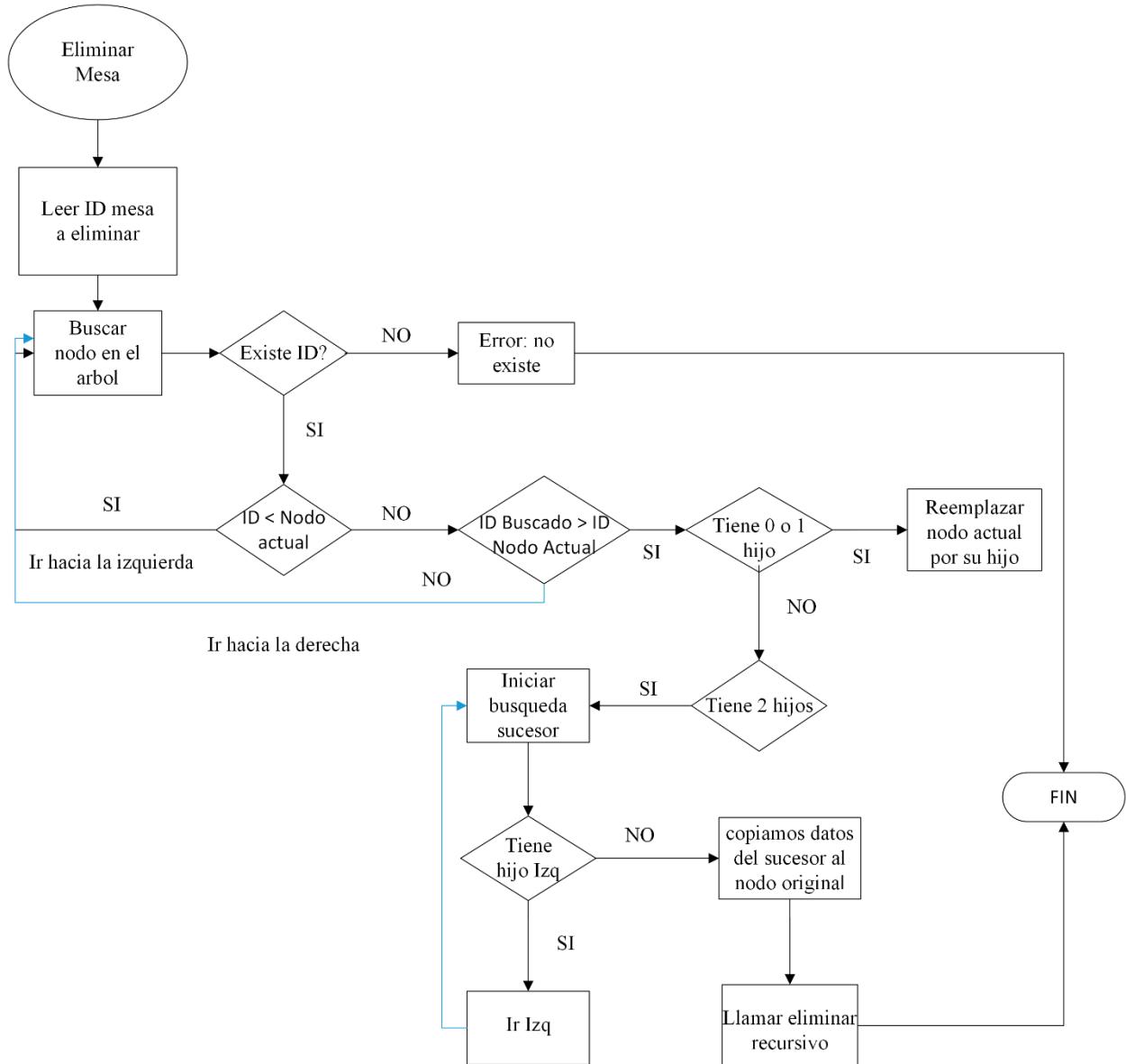


Figura 4.5: Eliminar Mesa

Este diagrama muestra la lógica detrás del algoritmo de eliminación implementado en la función **eliminarMesa**, se encuentra en el tercer nivel de anidación del sistema. El flujo describe el proceso de búsqueda recursiva en el árbol binario de búsqueda (ABB) para localizar el nodo por su ID. Una vez se encuentra, la función evalúa la topología del nodo para determinar la estrategia de borrado; Para casos simples (nodos hoja o con un solo hijo), este se reemplaza directamente con un puntero; para el caso complejo (nodo con dos hijos), se ejecuta el procedimiento de sustitución por el sucesor **in-order**, de esta manera se copian sus datos y eliminando recursivamente el nodo duplicado para preservar la integridad estructural y el orden del árbol.

5. PLANIFICACIÓN

5.1. Planificación Grupal

5.2. Planificación Franco Allendes

5.3. Planificación Francisco Ceballos

5.4. Planificación Alonso Diaz



Figura 5.1: Tabla Gantt Grupal



Figura 5.2: Tabla Gantt Franco Allendes



Figura 5.3: Tabla Gantt Francisco Ceballos



Figura 5.4: Tabla Gantt Alonso Diaz

6. CONCLUSIÓN

Como Grupo para este gran proyecto, nos vimos en la necesidad de tomar numerosas decisiones apoyándonos en lo aprendido durante el semestre y en el consenso del equipo. En esta conclusión expresamos nuestras reflexiones finales sobre la forma en que abordamos el problema, la solución que planteamos y los conocimientos que obtuvimos a lo largo del desarrollo.

6.1. Dominios elegidos

En este apartado ofreceremos una mirada general a nuestras reflexiones finales sobre las diversas herramientas empleadas durante el desarrollo del proyecto.

- GitHub: Lo utilizamos como plataforma para resguardar nuestro código, y resultó especialmente útil por su facilidad para llevar un control de versiones. Gracias a ello contamos con un historial de cambios que nos permite revisar en qué momento se realizó cada avance cuando lo necesitamos.
- JetBrains CLion: fue una herramienta muy valiosa para el trabajo en equipo durante este proyecto. Pudimos aprovechar varias de sus funciones que agilizaron el desarrollo del código, como la búsqueda de estructuras de datos ya definidas y la posibilidad de verificar rápidamente si estaban contenidas dentro de otras. También facilitó identificar los cambios realizados en cada línea gracias al color indicador junto al número, además de permitirnos actualizar nuestro código en GitHub mediante la terminal integrada.
- Microsoft Visio: Este software está orientado específicamente a la creación de diagramas, por lo que, aunque fuera una herramienta obligatoria para el trabajo, también resultó ser la más adecuada para elaborar el nuestro. Sin embargo, tuvimos algunas dificultades al usarlo por primera vez y su interfaz nos pareció un tanto poco intuitivo.
- Canva: Gracias a sus plantillas prediseñadas y a lo fácil que resulta utilizarlo, pudimos crear de manera rápida las diapositivas necesarias para la presentación de nuestro trabajo.

6.2. Solución propuesta

La solución que diseñamos resultó ser efectiva para modelar el Proceso Electoral Presidencial, ya que logramos organizar correctamente cada etapa y mantener la integridad del sistema. Los aspectos más relevantes son:

- Estructura central jerárquica, que coordina elecciones, mesas, votantes y candidatos de forma clara y ordenada.
- Árbol Binario de Búsqueda para las mesas, permitiendo búsquedas rápidas por ID y una administración eficaz del flujo de votos.
- Lista Dblemente Enlazada para el padrón electoral, facilitando el manejo de votantes y el recorrido en ambos sentidos.
- Listas Circulares para los votos, asegurando continuidad, imposibilidad de eliminación y trazabilidad del registro.

- Arreglo estático con pLibre para candidatos, optimizando memoria y evitando duplicación de IDs.
- Validaciones legales integradas, como edad mínima, firmas, voto único y tratamiento correcto de nulos y blancos.
- Soporte para segunda vuelta, manteniendo la lógica del proceso electoral real.

Este conjunto de decisiones permitió construir un sistema coherente, funcional y alineado con las exigencias del proceso electoral chileno.

6.3. Experiencias adquiridas

Durante el proyecto reforzamos el uso práctico de estructuras de datos y su aplicación en sistemas reales. Aprendimos a diseñar soluciones modulares, mejorar la organización del código y trabajar con múltiples niveles de anidación. A nivel grupal, fortalecimos la colaboración mediante GitHub y el uso coordinado de herramientas como CLion y Visio.

Este trabajo nos permitió comprender mejor la importancia de la integridad de los datos y la eficiencia en procesos complejos como una elección presidencial.

7. ANEXOS

A continuación, se adjunta el enlace al repositorio de GitHub correspondiente al desarrollo del código:

- GitHub: <https://github.com/FrancoAllendes/Proyecto-ProcesoElectoral>