

PONTIFICIA UNIVERSIDAD CATÓLICA DE
VALPARAÍSO

FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**DESARROLLO E IMPLEMENTACION DE UN
SISTEMA DE GESTION PARA EL PROCESO
LEGISLATIVO CHILENO EN C**

GABRIEL ALEJANDO FUENTES SAGARDIA

ALEXIS EDUARDO ESCOBAR FIGUEROA

ESTRUCTURA DE DATOS(INF2223-1)
INGENERIA EN INFORMÁTICA

DICIEMBRE 2024

Índice

1

2	Resumen	3
3	ABSTRACT	3
4	Glosario	4
5	Lista de Figuras	5
6	Introducción.....	6
7	Dominio Del Problema.....	7
8	Análisis	8
8.1	Identificación de Estructuras de Datos:	8
8.2	Arreglo dinámico para congresistas (Diputados - Senadores) y comisiones	8
8.3	Lista Simplemente Enlazada para congresistas en comisiones y votaciones.....	8
8.4	Árbol Binario De Búsqueda (ABB) para proyectos de ley	9
8.5	Lista Simplemente Enlazada para las votaciones de cada proyecto de ley	10
8.6	Grafo en forma de matriz	10
8.7	Justificación de las estructuras de datos utilizadas.....	11
9	Diseño	11
9.1	Estructuras de Datos:	11
9.2	Funciones Principales del proyecto	12
9.3	Gestiones de proyectos de ley.....	13
9.4	Gestión congresistas	13
9.5	Gestión de comisiones	14
9.6	Gestión de votaciones	15
9.7	Gestión de grafo	16
9.8	Diagrama de Estructura General:	17
9.9	Diagrama para Agregar un Elemento (ultimo nivel de anidación):.....	17
9.10	Diagrama para Eliminar un Elemento (3er nivel de anidación):	18
10	Planificación.....	18
10.1	Planificación Grupal.....	18
10.2	Planificación Individual	19
11	Conclusiones	19
12	Bibliografía	20
13	Anexo.....	21

13.1	Código Fuente.....	21
13.2	Currículums de los Integrantes	21

2 Resumen

El presente informe e implementación de un sistema de gestión para el proceso legislativo chileno, programado en lenguaje ANSI C. El proyecto busca modelar y gestionar estructuras legislativas, permitiendo la manipulación de datos relacionados con congresistas y proyectos de ley a través de estructuras de datos como Árboles Binarios de Búsqueda (ABB), listas enlazadas y arreglos.

El sistema abarca funcionalidades claves, incluyendo la creación, agregación y gestión de congresistas, la inserción y búsqueda de proyectos de ley, y la interacción con una interfaz de menú que permite a los usuarios realizar múltiples operaciones. El objetivo principal es proporcionar una herramienta eficiente que facilite la administración y consulta de información legislativa. Se detallan las decisiones de diseño y las estructuras de datos empleadas. Además, se abordan los desafíos encontrados, como la gestión de la memoria y la validación de datos, y las soluciones implementadas para asegurar la eficacia del programa.

Palabras claves: *Proceso Legislativo Chileno, ANSI C, Árbol Binario de Búsqueda (ABB), Listas Enlazadas, Arreglos, Proyectos de Ley.*

3 ABSTRACT

This work presents the development and implementation of a management system for the Chilean legislative process, designed in ANSI C. The system models and manages legislative structures using data structures such as Binary Search Trees (BST), linked lists, and arrays, optimizing the handling of information related to legislators and bills. Key features include the creation, management, and query of legislators, as well as the insertion and search of bills, all managed through an interactive menu interface.

The primary goal of the system is to provide an efficient tool for the administration and consultation of legislative information. Design decisions, chosen data structures, and

challenges such as memory management and data validation are discussed, ensuring the program's robustness and effectiveness.

Keywords: *Chilean Legislative Process, ANSI C, Binary Search Tree (BST), Linked Lists, Arrays, Bills.*

4 Glosario

Proceso Legislativo Chileno: Conjunto de procedimientos y etapas a través de los cuales se elaboran, debaten y aprueban las leyes en Chile.

ANSI C: Estándar del lenguaje de programación C definido por el American National Standards Institute (ANSI), que garantiza portabilidad y consistencia en diferentes sistemas.

Estructura de Datos: Es un modelo organizativo utilizado para almacenar, gestionar y operar con información de manera eficiente. Ejemplos de estructuras de datos utilizadas en este proyecto incluyen Árboles Binarios de Búsqueda (ABB), listas enlazadas y arreglos.

Árbol Binario de Búsqueda (ABB): Estructura de datos jerárquica en la que cada nodo tiene como máximo dos hijos (izquierdo y derecho), y los valores en el subárbol izquierdo son menores que el nodo padre, mientras que los del subárbol derecho son mayores.

Listas Enlazadas: Estructura de datos en la que los elementos (nodos) están conectados secuencialmente mediante punteros, permitiendo una gestión flexible del tamaño y almacenamiento dinámico de datos.

Nodo: Es la unidad básica de muchas estructuras de datos, como árboles y listas enlazadas. Contiene un valor o dato y, en algunos casos, referencias a otros nodos, lo que permite establecer relaciones entre ellos.

Arreglos: Estructura de datos que almacena elementos del mismo tipo en posiciones consecutivas de memoria, lo que permite el acceso rápido mediante índices.

Congresistas: Miembros del Congreso que participan en el proceso legislativo mediante la elaboración, discusión y aprobación de leyes.

Fases: Corresponden a las etapas específicas por las cuales transita una propuesta desde su creación hasta su aprobación o rechazo, incluyendo la presentación, discusión, votación y promulgación.

5 Lista de Figuras

Figura 9.1 Diagrama Estructura De Datos General.....	17
Figura 9.2 Diagrama de agregación ultimo nivel anidación congresista.....	17
Figura 9.3 Diagrama De Eliminación 3er nivel de anidación congresista en lista	18

6 Introducción

En este informe se describe el desarrollo e implementación de un sistema de gestión para el sistema legislativo chileno, programado en el lenguaje C. Este proyecto tiene como objetivo principal crear una solución eficiente y organizada para manejar información legislativa, que incluye el registro, consulta y administración de congresistas y proyectos de ley, así como la vinculación de datos relevantes, como las votaciones.

En el contexto del sistema legislativo, la gestión de grandes volúmenes de datos y la necesidad de mantener información actualizada y accesible es crucial para la toma de decisiones. Este sistema busca abordar estas necesidades mediante la utilización de estructuras de datos adecuadas, como Árboles Binarios de Búsqueda (ABB), arreglos dinámicos y listas enlazadas, que permitan operaciones eficientes, tales como inserción y eliminación de manera rápida, y la manipulación de datos de forma correcta.

A lo largo del informe, se detallarán las problemáticas específicas que el sistema pretende resolver, las decisiones tomadas en términos de diseño y elección de estructuras de datos, y se proporcionarán diagramas explicativos para una mejor comprensión del flujo y organización del sistema. También se documentará la planificación realizada, tanto grupal como individual, y se incluirá una reflexión sobre las lecciones aprendidas y la experiencia obtenida durante el desarrollo del proyecto.

7 Dominio Del Problema

El sistema legislativo chileno es una entidad compleja que involucra a múltiples actores y procesos para la creación y aprobación de leyes. Gestionar y organizar la información de manera eficiente es un desafío importante, ya que se requiere un almacenamiento estructurado y un acceso rápido a datos esenciales, como los detalles de congresistas, comisiones, proyectos de ley y sus respectivas fases y votaciones.

El problema principal que aborda este proyecto es la necesidad de una solución programática para gestionar de manera ordenada y eficiente la información legislativa. Esto incluye el almacenamiento, la búsqueda, la agregación, la inserción y la administración de datos asociados a los actores involucrados, que en el contexto real son numerosos y están en constante evolución. Es importante que las relaciones entre estos elementos estén correctamente implementadas para permitir la trazabilidad y el análisis de los datos.

Los elementos y datos clave involucrados en este sistema legislativo son:

- **Congresistas:** Información sobre sus nombres, RUTs, partidos políticos y ocupación (diputado o senador).
- **Proyectos de Ley:** Datos específicos como ID únicos, nombre, tipo, niveles de urgencia y fase de los proyectos.
- **Votaciones:** Registro de cómo se ha votado en cada proyecto, lo que es crucial para entender el proceso legislativo, incluye datos como ID únicos, ID del proyecto asociado y resultado de la votación.
- **Cámaras:** Esta compuesta por congresistas y se dividen en cámara de Diputados y cámara de Senadores, su función principal son la aprobación de leyes.
- **Comisiones:** Grupos de congresistas expertos en cierta área, encargados de analizar y debatir proyectos para realizar su posterior votación y determinar si el proyecto avanza.

8 Análisis

8.1 Identificación de Estructuras de Datos:

En el desarrollo del sistema de gestión legislativa, se han definido varias estructuras de datos clave que permiten organizar y manipular eficientemente la información legislativa para así de esta manera cumplir con los elementos y actores claves mencionados anteriormente. A continuación, se describen las estructuras y sus justificaciones

8.2 Arreglo dinámico para congresistas (Diputados - Senadores) y comisiones

En el contexto del sistema legislativo chileno se conoce el número aproximado de congresistas (100 diputados y 50 senadores), la decisión de utilizar un arreglo dinámico debido a la flexibilidad que ofrece ante cambios futuros. Si bien el número de congresistas es conocido actualmente, el uso de un arreglo dinámico permite adaptarse fácilmente a posibles modificaciones que alteren este número, así como situaciones como renuncias o cambios en la ley que puedan modificar el número de congresistas. Además, los arreglos dinámicos permiten agregar o eliminar elementos sin necesidad de redimensionar manualmente la estructura de datos.

Por otro lado, existen aproximadamente 55 comisiones en total, distribuidas entre 28 comisiones para diputados y 27 comisiones para senadores. Se ha decidido utilizar un arreglo dinámico para almacenar y gestionar estas comisiones debido a la misma necesidad de flexibilidad. A pesar de que el número de comisiones es conocido actualmente, un arreglo dinámico ofrece la capacidad de adaptarse a cambios futuros, como la creación de nuevas comisiones o la eliminación de otras, sin necesidad de realizar ajustes manuales en la estructura de datos. Además, un arreglo dinámico permite agregar o eliminar comisiones de manera eficiente, facilitando la actualización del sistema conforme a la evolución de la organización legislativa.

8.3 Lista Simplemente Enlazada para congresistas en comisiones y votaciones

En se ha optado por utilizar una lista simplemente enlazada para almacenar los congresistas en cada comisión y votaciones. Esta decisión responde a las características y

requerimientos específicos del problema. Las comisiones en el Congreso están formadas por un grupo de congresistas que pueden ingresar o salir de una comisión a lo largo del tiempo, por ejemplo, debido a cambios de funciones o redistribución de congresistas. Dado que no se conoce con certeza cuales congresistas estarán en cada comisión, utilizar una lista simplemente enlazada permite manejar esta dinamicidad de manera eficiente. La lista enlazada permite agregar o quitar congresistas sin la necesidad de mover otros elementos, lo que optimiza el proceso de modificación.

Similar a las comisiones, las votaciones también requieren un registro de congresistas, pero en este caso, se tiene una lista de congresistas que votan a favor, en contra o abstenciones en cada proyecto de ley. Al igual que en las comisiones, el número de congresistas que participan en las votaciones puede variar dependiendo del contexto, y un congresista puede ser añadido o eliminado de una lista de votación en cualquier momento (por ejemplo, si cambia su voto en el proceso o si un congresista se ausenta). La lista simplemente enlazada es adecuada porque permite manejar estos cambios dinámicos sin la necesidad de reestructurar otros datos o elementos.

8.4 Árbol Binario De Búsqueda (ABB) para proyectos de ley

El uso de un Árbol Binario de Búsqueda (ABB) para gestionar los proyectos de ley está justificado por varias razones. El ABB es una estructura de datos eficiente para la búsqueda de elementos, lo que es fundamental en un sistema legislativo donde los proyectos de ley deben ser gestionados, ordenados y consultados rápidamente. Cada proyecto de ley tiene un ID único, que se puede utilizar como clave para la búsqueda en el árbol. Esta característica permite que la búsqueda de proyectos de ley con un identificador específico se realice en tiempo logarítmico determinado, lo que es mucho más eficiente que otros métodos de búsqueda, como la búsqueda lineal en un arreglo, especialmente considerando el gran volumen de datos que el sistema puede manejar.

Además, los proyectos de ley se constantemente en el sistema. El ABB permite insertar nuevos proyectos, manteniendo el orden de los elementos dentro de la estructura sin la necesidad de reorganizar todos los proyectos. Esto asegura que el sistema pueda gestionar dinámicamente el flujo de proyectos de ley, optimizando las operaciones de inserción y búsqueda.

8.5 Lista Simplemente Enlazada para las votaciones de cada proyecto de ley

La justificación para usar una Lista Simplemente Enlazada para las votaciones de cada proyecto de ley es que esta estructura de datos es ideal para manejar colecciones de elementos que se modifican con frecuencia, como en el caso de las votaciones. En el contexto de un proyecto de ley, las votaciones son eventos que pueden variar dependiendo de los actores involucrados (congresistas), y es necesario almacenar los resultados de cada votación de manera eficiente a medida que se van registrando.

La Lista Simplemente Enlazada permite agregar nuevos nodos (votaciones) en cualquier punto, sin necesidad de reorganizar o mover elementos ya existentes. Esto es útil porque las votaciones suelen ser dinámicas y se agregan una tras otra a medida que el proyecto de ley avanza, ya sea en las comisiones o en las cámaras. Además, facilita la gestión de los congresistas involucrados en cada votación y permite recorrer fácilmente las votaciones de un proyecto, ya que solo se necesita recorrer la lista. Su simplicidad y capacidad de adaptación la hacen adecuada para registrar votaciones sin complicaciones adicionales.

8.6 Grafo en forma de matriz

El uso de un grafo para representar el congreso se fundamenta en la necesidad de modelar las relaciones entre congresistas, como su colaboración en proyectos de ley, participación conjunta en comisiones o sus partidos políticos. El grafo ofrece diversas ventajas en este contexto.

La matriz de adyacencia permite una representación directa de las conexiones entre congresistas. Es especialmente útil para realizar consultas rápidas sobre la existencia de relaciones específicas entre dos congresistas, con una complejidad de acceso constante. Por otro lado, la lista de congresistas almacena los datos básicos de cada congresista y establece un mapeo claro entre los índices de la matriz de adyacencia y los congresistas reales. Esto brinda flexibilidad para obtener información detallada de cada congresista mientras se preservan las conexiones definidas en la matriz. El grafo también facilita análisis complejos, como la identificación de grupos de congresistas con alta colaboración, lo cual es particularmente relevante para estudiar patrones en proyectos de ley o votaciones. Además, esta estructura puede emplearse para optimizar la asignación de congresistas a comisiones o visualizar alianzas políticas.

8.7 Justificación de las estructuras de datos utilizadas

Las estructuras de datos seleccionadas para este sistema legislativo han sido elegidas por su eficiencia y flexibilidad, esenciales para manejar operaciones de almacenamiento, acceso y modificación de datos sobre congresistas, proyectos de ley, votaciones y comisiones. Los arreglos dinámicos para congresistas (diputados y senadores) y comisiones permiten adaptarse a cambios en el número de elementos sin la necesidad de redimensionar manualmente. Las listas simplemente enlazadas se utilizan para gestionar las votaciones, ya que permiten la inserción dinámica de nodos sin reorganizar los elementos existentes, ideal para el registro continuo de votaciones. Finalmente, el Árbol Binario de Búsqueda (ABB) se usa para almacenar los proyectos de ley, facilitando búsquedas rápidas y eficientes mientras mantiene el orden al insertar nuevos proyectos. Estas estructuras son adecuadas para gestionar el crecimiento y los cambios dinámicos de los datos en el sistema legislativo.

9 Diseño

El diseño del sistema de gestión legislativa ha sido orientado a la modularidad y eficiencia, utilizando estructuras de datos adecuadas para organizar y manejar la información de manera clara y efectiva. Cada módulo del sistema está diseñado para gestionar un conjunto específico de elementos dentro del contexto legislativo, facilitando el acceso, manipulación y almacenamiento de datos. Las estructuras de datos seleccionadas permiten un manejo dinámico de congresistas, proyectos de ley, votaciones y comisiones, asegurando una rápida adaptación a cambios futuros en la legislación o en la estructura del Congreso.

Las principales estructuras desarrolladas incluyen:

9.1 Estructuras de Datos:

- **Congresistas:** Se gestionan utilizando arreglos dinámicos para almacenar tanto a diputados como a senadores. Cada congresista tiene atributos clave como nombre, RUT, ocupación (diputado o senador) y partido político, lo que permite realizar búsquedas rápidas y gestionar los congresistas según su rol.
- **Comisiones:** Se utilizan arreglos dinámicos para almacenar las comisiones del

Congreso, permitiendo su expansión o modificación conforme a cambios legislativos. Cada comisión tiene un conjunto de congresistas y se clasifica según su tipo (diputados o senadores), nombre e ID.

- **Proyectos de ley:** Se almacenan en un ABB, lo que permite realizar búsquedas eficientes por ID de proyecto y mantiene el orden al agregar nuevos proyectos. Cada proyecto de ley posee un nombre, ID único, nivel de urgencia, fase en la cual se encuentra y una lista de las votaciones que se han desarrollado sobre el mismo.
- **Votaciones:** Las votaciones de los proyectos de ley se gestionan mediante una Lista Simplemente Enlazada, permitiendo la inserción dinámica de votaciones. Cada votación contiene información sobre los congresistas que votaron a favor, en contra o se abstuvieron. También contienen datos como ID único de votación, ID del proyecto votado y el resultado de la votación.
- **Grafo:** Introducido para modelar las relaciones entre congresistas, como colaboración en proyectos, participación conjunta en comisiones o afinidad partidaria. La matriz de adyacencia representa estas conexiones de forma directa, mientras que la lista de congresistas vincula los índices de la matriz con la información detallada de cada congresista.

9.2 Funciones Principales del proyecto

El diseño del sistema de gestión legislativa ha sido orientado a la modularidad y eficiencia, utilizando estructuras de datos adecuadas para organizar y manejar la información de manera clara y efectiva. Cada módulo del sistema está diseñado para gestionar un conjunto específico de elementos dentro del contexto legislativo, facilitando el acceso, manipulación y almacenamiento de datos. Las estructuras de datos seleccionadas permiten un manejo dinámico de congresistas, proyectos de ley, votaciones y comisiones, asegurando una rápida adaptación a cambios futuros en la legislación o en la estructura del Congreso

9.3 Gestiones de proyectos de ley

La gestión de proyectos se centra en la creación, modificación y administración de los proyectos de ley presentados en el Congreso. Su propósito principal es mantener un registro detallado de cada proyecto, incluyendo su título, descripción, estado y los congresistas involucrados en su elaboración. Este módulo facilita el seguimiento del progreso de los proyectos, desde su propuesta inicial hasta su eventual aprobación o rechazo, asegurando la transparencia del proceso legislativo. Las funciones se detallan a continuación:

crearProyectoLey: Crear una nueva instancia de un proyecto de ley. Reserva memoria para los campos del proyecto, copia los datos proporcionados, y asigna valores iniciales como NULL para votaciones relacionadas.

insertarProyectoLey: Insertar un nuevo proyecto de ley en un árbol binario de búsqueda, ordenado por el idProyecto.

buscarProyectoLeyPorID: Buscar y devolver un proyecto de ley en el árbol binario según su idProyecto.

modificarProyectoLey: Modificar la fase de un proyecto de ley identificado por su ID.

mostrarProyectoLeyPorID: Mostrar los detalles de un proyecto de ley buscado por su ID. Si no existe, notifica al usuario.

9.4 Gestión congresistas

La gestión de congresistas se encarga de registrar y administrar la información de los miembros del Congreso. Su propósito principal es mantener un registro estructurado de los congresistas, facilitando su identificación mediante atributos clave como nombre, RUT y partido político. Además, permite gestionar las relaciones entre congresistas a través de un grafo que representa las interacciones o colaboraciones entre ellos. Este módulo es fundamental para construir y analizar la estructura organizativa del Congreso y sirve de base para las demás gestiones. Las funciones se detallan a continuación:

crearCongresista: Crea una nueva instancia de un congresista, reservando memoria y copiando los datos proporcionados.

buscarCongresistaEnArreglo: Busca un congresista en un arreglo dinámico utilizando su RUT como criterio de búsqueda.

agregarCongresistaEnArreglo: Agrega un congresista al arreglo si hay espacio disponible y no existe otro congresista con el mismo RUT.

eliminarCongresistaEnArreglo: Elimina un congresista del arreglo dinámico basado en su RUT.

mostrarCongresistas: Recorre y muestra todos los congresistas presentes en el arreglo.

enlazarCongresista: Crea un nodo de lista enlazada para un congresista.

buscarCongresistaEnLista: Buscar un congresista en una lista enlazada basada en su RUT.

agregarCongresistaEnComisionLista: Agrega un congresista a una lista enlazada de una comisión si no está ya registrado.

eliminarCongresistaEnComisionLista: Elimina un congresista de una lista enlazada de una comisión utilizando su RUT.

mostrarCongresistaEnComisionLista: Mostrar todos los congresistas de una lista enlazada.

9.5 Gestión de comisiones

La gestión de comisiones se enfoca en la creación, asignación y administración de los comités especializados dentro del Congreso. Su propósito principal es organizar y coordinar el trabajo de los congresistas en áreas específicas, como economía, salud o justicia, para garantizar un análisis detallado y eficiente de los proyectos de ley. Este módulo también registra las reuniones, integrantes y decisiones tomadas por cada comisión, asegurando un flujo

de trabajo claro y bien documentado dentro del Congreso. Las funciones se detallan a continuación:

crearComision: Crea una nueva comisión, inicializando sus atributos y reservando memoria.

buscarComisionEnArreglo: Busca una comisión en un arreglo por su ID.

agregarComisionEnArreglo: Agrega una nueva comisión al arreglo, verificando que no se repita.

eliminarComisionEnArreglo: Elimina una comisión del arreglo utilizando su ID.

9.6 Gestión de votaciones

La gestión de votaciones tiene como objetivo principal registrar y procesar los votos emitidos por los congresistas en relación con proyectos de ley u otras decisiones parlamentarias. Este sistema garantiza que cada voto sea contabilizado correctamente, permitiendo analizar los resultados de manera precisa. Además, facilita la visualización de tendencias de votación y el cumplimiento de los quórums requeridos, promoviendo la transparencia y la participación democrática en el congreso. Las funciones se detallan a continuación:

crearVotacion: Crea una nueva votación, inicializando sus campos con valores predeterminados.

buscarVotacionEnLista: Busca una votación en una lista enlazada por su ID.

enlazarVotacion: Agrega una votación a una lista enlazada si no está ya registrada.

registrarVotoDeCongresistaEnVotacion: Registra el voto de un congresista en una votación (a favor, en contra, o abstención). Verifica si ya votó.

calcularResultadoVotacion: Calcula y asigna el resultado de la votación basado en los votos a favor y en contra.

mostrarVotantes: Muestra los congresistas que votaron a favor, en contra, o se abstuvieron en una votación.

mostrarVotaciones: Muestra los detalles de todas las votaciones almacenadas en una lista enlazada.

9.7 Gestión de grafo

Estas funciones trabajan en conjunto para gestionar un grafo de congresistas, donde los nodos representan a los congresistas y las conexiones representan relaciones entre ellos. Se permite agregar congresistas, establecer conexiones directas, y visualizar las relaciones mediante la matriz de adyacencia. Las funciones se detallan a continuación:

inicializarGrafo: Crea e inicializa un grafo vacío que representará las conexiones entre congresistas.

inicializarNuevasConexiones: Configura las nuevas filas y columnas en la matriz de adyacencia cuando se agrega un nuevo congresista al grafo.

agregarCongresistaAlGrafo: Agrega un congresista al grafo, representado tanto en la lista de nodos como en la matriz de adyacencia.

conectarCongresistas: Crea una conexión (relación) entre dos congresistas en el grafo.

obtenerIndiceCongresista: Encuentra el índice de un congresista en la lista del grafo según su RUT.

establecerConexionEntreCongresistas: Establece una conexión directa entre dos congresistas usando sus Ruts.

mostrarMatrizAdyacencia: Muestra la matriz de adyacencia del grafo.

9.8 Diagrama de Estructura General:

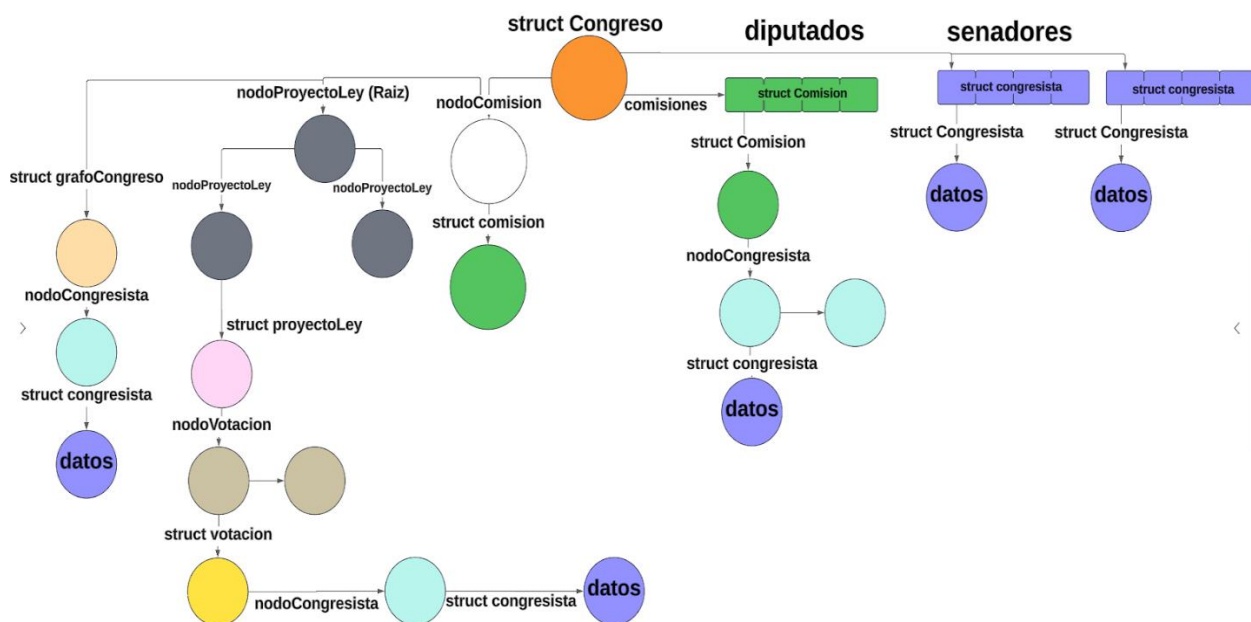


Figura 9.1 Diagrama Estructura De Datos General

9.9 Diagrama para Agregar un Elemento (ultimo nivel de anidación):

Este diagrama ilustra los pasos para agregar un nuevo congresista a la estructura de datos del grafo. El proceso es el siguiente:

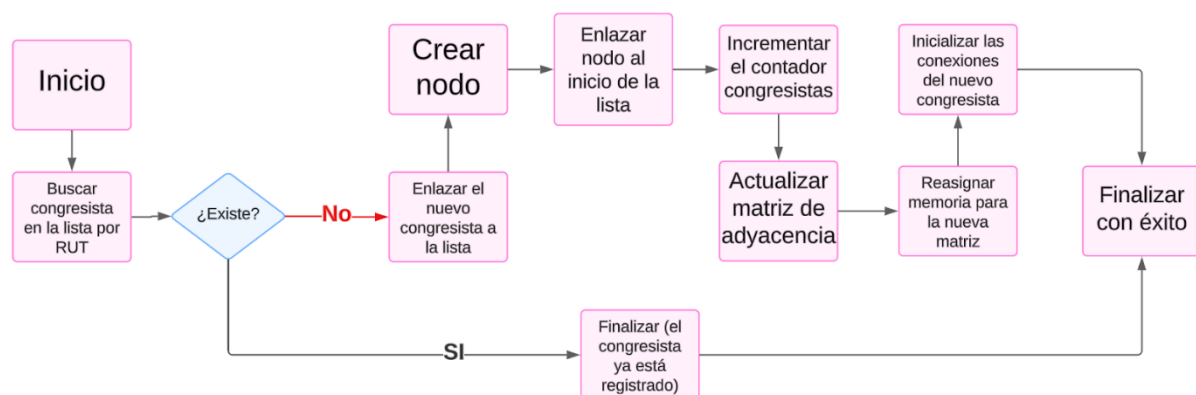


Figura 9.2 Diagrama de agregación ultimo nivel anidación congresista

9.10 Diagrama para Eliminar un Elemento (3er nivel de anidación):

El diagrama detalla los pasos para eliminar un nodo correspondiente a un congresista en la lista vinculada a una comisión, asegurando la consistencia de la estructura. El proceso es el siguiente:

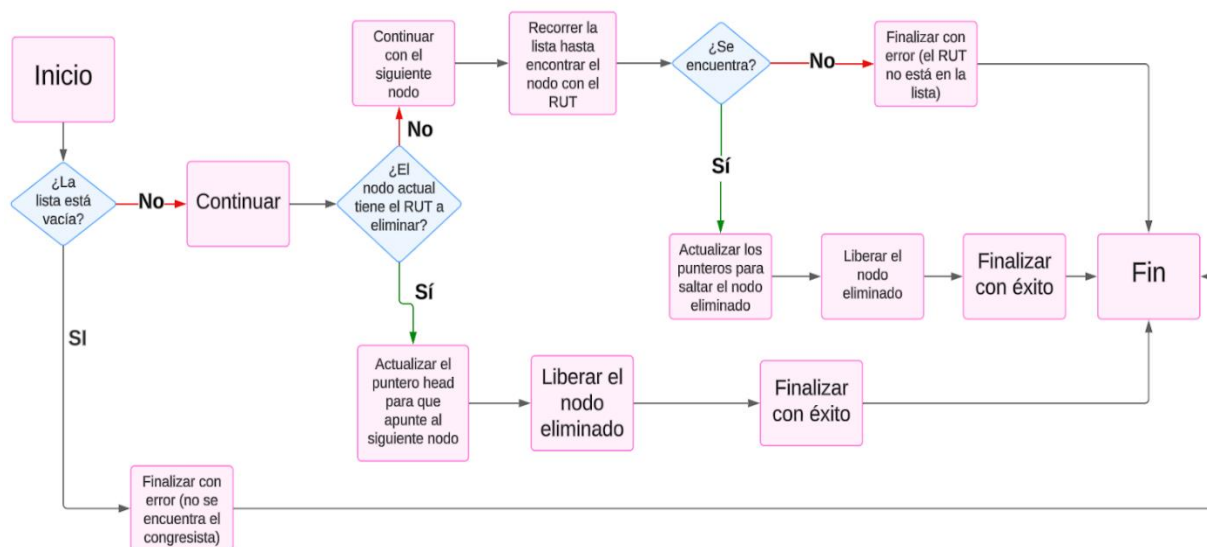


Figura 9.3 Diagrama De Eliminación 3er nivel de anidación congresista de una lista en comisión

10 Planificación

10.1 Planificación Grupal

En esta subsección, se incluirá una tabla Gantt que muestre las tareas del proyecto distribuidas en el tiempo. Esto proporcionará una visión general de las fases del proyecto, las fechas de inicio y finalización.

Nombre de la tarea	Progreso	Fecha de inicio	Fecha de termino
Informe tecnico	Completado	11/29/2024	12/01/2024
Realizar compilaciones y verificaciones	Completado	11/30/2024	12/01/2024
Agregar nuevos requerimientos	Completado	11/29/2024	11/30/2024
Ver menu proyecto y votaciones	Completado	11/29/2024	11/29/2024
Ver funciones proyecto y comisiones	Completado	11/28/2024	11/29/2024
Ver funciones congresistas y congresistas	Completado	11/27/2024	11/28/2024
Modificar struct previa autorizacion	Completado	11/27/2024	11/27/2024
Modificar y rediseñar la estructuras	Completado	11/26/2024	11/26/2024
Corregir errores generales	Completado	11/25/2024	11/26/2024

10.2 Planificación Individual

En esta subsección, se presentará una carta Gantt con las tareas asignadas a cada miembro del equipo.

Nombre de la tarea	Progreso	Asignado a	Fecha de inicio	Fecha de vencimiento
Rediseñar Estructuras	Completado	Gabriel Fuentes	11/25/2024	11/26/2024
Solicitar Autorizacion Modificaciones	Completado	Gabriel Fuentes	11/26/2024	11/26/2024
Realizar Compilaciones y Debug	Completado	Gabriel Fuentes	11/25/2024	11/27/2024
Modificar y agg funciones para congresistas	Completado	Gabriel Fuentes	11/27/2024	11/27/2024
Crear y Ver menu congresistas	Completado	Alexis Escobar	11/27/2024	11/27/2024
Agregar funciones para proyectos de ley	Completado	Gabriel Fuentes	11/28/2024	11/28/2024
Crear y Ver menu proyecto de ley	Completado	Alexis Escobar	11/28/2024	11/28/2024
Agregar funciones para comisiones	Completado	Gabriel Fuentes	11/29/2024	11/29/2024
Crear y Ver menu comisiones	Completado	Alexis Escobar	11/29/2024	11/29/2024
Verificar validaciones	Completado	Alexis Escobar	11/29/2024	11/29/2024
Realizar compilacion y debug nuevas	Completado	Alexis Escobar	11/29/2024	11/29/2024
Implementar ordenamiento por rut y non	Completado	Gabriel Fuentes Sagar	11/30/2024	11/30/2024
Implementar struct grafo	Completado	Gabriel Fuentes Sagar	11/30/2024	11/30/2024
Desarrollar informe	Completado	Gabriel Fuentes Sagar	11/27/2024	12/01/2024
Desarrollar cartas gantt	Completado	Alexis Escobar	12/01/2024	12/01/2024
Desarrollar diagramas de flujo	Completado	Alexis Escobar	11/30/2024	11/30/2024

11 Conclusiones

El desarrollo del sistema de gestión legislativa para el sistema chileno permitió enfrentar y resolver problemas relacionados con la organización y el acceso eficiente a la información legislativa. A través del análisis del dominio del problema, se identificaron las necesidades de un sistema automatizado que no solo gestionara datos de manera eficiente, sino que también brindara flexibilidad para adaptarse a posibles cambios en el entorno legislativo.

La elección de estructuras de datos como el Árbol Binario de Búsqueda (ABB) y las listas enlazadas demostró ser adecuada para operaciones clave como la búsqueda, inserción y actualización de información. Sin embargo, durante el desarrollo se identificaron áreas donde

la implementación podría mejorarse. Por ejemplo, el ABB puede presentar problemas de balanceo si los datos no se insertan de manera uniforme, lo que afectaría su rendimiento. En este sentido, una posible mejora futura sería la implementación de un árbol balanceado, como un AVL, para garantizar un rendimiento constante.

Asimismo, el uso de listas simplemente enlazadas, si bien es eficiente para operaciones dinámicas, puede resultar óptimo para consultas frecuentes que requieren acceso aleatorio. Explorar otras estructuras, como listas doblemente enlazadas, podría mejorar el rendimiento dependiendo de los patrones de uso específicos del sistema.

En cuanto a la experiencia del equipo, el proyecto permitió adquirir habilidades avanzadas en programación en C y en la implementación de estructuras de datos. Sin embargo, uno de los principales retos fue asegurar la robustez del sistema frente a entradas inesperadas o situaciones límite. Esto pone de manifiesto la importancia de realizar pruebas más exhaustivas y considerar herramientas adicionales para garantizar la estabilidad del sistema.

Finalmente, aunque el sistema mejora significativamente la eficiencia en la gestión de datos legislativos, también plantea desafíos en cuanto a su escalabilidad y mantenimiento a largo plazo. Incluir capacidades como el manejo de concurrencia o la posibilidad de integrar nuevas tecnologías, como bases de datos externas o interfaces gráficas, podría fortalecer aún más su funcionalidad.

En conclusión, este proyecto representa un avance importante hacia la automatización del proceso legislativo chileno, pero también deja claro que hay oportunidades para seguir perfeccionándolo, asegurando no solo la eficiencia del sistema, sino también su sostenibilidad y capacidad para adaptarse a las necesidades futuras del entorno legislativo.

12 Bibliografía

https://www.bcn.cl/formacioncivica/detalle_guia?h=10221.3/45683

<https://www.gob.cl/noticias/congreso-nacional-chile-que-hace-integrantes-diputados-senado-pasos-creacion-leyes/>

https://www.camara.cl/camara/historia_camara_de_diputados.aspx

13 Anexo

13.1 Código Fuente

A continuación, en un link de GitHub, se presenta el código fuente completo utilizado en el desarrollo del sistema de gestión legislativa.

<https://github.com/GaboCrvzy/SistemaLegislativo/blob/5ab62b03d68faecf83d5b77836b1bd6fd0c9f5d3/JOSE%20GONZALEZ.c>

13.2 Currículums de los Integrantes

A continuación, se presentan los currículums de los integrantes del equipo:

https://www.linkedin.com/in/gabriel-fuentes-sagardia-828a15336?utm_source=share&utm_campaign=share_via&utm_content=profile&utm_medium=ios_app

<https://www.linkedin.com/in/alexisescobar19>