

**UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA**

**CENTRO UNIVERSITARIO RETALHULEU**

**CURSO DE PROGRAMACIÓN III**

**CATEDRÁTICO ING. JORGE SALVADOR SANTOS**

# Diccionario en ABB

---

ALUMNO: ELMER UBALDO VICENTE PELICÓ

CARNE: 2890-21-17680

FECHA: 2-06-2023

# Introducción

---

El sistema de diccionario utilizando Árboles Binarios de Búsqueda (ABB) es una solución eficiente y organizada para almacenar y buscar palabras y sus definiciones. Este sistema se basa en la estructura de un árbol binario, donde cada nodo contiene una palabra y su respectiva definición. A través de operaciones de inserción, búsqueda, eliminación y actualización, este sistema proporciona una forma eficaz de gestionar y acceder a información lingüística de manera rápida y precisa.

# Justificación

---

Realizo esta investigación sobre la implementación del sistema de diccionario utilizando árboles binarios de búsqueda para comprender el propósito y la utilidad de este sistema y cómo puede beneficiar a los usuarios. Además, como estudiante, estoy motivado por aplicar mis habilidades de programación y aprender sobre estructuras de datos enfocándome en árboles binarios de búsqueda .

Se busca comprender el propósito y la utilidad de este sistema, identificar sus beneficios y aportes a la solución del problema, así como los beneficios sociales que brinda. Esta investigación me permite explorar cómo esta implementación puede mejorar la búsqueda y el acceso a las palabras y definiciones.

# Objetivos

---

**Objetivo General:** El objetivo principal de este sistema es proporcionar una estructura de datos eficiente que permita organizar las palabras y definiciones de manera ordenada. Utilizando árboles binarios de búsqueda, las palabras se pueden almacenar de forma jerárquica, lo que facilita su búsqueda y recuperación.

**Objetivos específicos:**

- Búsqueda rápida de palabras y definiciones.
- Optimizar el rendimiento del sistema de diccionario mediante técnicas adecuadas de optimización de árboles binarios de búsqueda, como el balanceo de árboles, para garantizar tiempos de búsqueda y actualización óptimos, independientemente del tamaño del diccionario.

# Propósito

---

El propósito de la implementación del sistema de diccionario utilizando árboles binarios de búsqueda es proporcionar a los usuarios una herramienta eficiente y fácil de usar para explorar y comprender el significado de las palabras. El sistema tiene como objetivo principal ofrecer una estructura organizada y accesible que permita buscar rápidamente palabras y obtener sus definiciones de manera precisa.

Con este sistema, se busca facilitar el aprendizaje y la mejora del vocabulario y comprensión del lenguaje, brindando a los usuarios una fuente confiable de información lingüística.

# Índice

---

## 1. Diccionario en ABB

- Inserción de palabras
- Eliminación de palabra
- Editar definición de palabra

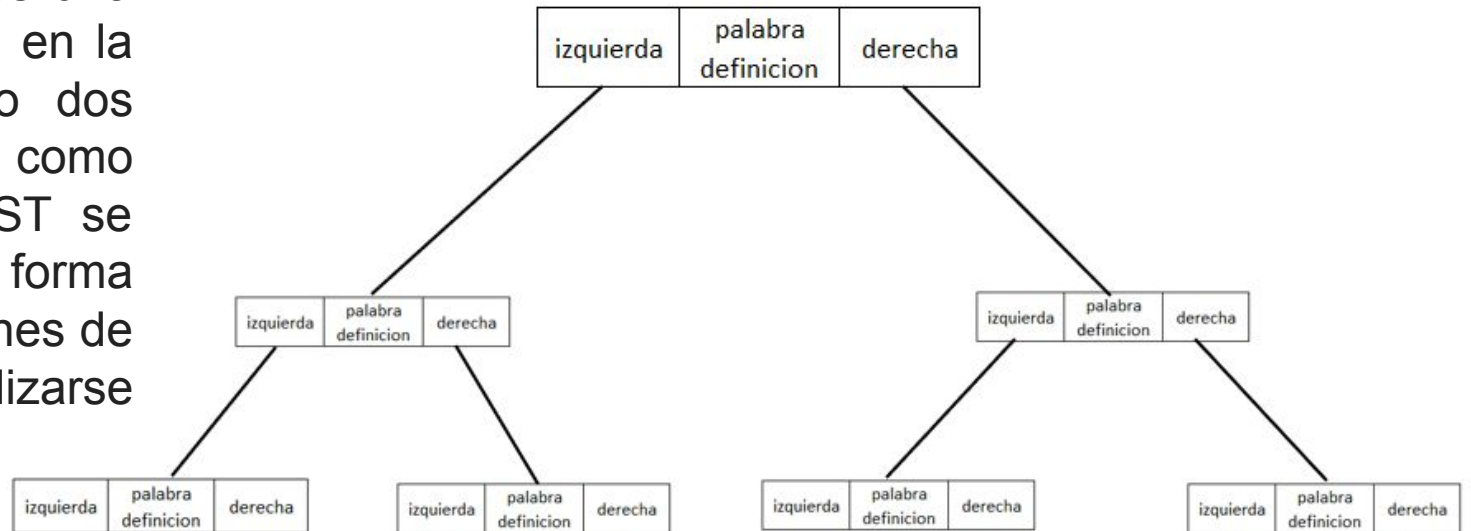
## 2. Bases de Datos

- Capas en bases de datos
- Capa de conexión
- Capa de Acceso de datos de objeto(DAO)
- Capa de servicio

# Diccionario en ABB

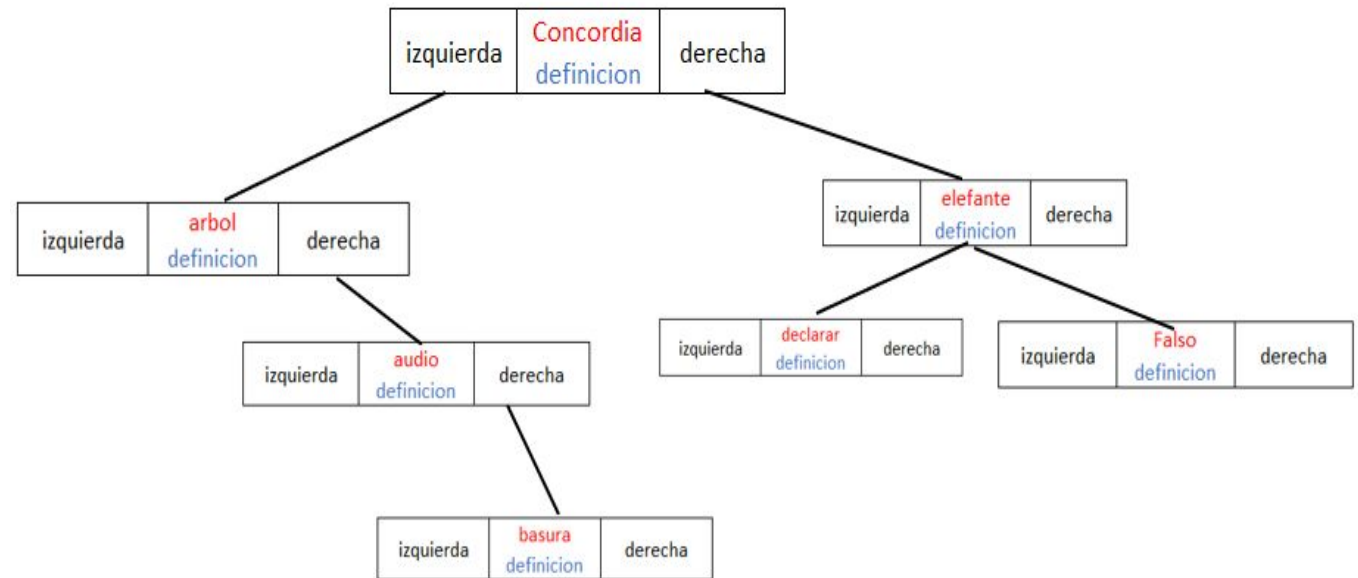
---

Un árbol de búsqueda binario (BST) es una estructura de datos en forma de árbol en la que cada nodo tiene como máximo dos nodos hijos, normalmente designados como hijo izquierdo e hijo derecho. Un BST se utiliza para almacenar datos de forma ordenada, de manera que las operaciones de búsqueda y recuperación puedan realizarse de forma eficiente.



# Búsqueda de palabra en ABB

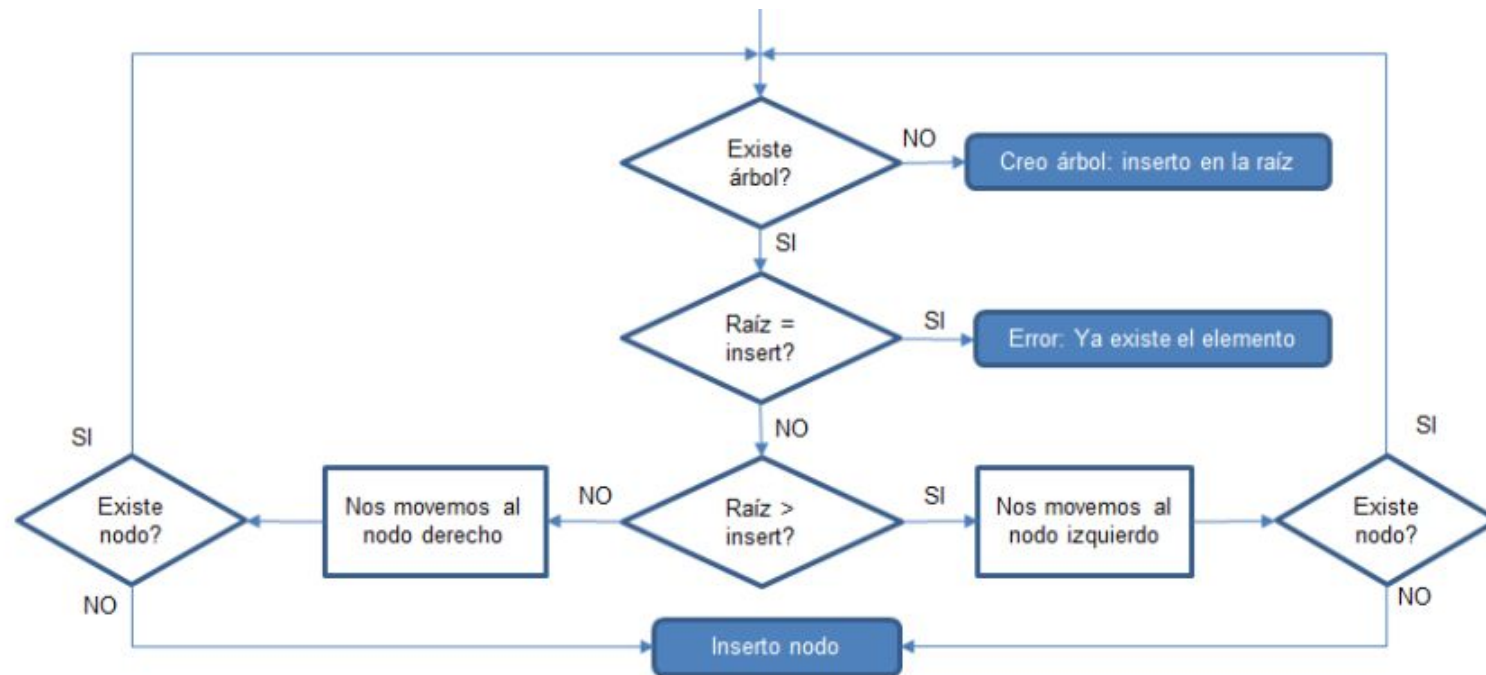
La búsqueda en un árbol binario de búsqueda consiste en acceder a la raíz del árbol, si el elemento a localizar coincide con este la búsqueda ha concluido con éxito, si el elemento es menor se busca en el subárbol izquierdo y si es mayor en el derecho. Si se alcanza un nodo hoja y el elemento no ha sido encontrado es que no existe en el árbol.





# Inserción de palabra

Para insertar un nodo en un árbol binario de búsqueda, recorreremos este de forma recursiva, y cuando lleguemos a un “hueco” libre insertaremos hay nuestro nodo.

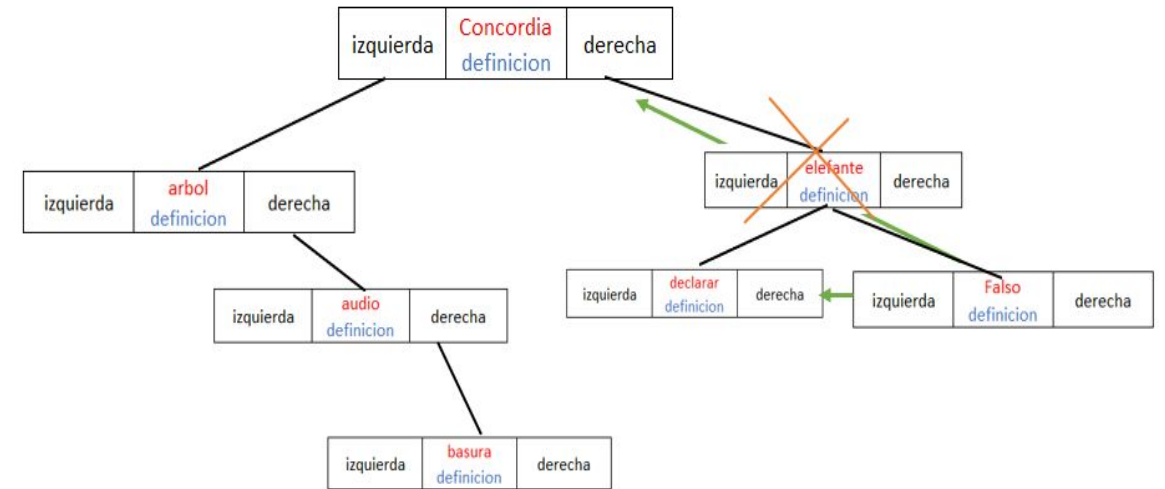


# Eliminación de palabra

Primero, buscar la palabra en el árbol. Una vez hemos localizado el nodo, tendremos que actuar de distinta manera para eliminarlo dependiendo del número de hijos que tenga. Básicamente nos puede encontrar con tres situaciones:

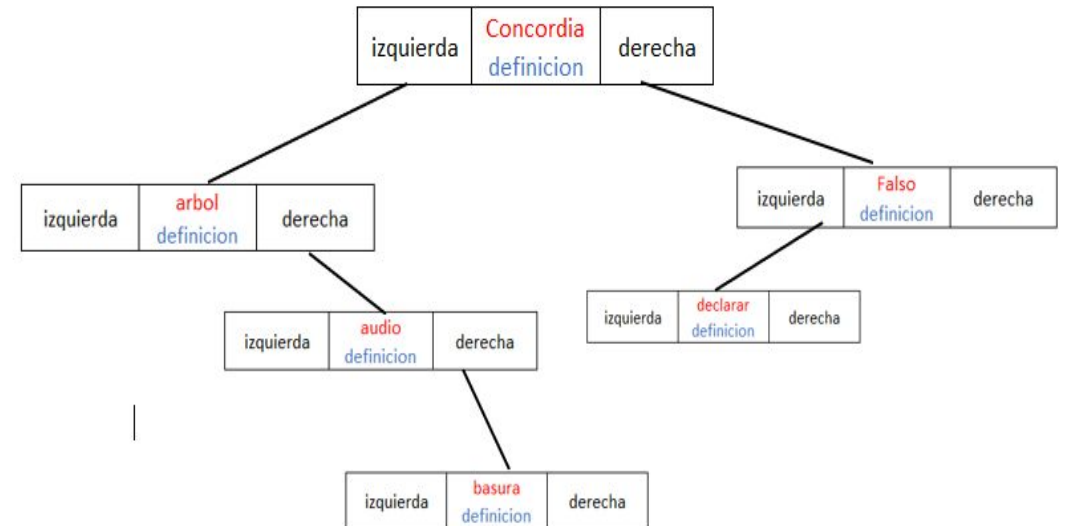
Que el nodo no tenga hijos, es una hoja: Sencillamente eliminamos el nodo y ponemos a null la referencia que tenía el padre apuntando a dicho nodo.

- Que tenga 1 hijo: Haremos que el nodo padre del nodo a eliminar apunte al único hijo que tiene el nodo a eliminar, y luego eliminaremos el nodo.
- Que tenga 2 hijos: Este es el caso más complejo. Al eliminar el nodo, ¿qué nodo de sus dos subárboles promocionamos al hueco que dejó el nodo eliminado? . Pues bien, tenemos dos opciones:



# Eliminación de palabra

1. Seleccionar del subárbol izquierdo el nodo que ocupara el sitio del nodo eliminado.  
Buscaríamos el nodo de mayor valor de todo el subárbol izquierdo, que debe ser el que se encuentre más a la derecha.
2. Seleccionar del subárbol derecho el nodo que ocupara el sitio del nodo eliminado.  
Buscaríamos el nodo de menor valor de todo el subárbol derecho, que debe ser el que se encuentre mas a la izquierda



# Editar la definición de una palabra

---

1. Comienza desde la raíz del árbol y compara la palabra deseada con el valor de "palabra" almacenado en el nodo actual.
2. Si la palabra es menor que el valor del nodo actual, realiza la búsqueda en el subárbol izquierdo.
3. Si la palabra es mayor que el valor del nodo actual, realiza la búsqueda en el subárbol derecho.
4. Si la palabra es igual al valor del nodo actual, has encontrado el nodo que contiene la palabra que deseas modificar.
5. Modifica la definición almacenada en el nodo encontrado con la nueva definición deseada.
6. Si es necesario mantener el orden del árbol binario de búsqueda por las palabras, no se necesitan ajustes adicionales, ya que solo se ha modificado la definición y no la palabra en sí.

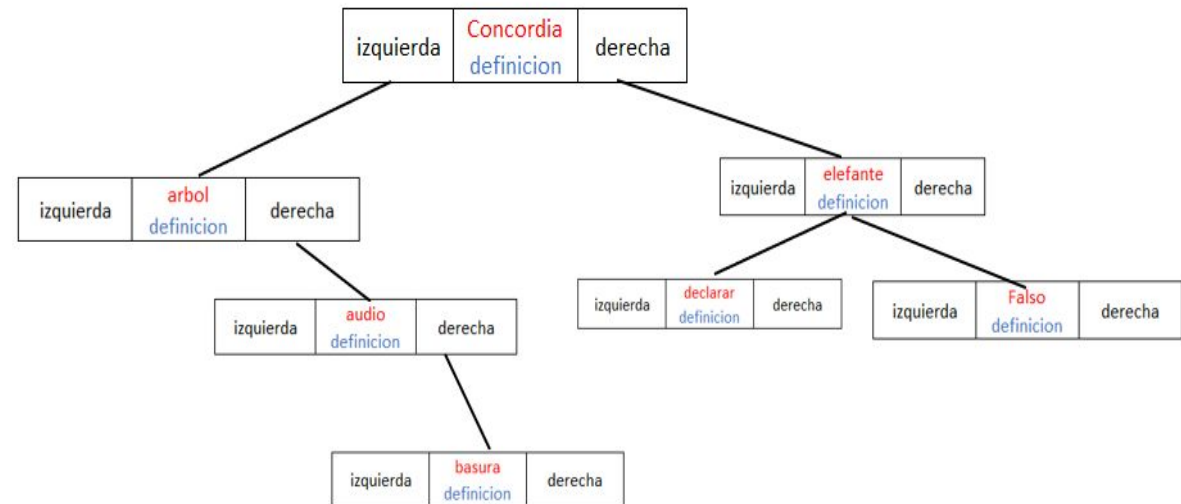
# Recorridos

**Por niveles**=(concordia-  
árbol-elefante-audio-declarar-falso-basura)

**InOrden**=(I,R,D)=(árbol-audio-basura-concordia-  
declarar-elefante-falso)

**Post orden**=(i,D,R)=(basura-audio-  
árbol-declarar-falso-elefante-concordia)

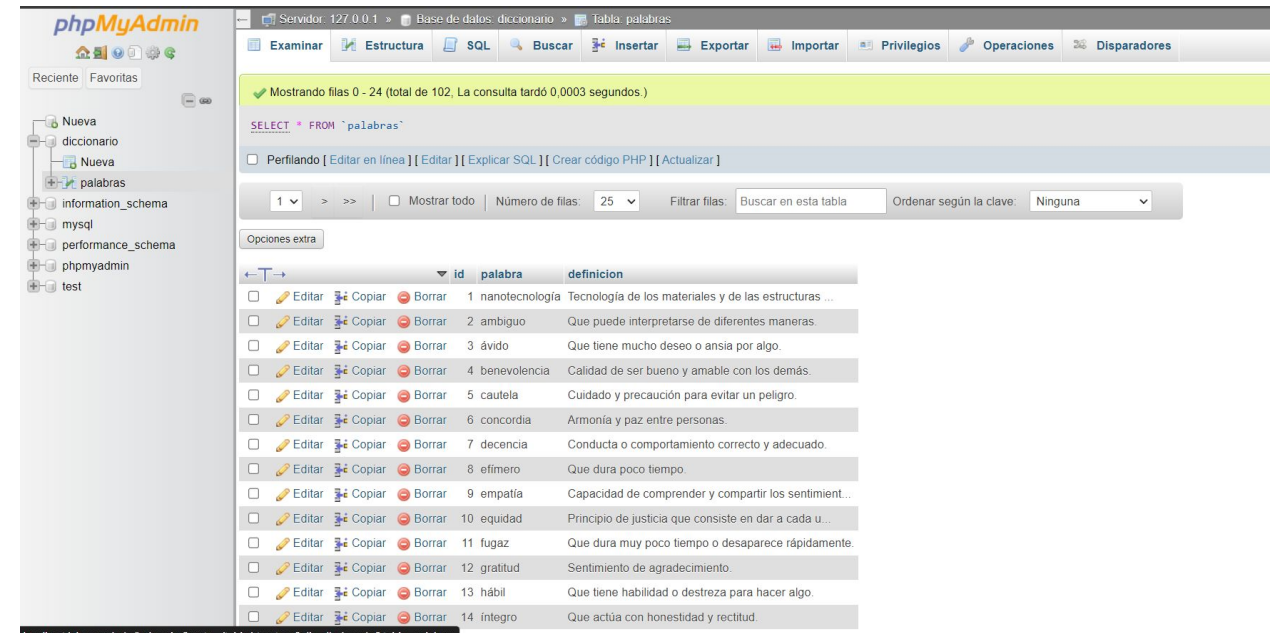
**PreOrden**=(R,I,D)=(concordia-  
árbol-audio-basura-elefante-declarar-falso)



# Tema 2: Bases de datos

En MySQL, una base de datos es un conjunto organizado de información estructurada que se almacena de forma persistente en un servidor. Una base de datos en MySQL está compuesta por una o más tablas, que contienen filas y columnas para almacenar datos de diferentes tipos.

Cada tabla en una base de datos MySQL tiene una estructura definida, que se especifica mediante la declaración de columnas con sus respectivos tipos de datos. Estas columnas representan los diferentes atributos de los datos que se van a almacenar. Por ejemplo, en una base de datos de diccionario, se puede tener una tabla "palabras" con columnas como "id", "palabra", "definicion", etc.



phpMyAdmin

Servidor: 127.0.0.1 » Base de datos: diccionario » Tabla: palabras

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Disparadores

Mostrando filas 0 - 24 (total de 102. La consulta tardó 0,0003 segundos.)

SELECT \* FROM `palabras`

Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

1 > >> ☐ Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

				id	palabra	definicion
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	1 nanotecnología Tecnología de los materiales y de las estructuras ...
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	2 ambiguo Que puede interpretarse de diferentes maneras.
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	3 ávido Que tiene mucho deseo o ansia por algo.
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	4 benevolencia Calidad de ser bueno y amable con los demás.
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	5 cautela Cuidado y precaución para evitar un peligro.
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	6 concordia Armonía y paz entre personas.
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	7 decencia Conducta o comportamiento correcto y adecuado.
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	8 efímero Que dura poco tiempo.
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	9 empatía Capacidad de comprender y compartir los sentiment...
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	10 equidad Principio de justicia que consiste en dar a cada u...
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	11 fugaz Que dura muy poco tiempo o desaparece rápidamente.
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	12 gratitud Sentimiento de agradecimiento.
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	13 hábil Que tiene habilidad o destreza para hacer algo.
<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	14 íntegro Que actúa con honestidad y rectitud.

# Capas en las bases de datos

---

En el contexto de desarrollo de software, las capas son una forma de organizar y estructurar el código en diferentes niveles de abstracción. En una arquitectura de aplicaciones basada en bases de datos, se suelen utilizar varias capas para separar las responsabilidades y mejorar la modularidad y la mantenibilidad del sistema. Algunas capas comunes en una arquitectura de bases de datos son la capa de conexión, la capa de acceso a datos (DAO) y la capa de servicio.

# Capa de conexión

---

La capa de conexión se encarga de establecer y gestionar la conexión con la base de datos. Proporciona los mecanismos para establecer la conexión inicial, autenticarse en el servidor de la base de datos y mantener la conexión abierta durante la duración de la aplicación. En esta capa, se establecen los parámetros de conexión, como la dirección IP del servidor, el nombre de usuario, la contraseña, etc.

```
1 package ConexionBD;
2
3 import java.sql.DriverManager;
4
5 /**
6  *
7  * @author PC
8  */
9
10 public class Conexion {
11     public Connection get_connection() {
12         Connection connection = null;
13
14         try {
15             connection = (Connection) DriverManager.getConnection("jdbc:mysql://localhost:3306/diccionario","root","");
16             if(connection != null){
17                 System.out.println("Conectado ");
18             }
19         } catch (SQLException e) {
20             System.out.println(e);
21         }
22
23         return (Connection) connection;
24     }
25 }
26
27
28
29
30
31
32
33
34 }
```



# Capa de Acceso a datos (DAO)

---

La capa de acceso a datos (Data Access Object, DAO) es responsable de interactuar directamente con la base de datos y realizar operaciones CRUD (crear, leer, actualizar, eliminar) en los datos. Proporciona una interfaz para realizar consultas y modificaciones en la base de datos, abstrayendo los detalles de la implementación subyacente y ofreciendo métodos más convenientes y orientados a objetos para acceder a los datos. En esta capa, se definen las consultas SQL, se ejecutan y se mapean los resultados a objetos del modelo de dominio de la aplicación.

# Capa de Servicio

---

La capa de servicio, también conocida como capa de negocio, contiene la lógica de negocio de la aplicación. Aquí se definen los servicios y operaciones que se pueden realizar en los datos, como validaciones, reglas de negocio, cálculos, etc. La capa de servicio utiliza la capa DAO para acceder a los datos necesarios y realizar las operaciones requeridas. Proporciona una interfaz más alta nivel que encapsula la lógica de negocio y puede ser consumida por otros componentes de la aplicación, como una interfaz de usuario o una API.

# Pasos para la implementación del algoritmo

---

1. Definir la estructura del nodo del árbol: Crea una clase o estructura que represente un nodo del árbol. Este nodo debe contener al menos dos atributos: la palabra y su definición. Puedes agregar otros atributos según tus necesidades, como enlaces a los nodos hijo izquierdo y derecho, y la altura del nodo.
2. Implementar la inserción de nodos: Crear un método para insertar nuevos nodos en el árbol. Este método debe seguir las reglas de un árbol binario de búsqueda, donde los nodos menores se ubican a la izquierda y los nodos mayores a la derecha. Asegúrate de mantener el balance del árbol durante la inserción.
3. Implementar la búsqueda de palabras: Crear un método para buscar una palabra en el árbol. Este método debe realizar una búsqueda recursiva comparando la palabra buscada con el nodo actual y continuando la búsqueda en el subárbol correspondiente.
4. Implementar la eliminación de nodos: Crear un método para eliminar nodos del árbol. Este método debe considerar diferentes casos, como eliminar un nodo sin hijos, eliminar un nodo con un solo hijo o eliminar un nodo con dos hijos. Asegúrate de mantener el balance del árbol durante la eliminación.
5. Implementar la actualización de definiciones: Crear un método para actualizar la definición de una palabra existente en el árbol. Este método debe buscar la palabra y actualizar su definición si se encuentra.
6. Implementar la interfaz de usuario: Crea una interfaz de usuario que Proporcione opciones para buscar palabras, agregar nuevas palabras, eliminar palabras y actualizar definiciones. Esta interfaz puede ser una salida y entrada de datos desde consola.
7. Probar el sistema.

# Descripción y funcionamiento del algoritmo

---

El sistema de diccionario es una aplicación que permite a los usuarios buscar y obtener definiciones de palabras de manera rápida y conveniente. Su objetivo principal es brindar una herramienta eficiente para explorar y comprender el significado de las palabras, ayudando así a mejorar el vocabulario y la comprensión del lenguaje.

El funcionamiento del sistema de diccionario se basa en la implementación de un árbol binario de búsqueda. Este árbol se construye utilizando las palabras y sus definiciones como datos, donde cada nodo del árbol representa una palabra y contiene su respectiva definición. La estructura del árbol facilita la búsqueda de palabras, ya que permite realizar búsquedas eficientes en tiempo logarítmico.

Al utilizar el sistema de diccionario, los usuarios pueden realizar búsquedas ingresando una palabra en particular. El sistema busca la palabra en el árbol binario de búsqueda y, si se encuentra, muestra la definición correspondiente. Si la palabra no se encuentra en el diccionario, se puede mostrar un mensaje indicando que la palabra no existe.

Además de la búsqueda de palabras, el sistema de diccionario también puede permitir otras funcionalidades, como la adición de nuevas palabras y definiciones, la eliminación de palabras existentes o la actualización de definiciones. Estas acciones pueden ser realizadas por usuarios autorizados, lo que garantiza la integridad y precisión de la información en el diccionario.

# Referencias

---

- dat science. (s. f.). Operaciones con árboles binarios de búsqueda. Recuperado de <https://dat-science.com/operaciones-con-arboles-binarios-de-busqueda/>
- Parzibyte. (2020, mayo 8). Árbol binario en Java. Recuperado de <https://parzibyte.me/blog/2020/05/08/arbol-binario-java/>
- Wikipedia. (s. f.). Recorrido de árboles. En Wikipedia, la enciclopedia libre. Recuperado de [https://es.wikipedia.org/wiki/Recorrido\\_de\\_árboles](https://es.wikipedia.org/wiki/Recorrido_de_árboles)
- Oracle. (s. f.). ¿Qué es una base de datos? Recuperado de [https://www.oracle.com/mx/database/what-is-database/#:~:text=Una%20base%20de%20datos%20es,ba ses%20de%20datos%20\(DBMS\).](https://www.oracle.com/mx/database/what-is-database/#:~:text=Una%20base%20de%20datos%20es,ba ses%20de%20datos%20(DBMS).)
- Blancarte, O. (2018, diciembre 10). Data Access Object (DAO) Pattern. Recuperado de <https://www.oscarblancarteblog.com/2018/12/10/data-access-object-dao-pattern/>