**Estudiante: Juan José Goyes Zambrano**

|  |  |  |
| --- | --- | --- |
| **Objetivos**   1. Recordar el proceso de carga de información al mundo desde archivos de datos 2. Generar reportes de información del mundo en archivos de datos. 3. Practicar el refresco de información en la interfaz de la aplicación luego de cargar información nueva | **Número de participantes**  participante.jpg | **Materiales**   * Internet * Eclipse |
| **Duración de la actividad**  **01:30** |
| **FASE 1.**   1. Se requiere construir un aplicativo que permita gestionar la información de un estudiante en una Universidad X, la información que se requiere del estudiante es el nombre, el apellido, el semestre, correo electrónico y el celular. Para ello la gestión debe permitir consultar los estudiantes, agregar un nuevo estudiante, modificar los datos del estudiante y eliminar un estudiante. El proyecto requiere un almacenamiento en la memoria del computador. El sistema debe permitir a la luz de un menú que el usuario elija la opción que desee, en esta primera fase se trabaja con la consola de java. | | |
| **Requerimientos funcionales**  Diligencie la información de los requerimientos funcionales del sistema solicitado.   |  |  | | --- | --- | | **RF1** | **Agregar estudiante** | | **Entradas** | **Nombre, apellido, semestre, correo, celular** | | **Descripción** | **Se reciben y se validan los datos proporcionados para crear un nuevo registro y almacenarlo** | | **Resultado** | **Se ingresa el estudiante a la base de datos y puede ser consultado junto con los demás** |  |  |  | | --- | --- | | **RF2** | **Consultar estudiante** | | **Entradas** | **NA** | | **Descripción** | **El programa presenta un listado de los estudiantes registrados** | | **Resultado** | **Se puede ver el listado de estudiantes** |  |  |  | | --- | --- | | **RF3** | **Modificación estudiante** | | **Entradas** | **Nombre o ID del estudiante** | | **Descripción** | **Se busca y se actualizan/modifican los datos** | | **Resultado** | **Se actualizan o modifican los datos** |  |  |  | | --- | --- | | **RF4** | **Eliminar estudiante** | | **Entradas** | **Nombre o ID del estudiante** | | **Descripción** | **Se busca y se elimina de la base de datos** | | **Resultado** | **Se elimina el estudiante** | | | |
| **Diagrama de clases** | | |
| Realice el diagrama de clases teniendo en cuenta la clase Alumno y la clase proyecto | | |
| **Programación** | | |
| 1. Lo primero que va a realizar es un menú de opciones para el usuario, presentando en pantalla para que el usuario pueda elegir.   Utilice la función System.out.println(String); para construirlo    Pregunte por la opción que escoge el usuario a partir de un sistema de condicionales if elseif       1. Para que quede en un bucle debe colocar el código en una sentencia do – while e incorporar una bandera para terminar el programa, esta variable se llama activo y puede ser un tipo boolean. 2. Para tomar la elección del usuario se debe utilizar una variable de tipo Scanner llamada “lector” y luego validarla con los if – else if. 3. Comience por capturar la información del estudiante en la primera opción      1. Debe utilizar un ArrayList para almacenar la información de cada alumno. | | |
| Comience a documentar el ejercicio a manera de tutorial, de tal forma que cualquier persona pueda entender el ejercicio. | | |
| import java.util.ArrayList;  import java.util.Scanner;  /\*\*  \*  \* @author Juan Goyes  \*/  public class Main {  public static void main(String[] args) {  //funcion que permite leer la opcion del usuario  Scanner lector = new Scanner(System.in);  //bandera que permite terminar el programa  boolean activo = true;  //array para guardar los estudiantes registrados  ArrayList<Estudiante> listadoEstudiantes = new ArrayList<Estudiante>();  do {  System.out.println("=============== Menu de opciones ============");  System.out.println("1: Insertar Estudiante");  System.out.println("2: Eliminar Estudiante");  System.out.println("3: Modificar Estudiante");  System.out.println("4: Consultar Estudiantes");  System.out.println("5: Terminar programa");  int opcion = lector.nextInt();  switch (opcion) {  case 1:  System.out.println("=========Sección ========");  System.out.println("Agregar estudiante");  agregarEstudiante(listadoEstudiantes, lector);  break;  //System.out.println("se agrego un nuevo estudiante con los sifuientes datos \n Nombre: " + a.getNombre() + "\n Apellido: " + a.getApellido() + "\n Cedula: " + a.getCedula() + "\n Semestre: " + a.getSemestre() + "\n Correo: " +a.getCorreo() + "\n Celular: " + a.getCelular());  case 2:  System.out.println("=========Sección ========");  System.out.println("Eliminar Estudiante");  eliminarEstudiante(listadoEstudiantes, lector);  break;  case 3:  System.out.println("=========Sección ========");  System.out.println("Modificar estudiante ");  modificarEstudiante(listadoEstudiantes, lector);  break;  case 4:  System.out.println("Consultar estudiantes");  System.out.println("");  mostrarEstudiantes(listadoEstudiantes);  break;  case 5:  System.out.println("Terminar");  activo = false;  break;  default:  System.out.println("Debe seleccionar una de las opciones del menú");  }  } while (activo);  lector.close();  }  public static void agregarEstudiante(ArrayList<Estudiante> registroEstudiantes, Scanner lector) {  //Crear un nuevo estudainte desde la clase Estudiante  Estudiante registroNuevoEstudiante = new Estudiante();  //Lectura de la cedula del estudiante  System.out.println("Ingrese la cedula del estudiante");  registroNuevoEstudiante.setCedula(lector.nextInt());  //lectura el nombre del estudiante  System.out.println("Ingrese el nombre del estudiante");  registroNuevoEstudiante.setNombre(lector.nextLine());  //lectura del apellido del estudiante  System.out.println("Ingrese el apellido del estudiante");  registroNuevoEstudiante.setApellido(lector.nextLine().trim());  //lectura del semestre del estudiante  System.out.println("Ingrese el semestre del estudiante");  registroNuevoEstudiante.setSemestre(lector.nextInt());  //lectura del correo del estudiante  System.out.println("Ingrese el correo del estudiante");  registroNuevoEstudiante.setCorreo(lector.nextLine().trim());  //lectura del numero de celular del estudiante  System.out.println("Ingrese el numero del estudiante");  registroNuevoEstudiante.setCelular(lector.nextInt());  //agregar estudiante al array  registroEstudiantes.add(registroNuevoEstudiante);  }  /\*\*  \* metodo para modificar alumno  \*/  public static void modificarEstudiante(ArrayList<Estudiante> listadoEstudiantes, Scanner lector) {  System.out.println("Ingrese el numero de cedula del estudiante para modificación de datos");  int modificacion = lector.nextInt();  boolean cambio = false;  for (Estudiante estudiante : listadoEstudiantes) {  if (estudiante.getCedula() == modificacion) {  System.out.println("El estudiante que modificara es " + estudiante.getNombre() + " " + estudiante.getApellido());  }  //Lectura del nombre modificado  System.out.println("Ingrese el nuevo nombre del estudiante");  String nombreModificado = lector.nextLine().trim();  estudiante.setNombre(nombreModificado);  //lectura del apellido modificado  System.out.println("Ingrese el nuevo apellido del estudiante");  String apelldioModificado = lector.nextLine().trim();  estudiante.setApellido(apelldioModificado);  //lectura del apellido modificado  System.out.println("Ingrese el nuevo apellido del estudiante");  int semestreModificado = lector.nextInt();  estudiante.setSemestre(semestreModificado);  //lectura del correo del estudiante  System.out.println("Ingrese el nuevo correo del estudiante");  String correoModificado = lector.nextLine().trim();  estudiante.setCorreo(correoModificado);  //lectura del numero de celular del estudiante  System.out.println("Ingrese el nuevo numero de celular del estudiante");  int numeroModificado = lector.nextInt();  estudiante.setCelular(numeroModificado);  //cerrar la bandera para finalizar la modificacion  cambio = true;  break;  }  if (cambio == false) {  System.out.println("No hay registro del estudiante");  }  }  /\*\*  \* metodo para eliminar a un estudiante del listado  \*/  public static void eliminarEstudiante(ArrayList<Estudiante> listadoEstudiantes, Scanner lector) {  System.out.println("Ingrese el numero de cedula para eliminar al estudiante ");  int eliminarEstudiante = lector.nextInt();  Estudiante eliminacionEstudiante = null;  for (Estudiante estudiante : listadoEstudiantes) {  if (estudiante.getCedula() == eliminarEstudiante) {  estudiante = eliminacionEstudiante;  }  }  if (eliminacionEstudiante == null) {  listadoEstudiantes.remove(eliminacionEstudiante);  }  }  /\*\*  \* Metodo para consultar el listado de los estudiantes  \*/  public static void mostrarEstudiantes(ArrayList<Estudiante> listadoEstudiantes) {  for (Estudiante estudiantesTotales : listadoEstudiantes) {  System.out.println("Los estudiantes registrados son: /n " + estudiantesTotales);  }  }  }  /\*\*  \*  \* @author Juan Goyes  \* clase Estudiante  \* En esta clase estan los atributos y constructores con metodos getters y setters  \*/  public class Estudiante {  private int cedula;  private String nombre;  private String apellido;  private int semestre;  private String correo;  private int celular;  public Estudiante() {  }  public Estudiante(int cedula, String nombre, String apellido, int semestre, String correo, int celular) {  this.cedula = cedula;  this.nombre = nombre;  this.apellido = apellido;  this.semestre = semestre;  this.correo = correo;  this.celular = celular;  }  public int getCedula() {  return cedula;  }  public String getNombre() {  return nombre;  }  public String getApellido() {  return apellido;  }  public int getSemestre() {  return semestre;  }  public String getCorreo() {  return correo;  }  public int getCelular() {  return celular;  }  public void setCedula(int cedula) {  this.cedula = cedula;  }  public void setNombre(String nombre) {  this.nombre = nombre;  }  public void setApellido(String apellido) {  this.apellido = apellido;  }  public void setSemestre(int semestre) {  this.semestre = semestre;  }  public void setCorreo(String correo) {  this.correo = correo;  }  public void setCelular(int celular) {  this.celular = celular;  }          } | | |