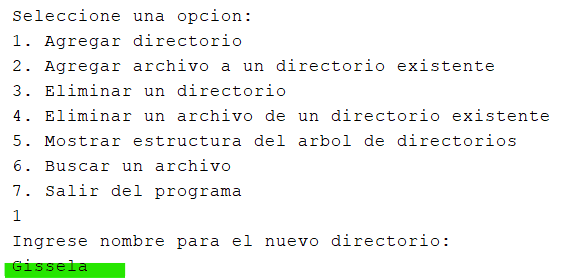
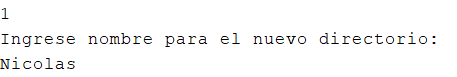
**Codigo**

|  |
| --- |
| **Directorio**  package arboldirectorios;  import java.util.LinkedList;  import java.util.Queue;  // Clase que representa un directorio en el árbol de directorios  public class Directorio {  String nombre; // Nombre del directorio  LinkedList<Directorio> subdirectorios; // Lista de subdirectorios  LinkedList<String> archivos; // Lista de archivos en el directorio  // Constructor para inicializar un directorio con un nombre dado  public Directorio(String nombre) {  this.nombre = nombre;  this.subdirectorios = new LinkedList<>();  this.archivos = new LinkedList<>();  }  // Método para agregar un nuevo directorio al directorio actual  public void agregarDirectorio(Directorio directorio) {  //agregar un nuevo directorio al directorio actual.  subdirectorios.add(directorio);//agrega el nuevo directorio a la lista de subdirectorios del directorio actual usando el método add  }  // Método para agregar un nuevo archivo al directorio actual  public void agregarArchivo(String archivo) {  archivos.add(archivo);  }  // Método para eliminar un directorio del directorio actual por nombre  public void eliminarDirectorio(String nombre) {  //agregar un nuevo archivo al directorio actual.  subdirectorios.removeIf(d -> d.nombre.equals(nombre));//agrega el nuevo archivo a la lista de archivos del directorio actual usando el método add  }  // Método para eliminar un archivo del directorio actual por nombre  public void eliminarArchivo(String archivo) {  //eliminar un directorio del directorio actual por nombre.  archivos.remove(archivo);  }  // Método para mostrar la estructura del árbol de directorios  public void mostrarDirectorio() { //recorrido por niveles  Queue<Directorio> cola = new LinkedList<>();//Comienza en el directorio actual y agrega cada directorio a la cola.  cola.offer(this);  while (!cola.isEmpty()) {  Directorio actual = cola.poll();  System.out.println(actual.nombre + "/");  for (String archivo : actual.archivos) {  System.out.println("-- " + archivo);  //retira un directorio de la cola, imprime su nombre y luego agrega todos sus subdirectorios a la cola  }  for (Directorio subdir : actual.subdirectorios) {  cola.offer(subdir);  }  }  }  // Método para buscar un archivo en el árbol de directorios  public boolean buscarArchivo(String archivo) {  Queue<Directorio> cola = new LinkedList<>(); //para recorrer todos los directorios y archivos en el árbol.  cola.offer(this); //para realizar el recorrido por niveles.    while (!cola.isEmpty()) { //Comienza en el directorio actual y agrega cada directorio a la cola.  Directorio actual = cola.poll();  if (actual.archivos.contains(archivo)) {  System.out.println("El archivo \"" + archivo + "\" fue encontrado en el directorio \"" + actual.nombre + "\" ");  return true;  }//retira un directorio de la cola y verifica si la lista de archivos contiene el archivo buscado.  for (Directorio subdir : actual.subdirectorios) {  cola.offer(subdir);  }  }  //Si termina de recorrer el árbol sin encontrar el archivo, devuelve false.  return false;  }  } |
| **Principal**  package arboldirectorios;  import java.util.Scanner;  public class ArbolDirectorios {  public static void main(String[] args) {  // TODO code application logic here  Scanner scanner = new Scanner(System.in);  Directorio raiz = new Directorio("Raiz"); // Crear el directorio raíz  boolean salir = false;  while (!salir) { //cuando se de la opcion 7 se saldra del programa directamente  // Mostrar el menú de opciones al usuario  System.out.println("Seleccione una opcion:");  System.out.println("1. Agregar directorio");  System.out.println("2. Agregar archivo a un directorio existente");  System.out.println("3. Eliminar un directorio");  System.out.println("4. Eliminar un archivo de un directorio existente");  System.out.println("5. Mostrar estructura del arbol de directorios");  System.out.println("6. Buscar un archivo");  System.out.println("7. Salir del programa");  int opcion = scanner.nextInt(); // Leer la opción seleccionada por el usuario  scanner.nextLine(); // Consumir el salto de línea después del entero  switch (opcion) {  case 1: // Agregar un nuevo directorio  System.out.println("Ingrese nombre para el nuevo directorio:");  String nombreDirectorio = scanner.nextLine();  raiz.agregarDirectorio(new Directorio(nombreDirectorio));  break;  case 2: // Agregar un archivo a un directorio existente  System.out.println("Ingrese el nombre del directorio existente:");  String nombreDir = scanner.nextLine(); //para ingresar el nuevo nombre  Directorio directorioExistente = null;  for (Directorio dir : raiz.subdirectorios) {  if (dir.nombre.equals(nombreDir)) { //busca el directorio  directorioExistente = dir;  break;  }  }  if (directorioExistente != null) {//en el directorio existente pide el ingreso del nombre del archivo  System.out.println("Ingrese el nombre del nuevo archivo:");  String nombreArchivo = scanner.nextLine(); //ingreso el nombre  directorioExistente.agregarArchivo(nombreArchivo);//Se agrega el archivo al directorio  System.out.println("Nuevo archivo agregado");  } else {  System.out.println("El directorio especificado no existe."); //si el directorio no existe muestra este mensaje  }  break;  case 3: // Eliminar un directorio  System.out.println("Ingrese el nombre del directorio a eliminar:"); //pide el nombre del directorio para eliminar  String nombreEliminar = scanner.nextLine();  raiz.eliminarDirectorio(nombreEliminar);//buscar el directorio y lo elimina  System.out.println("Directorio eliminado");  break;  case 4: // Eliminar un archivo de un directorio existente  System.out.println("Ingrese el nombre del directorio donde se encuentra el archivo:");  String nombreDirEliminar = scanner.nextLine(); // ingresa el nombre del direectorio  Directorio dirEliminar = null;  for (Directorio dir : raiz.subdirectorios) { //busca el directorio  if (dir.nombre.equals(nombreDirEliminar)) {  dirEliminar = dir;  break;  }  }  if (dirEliminar != null) { //recorrido para eliminar el archivo  System.out.println("Ingrese el nombre del archivo a eliminar:");  String archivoEliminar = scanner.nextLine(); //ingrso del nombre del archivo a elimianr  dirEliminar.eliminarArchivo(archivoEliminar);//elimina el archivo  System.out.println("Archivo eliminado");  } else {  System.out.println("El directorio especificado no existe.");// si no existe el directorio mustra este mensaje  }  break;  case 5: // Mostrar la estructura del árbol de directorios  System.out.println("Estructura del arbol de directorios:");  raiz.mostrarDirectorio();//muestra la extructura del arbol crea con todos los datos actualizados  break;  case 6: // Buscar un archivo en el árbol de directorios  System.out.println("Ingrese el nombre del archivo a buscar:");  String nombreArchivoBuscar = scanner.nextLine(); //busca el nombre del archivo  if (raiz.buscarArchivo(nombreArchivoBuscar)) {    System.out.println("El archivo \"" + nombreArchivoBuscar + "\" fue encontrado.");    } else {  System.out.println("El archivo \"" + nombreArchivoBuscar + "\" no fue encontrado.");  }  break;  case 7: // Salir del programa  salir = true;  break;  default: // Opción no válida  System.out.println("Opcion no valida. Por favor, ingrese un numero valido.");  }  }  scanner.close(); // Cerrar el scanner al salir del programa  }  } |

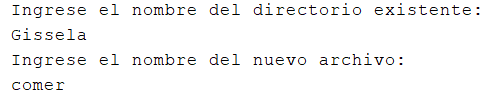
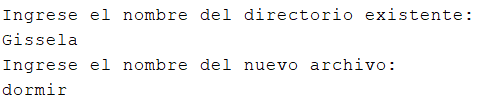
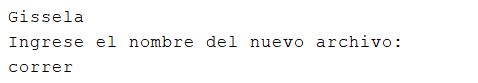
Capturas Ejecucion

**Opcion 1 : agregar directorio**

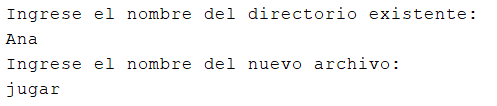
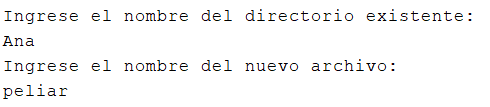
  

* **Relice el ingreso de 3 directorio Gissela, Ana, Nicolas**

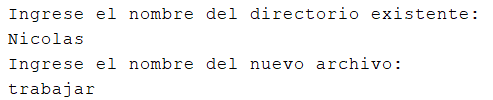
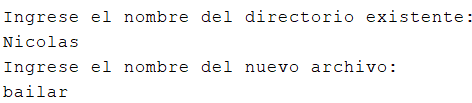
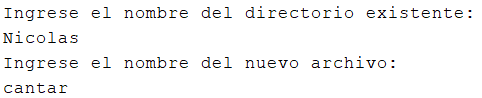
**Opcion 2: agregar archivo**

  ****

* **Ingrese 3 archivos nuevos al directorio Gissela**

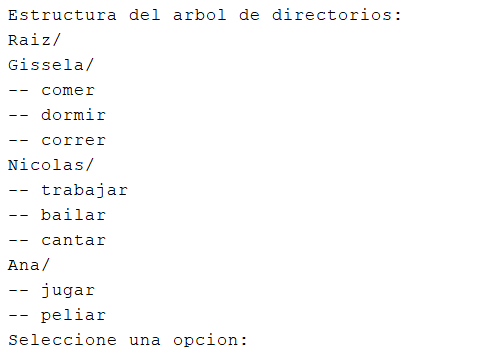
 

* **Ingrese 2 archivos nuevos al directorio Ana**

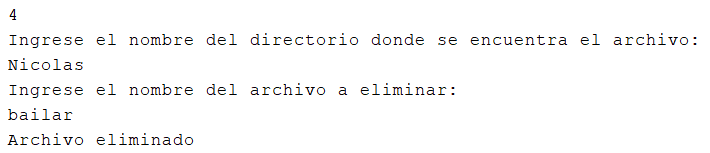
* **Ingrese 3 archivos nuevos al directorio Nicolas**

**Opcion 5: mostrar**



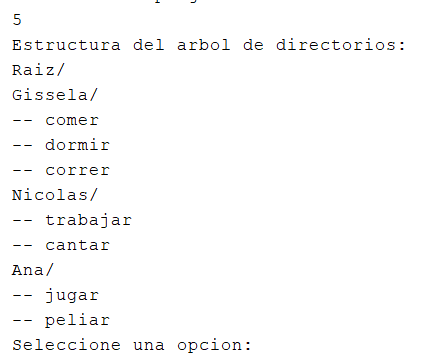
* **Muestro el arbol depues de ingresar directorios y archivos**

**Opción 4: eliminar archivo**



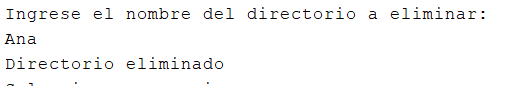
* **Ingreso al directorio Nicolas para eliminar un archivo que es Bailar**

**Opción 5: archivo eliminado**



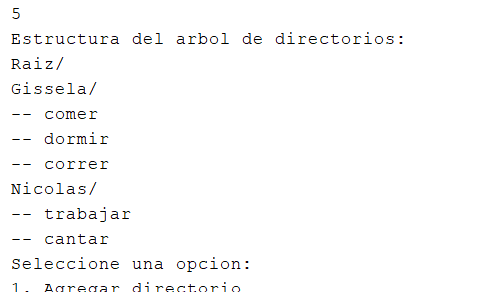
* **Muestro que en el directorio Nicolas se elimino el archivo Bailar**

Opción 3 eliminar directorio:



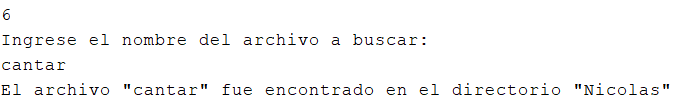
* **Ingreeso el directorio que voy a eliminar**

**Opción 5 mostra directorio eliminado:**



* **Muestro que el directorio se elimino**

**Opción 6: buscar archivo**



* **Ingreso el nombre del archivo y los busca indicandome en que directorio esta**