UNIVERSIDAD DE COLIMA

Facultad de Telemática

Ingeniería en tecnologías del internet
Convalidación

Materia:

Estructura de datos

Docente:

“Gutiérrez Pulido Jorge Rafael”

Alumno

López Morales Karen

Pizano Saucedo Jorge Luis

Fecha de entrega

“25/11/2025“

Este proyecto fue realizado con apoyo de GeminiPro para la asistencia de subir el proyecto a github y la creación del mismo.

Tabla de contenido

a. Introducción

El presente proyecto consiste en el desarrollo de un visualizador interactivo de un Trie , una estructura de datos ampliamente utilizada para representar palabras y realizar búsquedas rápidas basadas en prefijos. Permitiendo al usuario comprender cómo se forma una palabra letra por letra, cómo se validan las combinaciones dentro de un diccionario y cómo se representan visualmente las rutas internas del Trie.

Para lograr esto, se implementó una interfaz web dinámica donde las letras del alfabeto (a–z) se presentan como opciones que el usuario puede seleccionar para construir una palabra. Conforme se avanza, el sistema valida la palabra formada comparándola con un diccionario base y se ajusta automáticamente según la ruta seleccionada.

Este visualizador está realizado completamente con HTML, CSS y JavaScript , incorporando animaciones, estructura de nodos y aristas en SVG, y un diccionario real que permite verificar palabras o prefijos válidos.

b. Objetivo

Visualizar el funcionamiento interno de un Trie de forma clara e interactiva.

Para lograrlo, el programa se propone:

- Mostrar un Trie en tiempo real, expandiendo dinámicamente los nodos de la A a la Z.
- Permitir al usuario construir palabras seleccionando letras una por una.
- Validar si la palabra formada:
    - es una coincidencia exacta del diccionario,
    - es un prefijo válido,
    - o es inválida.
- Facilitar la comprensión del concepto de **prefijo, nodo padre, nodo hijo y rutas del Trie** .
- Permitir agregar nuevas palabras al diccionario y utilizarlas inmediatamente.

El sistema está diseñado con fines **educativos y demostrativos** , especialmente útil para materiales como Estructuras de Datos, Programación o Algoritmos.

b. Desarrollo

El programa consiste en un visualizador interactivo de un Trie , una estructura de datos utilizada para almacenar palabras y recorrerlas letra por letra. El objetivo principal del desarrollo es permitir al usuario escribir o seleccionar una palabra de manera visual, verificando si pertenece al diccionario cargado y mostrando en pantalla el recorrido que realiza el sistema dentro del árbol.

Componentes principales del programa

El programa utiliza dos clases:

1. **TrieNode**
    - Representa cada nudo del árbol.
    - Guarda una letra.
    - Contiene un mapa con sus hijos (siguientes letras posibles).
    - Tiene un enlace al nodo padre para reconstruir rutas.
2. **Intentar**
    - Contiene el nodo raíz.
    - Guarda la ruta actual que va formando el usuario.
    - Controle en qué parte del árbol se encuentra el usuario.
    - Tiene funciones para expandir hijos de A a Z y reiniciar el árbol.

Estructura general del programa

El código está dividido en tres grandes partes:

1. ***HTML :*** define la interfaz gráfica, botones y contenedores donde se dibuja el Trie.
2. ***CSS :*** el estilo visual, colores, tamaños y animaciones.
3. ***JavaScript :*** contiene toda la lógica del Trie, la validación de palabras, la animación del árbol y la interacción con el usuario.

Explicación del código

Diccionario de palabras

El programa inicia con un objeto llamado palabras, que contiene palabras base como “agua”, “boca”, “casa”, etc., junto con sus derivados:

const palabras = {

"agua": ["aguacero","aguamarina","aguafiestas","aguacate","aguadura"],---z

};

Luego, estas palabras se convierten en un diccionario unificado llamado palabrasBase, más fácil de consultar durante la validación.

Clase TrieNode

class TrieNode {

constructor(letter = "") {

this.letter = letter;

this.children = new Map();

this.parent = null;}}

Cada nodo del Trie contiene:

- *una letra,*
- *un conjunto de hijos,*
- *un puntero al nodo padre para reconstruir rutas*.

Clase Trie

class Trie {

constructor() {

this.root = new TrieNode();

this.current = this.root;

this.selectedPath = [];}

Esta clase administra:

- *el nodo raíz,*
- *el nodo actual donde se encuentra el usuario,*
- *la palabra que se va formando.*

Validación de palabras

El sistema revisa cada letra que selecciona el usuario para decidir si la palabra:

- es válida completa,
- es solo un prefijo válido,
- o es inválida.

La validación compara lo que lleva escrito el usuario con el diccionario:

```
const esPalabraExacta =
palabrasBase[palabraFormada] ||
Object.values(palabrasBase).flat().includes(palabraFormada);
```

Si no es válido, se reinicia el Trie y se muestra un mensaje de error.

Dibujado del Trie (SVG)

El árbol se dibuja dentro de una etiqueta <*svg*>.
La función principal es:

```
function renderFocused(svg, trie)
```

Esta función:

1. Dibuja la columna vertical desde la raíz hasta la letra actual.
2. Calcula posiciones para representar nodos hijos en la parte inferior.
3. Dibuja líneas que conectan cada nodo.
4. Muestra los nodos A–Z para que el usuario siga escribiendo.

además, se ajusta tamaños, separaciones, escalas y animaciones según la longitud de la palabra en construcción.

Manejo de clics en las letras

Cada nudo de letra (a–z) tiene un evento:

```
g.onclick = (ev) => {
...
}
```

Cuando el usuario selecciona una letra:

1. *Se anima el nodo*.
2. *Se agrega la letra a trie.selectedPath*.
3. *Se valida la palabra*.
4. *Si es válido o prefijo → se avanza al siguiente nodo*.
5. *Si es inválida → se reinicia el árbol*.
6. *Se vuelve a dibujar las opciones A–Z*.

Es el núcleo interactivo del visualizador.

Entrada de palabras nuevas

El usuario también puede agregar palabras mediante un ingreso de texto:

```
document.getElementById("addWord").onclick = () => { ... }
```

El sistema:

- *valida que la palabra solo tenga letras,*
- *toma las primeras 3 letras para clasificarla,*
- *agrega la nueva palabra al diccionario,*
- *reinicia el Trie para permitir buscarla*.

Funcionamiento desde el punto de vista del usuario

El usuario interactúa de forma muy visual:

1. Elige una letra entre las 26 disponibles.
2. El programa construye la palabra paso a paso.
3. Puede ver:
    - la palabra actual,
    - los nodos visitados,
    - los hijos posibles.
4. Si la palabra es válida, recibe un aviso de confirmación.
5. Si es inválida, el sistema se reinicia.
6. Puede agregar nuevas palabras y luego buscarlas de inmediato.

Este funcionamiento permite comprender de forma intuitiva cómo se navega dentro de un Trie real.

c. Conclusiones

Este programa permite entender de una forma clara cómo funciona un Trie y cómo se construyen las palabras paso a paso. Gracias a su visualización, el usuario puede ver cómo cada letra crea un nuevo nivel del árbol y cómo se validan las palabras del diccionario. También ofrece la opción de agregar nuevas palabras, lo que hace el sistema más flexible y fácil de usar. En general, el proyecto cumple con su objetivo: explicar el funcionamiento de un Trie de manera práctica, visual y sencilla para cualquier usuario.