

¿Cuál es la principal desventaja de usar tablas hash sobre Árboles balanceados para un diccionario?

- a) Imposibilidad de insertar los datos ordenadamente eficientemente.
- b) Complejidad de implementación.
- c) Mayor uso de memoria en todos los casos.
- d) Peor complejidad en el caso promedio.

import java.util.HashMap;  
public class Main {  
 public static void main(String[] args) {  
 HashMap<String, Object> persona = new HashMap<>();  
 persona.put("nombre", "Ana");  
 persona.put("edad", 20);  
 System.out.println(persona.get("edad"));  
 }  
}

¿Qué imprime el programa?

- a) "edad"
- b) 20
- c) "Ana"

Se tiene una tabla hash de tamaño 7 con sondeo lineal  $(h(u) = u \bmod 7)$ . Tras insertar las claves 14, 21, 28, 35, se elimina el 21. ¿Cuántos sondeos requiere buscar el 28 posteriormente?

- a) 1 sondeo
- b) 2 sondeos
- c) 3 sondeos
- d) 4 sondeos

¿Qué almacena un Diccionario?

- a) Solo valores
- b) Solo claves
- c) Clave y valor

¿Cuál es una ventaja de implementar un diccionario con una lista frente a una tabla hash?

- a) Inserción de elementos más rápida ( $O(1)$ )
- b) Búsqueda de elementos más rápida ( $O(\log n)$ )
- c) Implementación más simple.
- d) No requiere manejo de colisiones.

¿Las claves en un Diccionario pueden repetirse?

- a) Sí
- b) No

¿Cuál es la principal ventaja de usar un árbol balanceado de búsqueda (BST) balanceado frente a una tabla hash para almacenar un diccionario?

- a) Mantiene los datos ordenados, permitiendo búsquedas por rango y recorridos ordenados.
- b) Proporciona tiempos de búsqueda constantes  $O(1)$  en todos los casos.
- c) Utiliza menos memoria que una tabla hash.
- d) No requiere balanceo para mantener su eficiencia.

En Java, ¿cuál de las siguientes clases se usa comúnmente para implementar un Diccionario?

- a) ArrayList
- b) HashMap
- c) Queue

¿Cómo se accede a un valor dentro de un Diccionario?

- a) Por índice numérico
- b) Por su clave
- c) Aleatoriamente

En un Árbol Rojo-Negro, ¿qué garantiza que la altura sea  $\max 2 \log_2(n+1)$ ?

- a) El balanceo perfecto de todos los nodos.
- b) Las propiedades de rotación y balance estables.
- c) El uso de un factor de balance por nodo.
- d) La reorganización aleatoria de los datos.

Imagina que quieres registrar los parámetros de algunos usuarios en un diccionario. ¿Qué tipo de parámetro mejor usar como clave: el nombre o la contraseña? Justifica tu respuesta.

¿Cuál de estas opciones muestra correctamente una operación básica de diccionarios con su complejidad de ejecución?

- a) Inserción: agregar un elemento a un registro de calificaciones.
- b) Ordenación: organizar los datos por valor hash.
- c) Búsqueda: incrementar el contador de visitas de un usuario.
- d) Eliminación: eliminar los datos del diccionario.

¿Por qué se dice que la búsqueda en una tabla hash tiene complejidad promedio  $O(1)$ , pero en el peor caso puede ser  $O(n)$ ?

- a) Porque el tamaño de la tabla siempre es fijo.
- b) Porque con una buena función hash los nodos son distribuidos por los muchos valores se debe recorrer muchos elementos.
- c) Porque las tablas hash se implementan sobre arrays que tienen acceso  $O(1)$ .
- d) Porque al peor caso ocurre cuando la tabla está vacía.

Si en una tabla hash dos claves distintas producen el mismo valor hash, ¿qué problema se genera y qué se usa para evitarlo?

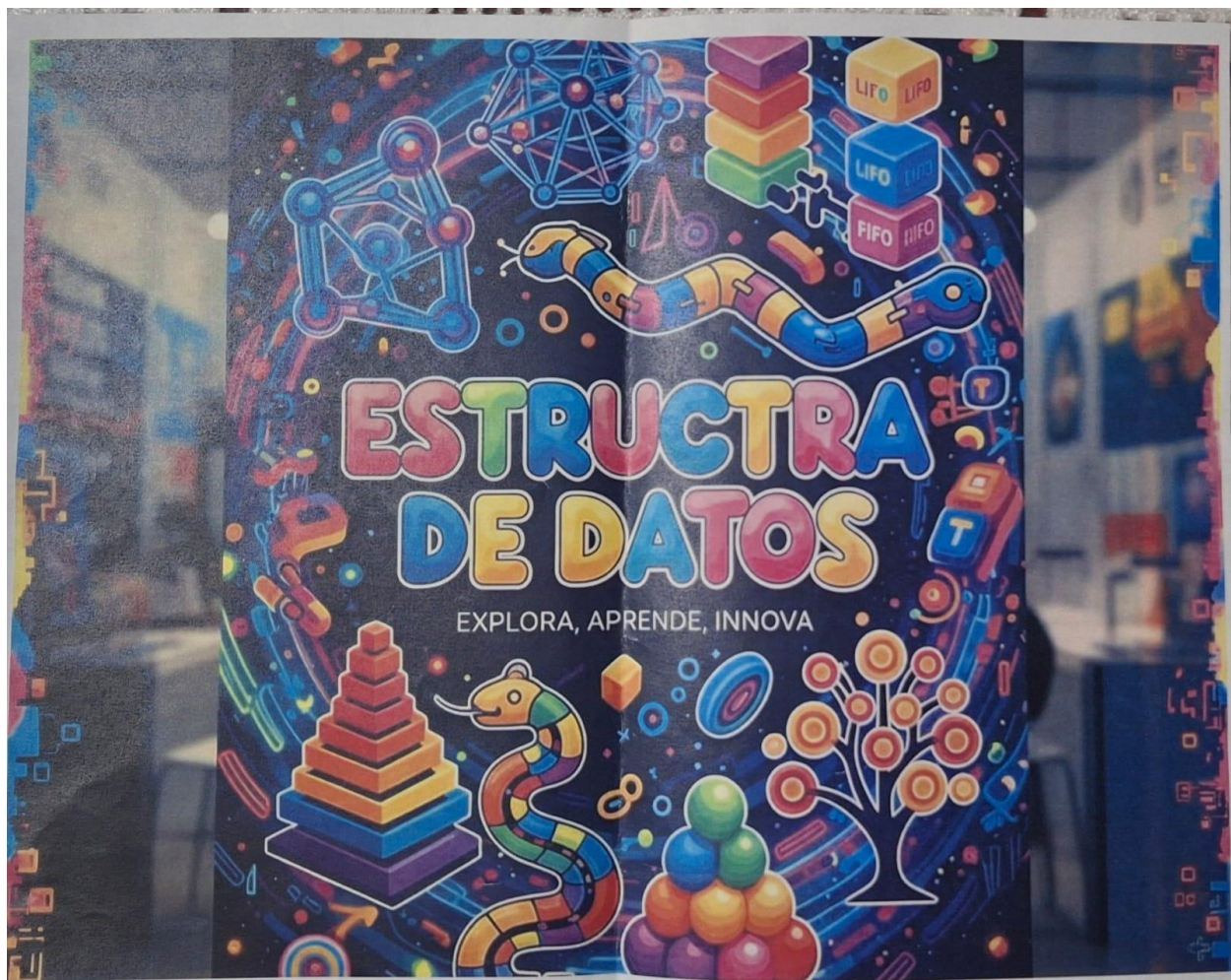
- a) Se genera una colisión, que se resuelve mediante encadenamiento (listas enlazadas).
- b) Se genera un desbalanceo de datos, que se resuelve reinsertando la tabla.
- c) Se corrompe la memoria, que se resuelve con una nueva función hash.
- d) Se genera un error de sintaxis, que se resuelve reescribiendo la siguiente clave.

En un Diccionario, las claves deben ser...

- a) Únicas
- b) Repetidas
- c) Numéricas únicamente

- 1 c) Clave y valor
- 2 b) No
- 3 b) Por su clave
- 4 b) HashMap
- 5 a) Únicas
- 6 b) 20
- 7 c) Implementación más simple.
- 8 a) Se genera una colisión.
- 9 b) Porque con una buena función hash los accesos son directos.
- 10 a) Inserción.
- 11 a) Mantener los datos ordenados.
- 12 Matricula, porque es un identificador único y no se repite
- 13 c) 3 sondeos.
- 14 a) Imposibilidad de recorrer.
- 15 b) Las propiedades.





3pts → Pila

1 pts → Arreglos

2 pts → POO

2 pts → Cola

2 pts → Conjuntos