

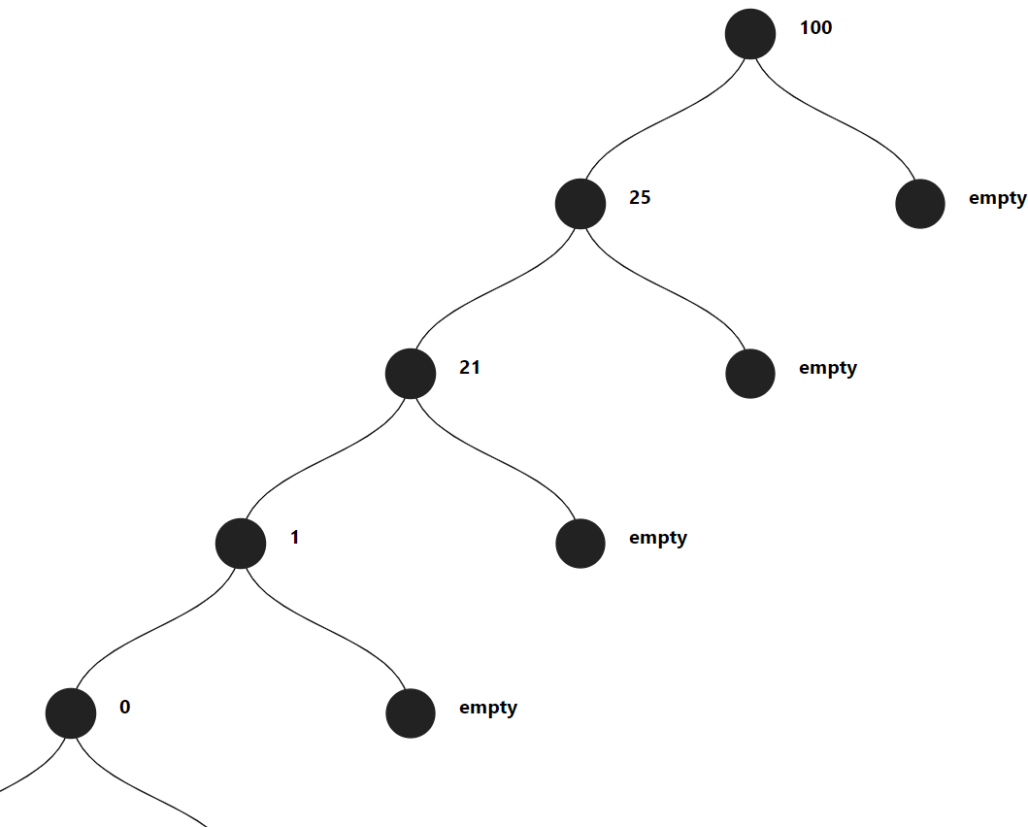
MANUAL DEL PROGRAMADOR - Proyecto Árbol Binario de Búsqueda Máximo

Autores

- Alejandro Soto Aguirre 21550303
- Azul Siret Cordero 21550381
- Naomi Regina Garcia Molina 21550292

05 de Diciembre de 2024

Version 1.0



Descripción

Este proyecto implementa un Árbol Binario de Búsqueda (ABB) en Prolog, con una interfaz de cliente en React. El objetivo es proporcionar una herramienta educativa para entender y manipular árboles binarios de búsqueda.

Alcance

El manual cubre la implementación del ABB en Prolog, así como la interfaz de cliente en React. Se incluyen ejemplos de uso y explicaciones detalladas de cada función. Esta dirigido a programadores y estudiantes de sistemas computacionales y/o carreras de desarrollo de software interesados en aprender sobre árboles binarios de búsqueda.

Estructura del Proyecto

- **abb-maximo:** Contiene el backend de la aplicacion. Tiene la logica del árbol binario de búsqueda, ademas de la implementación de las funciones de inserción, búsqueda y eliminación de nodos. Aqui tambien se levanta el servidor.
- **react-cliente:** Contiene el frontend de la aplicacion, conlleva todos los componentes de la interfaz y el servicios que consume el backend.
 - **src:** Contiene la estructura de la aplicacion, incluyendo los componentes y servicios.
 - **components:** Contiene todos los componentes de la aplicacion.
 - **services:** Contiene la api para consumir el backend.
 - **styles:** Contiene las hojas de estilo de la aplicacion.

Características

- **Inserción de valores:** Permite insertar nuevos valores en el árbol.
- **Eliminación de valores:** Permite eliminar valores existentes del árbol.
- **Actualización de valores:** Permite cambiar un valor existente por uno nuevo.
- **Visualización del árbol:** Muestra el árbol actual en formato JSON.
- **Vaciado del árbol:** Permite vaciar el árbol completamente.

Tecnologías Utilizadas

- **Prolog:** Para la lógica del árbol binario de búsqueda.
- **React:** Para la interfaz de usuario.
- **Vite:** Para el desarrollo y construcción del proyecto React.
- **SWI-Prolog:** Servidor HTTP para manejar las solicitudes del cliente.

Requisitos previos

- Tener instalado Node.js y npm.
- Contar con un editor de condigo.
- Tener instalado el paquete SWI-Prolog.

Instalación

1. Clona el repositorio:

```
git clone https://github.com/Achellx/ProyectoProlog.git
cd ProyectoProlog
```

2. Instala las dependencias del cliente React:

```
cd react-cliente
npm install
```

3. Inicia el servidor Prolog:

```
swipl -s server.pl
```

4. Inicia el cliente React:

```
npm run dev
```

Ejecución de microservicios

Este proyecto contiene la dependencia directa de un microservicio de Prolog y 1 terminal de Node para la ejecución, pues se ejecuta el entorno del Frontend y el Backend.

Ejecución del API y Frontend

Para ejecutar servidor y cliente, sigue los siguientes pasos:

1. Asegurate de que las dependencias estén instaladas en el directorio **root de este proyecto**.
2. Ejecuta el servidor de Prolog navegando al directorio **abb-maximo**:

```
swipl -s server.pl
```

3. Para ejecutar el Frontend, navega a la carpeta **react-cliente**:

```
npm run dev
```

Estos pasos te permitirán tener la API y el Frontend en funcionamiento.

Endpoints

- **GET** **/api/mostrar**: Mostrar todos los nodos:
 - Response:

```
{"arbol": [100, [25, [21, [1, [0, [], [] ], [] ], [] ], [] ], [] ]}
```

- **POST** **/api/insertar**: Crear un nuevo nodo:
 - Request:

```
{  
  "valor" : 30  
}
```

```
}
```

- Response:

```
{
  "arbol": [100, [30, [25, [21, [1, [0, [], [] ], [] ], [] ],
  "mensaje": "Valor insertado"
}
```

- **POST** /api/cambiar: Cambiar el valor de un nodo:

- Request:

```
{
  "valor_actual": 30,
  "nuevo_valor": 31
}
```

- Response:

```
{
  "arbol": [100, [31, [25, [21, [1, [0, [], [] ], [] ], [] ],
  "mensaje": "Valor cambiado"
}
```

- **POST** /api/eliminar: Eliminar un nodo:

- Request:

```
{
  "valor" : 31
}
```

- Reponse:

```
{
  "arbol": [100, [25, [21, [1, [0, [], [] ], [] ], [] ],
  "mensaje": "Valor eliminado"
}
```

Ejemplos de Solicitudes

Insertar Valor

```
curl -X POST http://localhost:8080/api/insertar -H "Content-Type: application/json" -d '{"valor": 5}'
```

Eliminar Valor

```
curl -X POST http://localhost:8080/api/eliminar -H "Content-Type: application/json" -d '{"valor": 5}'
```

Actualizar Valor

```
curl -X POST http://localhost:8080/api/cambiar -H "Content-Type: application/json" -d '{"valor_actual": 5, "nuevo_valor": 10}'
```