

Desarrollo e Implementación de un Sistema de Gestión para el Proceso Legislativo Chileno en C

Gabriel Fuentes
Alexis Escobar
INF2223-1



Introducción

- Objetivo : Diseñar e implementar un sistema que permita gestionar el modelo legislativo, utilizando una estructura adicional de grafo.
- Desafío Principal: Crear una solución que permita:
- Gestionar los congresistas y sus relaciones.
- Mantener la consistencia de los datos en una estructura dinámica que pueda adaptarse a cambios.
- Integrar herramientas que permitan analizar las conexiones entre congresistas.



Análisis del Problema

Problema



- Contexto : El sistema legislativo chileno maneja grandes volúmenes de datos relacionados con congresistas, proyectos de ley y votaciones. Para tomar decisiones es esencial contar con herramientas que gestionen y analicen estas relaciones y datos. Esto requiere un sistema que permita almacenar, consultar y procesar la información de manera accesible.
- Problema : Implementar un sistema de gestión que permita registrar y consultar datos legislativos fácilmente.
- Solución : Crear una aplicación en lenguaje C que maneje estos datos mediante estructuras adecuadas y den sustento al modelo legislativo.

Estructura de Datos (Solución)



Congresista: RUT, nombre, partido y ocupación.



Proyecto Ley: ID único, nombre, tipo, urgencia, fase y votaciones.



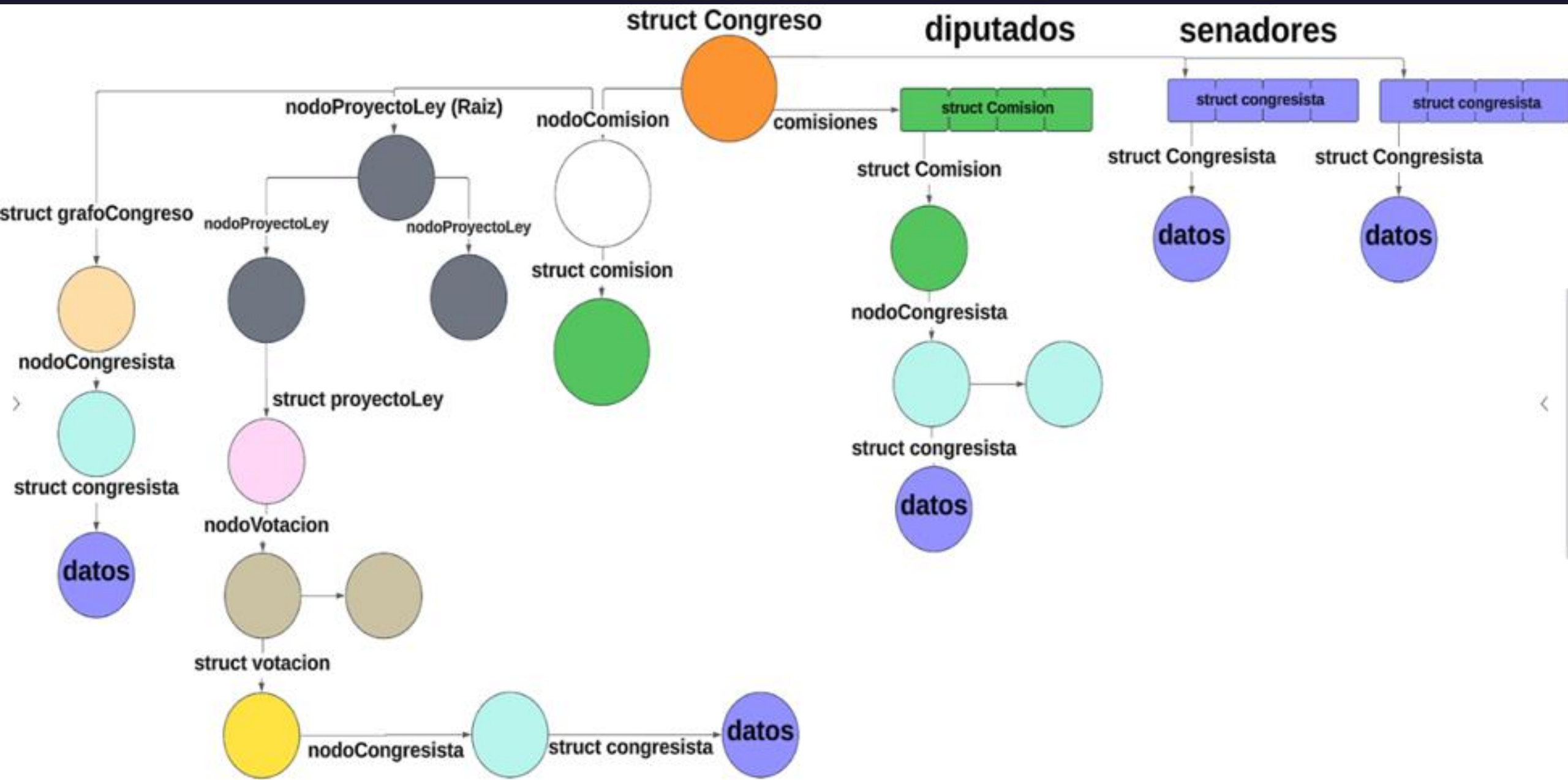
Grafo: Contiene una lista de los congresistas, n° congresistas y matriz adyacencia.



Comision: Grupo de parlamentarios encargados de analizar y discutir los proyectos.



Votación: Registra los detalles y resultados de las votaciones.



Funciones Básicas

Crear

Agregar

buscar

eliminar

mostrar



Funciones Propias Del Proyecto



Conectar congresistas en el grafo utilizando matriz



Gestión de votos en proyectos de ley, permitiendo registrar si un congresista aprueba, rechaza o se abstiene.



Ordenar congresistas por rut o nombre (método Shell)

Conclusión

Pruebas continuas para validar la robustez del sistema frente a entradas inesperadas (Lección)

El modularidad del código facilita su comprensión y mantenimiento (Fortaleza)

Ideas para mejoras futuras:
Ampliación del sistema, agregar nuevas funcionalidades.

Mejorar la validación de datos introducidos.

Mejorar la organización del menú.