

Taller Semana 6 - Métodos de Búsqueda

ESTUDIANTES:

ISAAC BENALCÁZAR

JUAN ESPINEL

JORGE GONZALEZ

YKER LOPEZ

MATIAS SANTILLAN

Explicación detallada del método implementado y su funcionamiento

El programa incluye una aplicación gráfica de usuario (GUI) escrita en Java utilizando la biblioteca Swing. Su tarea es buscar secuencialmente los números ingresados por el usuario en una matriz predefinida. Si hay un número en la matriz, la aplicación muestra su ubicación; de lo contrario, dice que el número no existe en la matriz. El programa incluye dos clases principales:

SearchSequentialGUI: gestiona la GUI y las interacciones del usuario. SequentialSearch: implementa un algoritmo de búsqueda secuencial. 1. Clase SearchSequentialGUI

Componente GUI

JPanel pGeneral: Este es el panel principal que contiene todos los elementos de la interfaz.

JTextField txtbuscarNumero: Campo de texto donde el usuario ingresa el número que desea

buscar. JButton searchButton: Un botón que cuando se presiona comienza a buscar el número

ingresado. Etiqueta de estado de JLabel: una etiqueta que muestra el estado o los resultados

de la búsqueda (por ejemplo, si se encontró el número y su ubicación). array int[]: una matriz

predefinida de números enteros para buscar. En este caso contiene los valores {2, 4, 6, 8, 10,

12, 14}. búsqueda GISequential pública() {

searchButton.addActionListener(nuevo ActionListener() {

@Superar

acción vacía pública realizada (ActionEvent e) {

Encuentra el número();

}

});

}

Función: vincula un objeto ActionListener al botón de búsqueda. Cuando el usuario hace clic en este botón, se ejecutará el método findNumber().

v. método findNumber().

número vacío privado findNumber() {

```

Número de línea = txtsearchNumber.getText();
probar {
int númerobúsqueda = Integer.parseInt(número);
int index = SequenceSearch.search(matriz, número de búsqueda);
si(índice!= -1){
State.setText("Número" en la posición: "índice");
} que{
Status.setText ("El número" no existe en la matriz");
}
} captura (Excepción NúmeroFormatException){
JOptionPane.showMessageDialog(null, "Ingrese un número válido");
}
}

```

Obtener Número: Obtiene el texto ingresado por el usuario en el campo txtbuscarNumero y lo almacena en una variable numérica. Convertir a número entero: intenta convertir el texto ingresado a un número entero (int searchNumber). Si el texto no es un número válido, se genera una excepción NumberFormatException. Búsqueda secuencial: llame al método de búsqueda estática de la clase SequentialSearch, pasando una matriz de matrices y el número de búsqueda a buscar. Este método devuelve el índice en el que se encontró el número, o -1 si no se encontró el número. Actualizar etiquetas de estado:

Si se encuentra un número (índice != -1): Muestra un mensaje informando sobre el número encontrado y su posición en la tabla. Si no se encuentra el número (índice == -1): informa que el número no existe en la matriz. Manejo de errores: si el usuario ingresa un valor no numérico, aparecerá un cuadro de diálogo informándole que ingrese un número válido.

d.Método manual

```

público estático vacío principal (String[] args) {;
JFrameframe = nuevo JFrame("FromSequentialSearch");
Frame.setContentPane(new SequentialSearchGUI().pGeneral);
Frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
marco.paquete();
Frame.setVisible (verdadero);
} }

```

Descripción: este método inicializa la aplicación GUI. JFrame: Crea una ventana principal llamada "GUISequentialSearch". Configuración de contenido: asigne la tabla pGeneral a la nueva instancia de SequentialSearchGUI como contenido del marco. Cerrar aplicación: configura la aplicación para que se cierre cuando se cierre la ventana. Cambiar tamaño: cambia automáticamente el tamaño de la ventana para que se ajuste a los elementos del interior. Visibilidad: Hace que la ventana sea visible para el usuario.

2. Clase de búsqueda secuencial

Buscando un método estático

Búsqueda secuencial de clase pública {

```
búsqueda estática pública int (matriz int [], valor int) {  
    para (int i = 0; i < matriz.longitud; i ){  
        si (matriz [i] == valor) {  
            vuelve a ti mismo;  
        }  
    }  
    devolver -1;  
}
```

Descripción del método:

Propósito: Implementar un algoritmo de búsqueda secuencial para encontrar un valor específico en una matriz de números enteros. Parámetro:

int[] array: Matriz a buscar. Valor int: número a encontrar en la matriz. Trabajar:

Iteración: itera la matriz de principio a fin mediante un bucle for. Comparar: en cada iteración, compare la matriz actual de elementos [i] con el valor deseado. Devolver:

Si se encuentra valor: devuelve el índice i en el que se encontró el valor en la matriz. Si no se encuentra el valor: después de recorrer toda la matriz sin encontrar un valor, se devuelve -1, lo que significa que el valor no existe en la matriz. Características de los algoritmos de búsqueda secuencial.

Simplicidad: este es uno de los algoritmos de búsqueda más simples.