



UNIVERSIDAD EUROPEA



ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

CICLO FORMATIVO DE GRADO SUPERIOR 1º DESARROLLO DE
APLICACIONES WEB

PROYECTO INTEGRADOR

Aaron Escudero | Daniel Herrero | Alejandro Junyent | Javier Chicano

CURSO 2022-23



1. Objetivo.....	2
2. Introducción.....	2
2.1. Descripción del proyecto.....	2
2.2. Retos del proyecto.....	2
3. Recursos.....	3
4. Procedimiento.....	4
4.1. Diagramas.....	4
4.1.1. Diagrama de casos de uso.....	4
4.1.2. Diagrama de clases.....	6
4.2. Diseño de interfaces.....	6
4.3. Creación e implementación de una base de datos.....	7
4.4. Implementación del proyecto en Java.....	9
5. Anexos.....	12
5.1. Diseño de interfaces.....	12
5.2. Script Sql.....	18
5.3. Java.....	21
5.3.1. Paquete Modelo.....	21
5.3.5. Paquete Vista.....	26
5.3.6. Paquete Controlador.....	97
5.4. Paquete Main.....	98
5.5. Paquete Acceso a la base de datos.....	100

1. Objetivo

El objetivo de este proyecto es desarrollar una aplicación de gestión para los proyectos integradores creados por los alumnos de los ciclos de DAM, ASIR, DAW y Animaciones 3D y Videojuegos. La aplicación proporcionará los mecanismos necesarios para llevar a cabo un CRUD básico, permitiendo dar de alta, baja, modificar y consultar tanto los proyectos integradores como los alumnos y las áreas correspondientes. Esto facilitará la administración y seguimiento de los proyectos, brindando una herramienta eficiente para gestionar la información relacionada con ellos.

2. Introducción

2.1. Descripción del proyecto

La aplicación de gestión para proyectos integradores es un programa diseñado para administrar proyectos integradores. Esta herramienta permite funcionalidades para dar de alta, baja, modificar y consultar información relacionada con los proyectos, los alumnos y las áreas involucradas.

Con esta aplicación, los administradores podrán crear registros de proyectos integradores, alumnos y áreas, proporcionando los datos pertinentes en cada caso. Además, tendrán la capacidad de realizar modificaciones en dichos registros en caso de cambios. La aplicación también permite la eliminación de proyectos, alumnos o áreas en caso de que ya no sean necesarios.

Además, la aplicación proporciona una interfaz de consulta y listado, lo que permite a los usuarios visualizar y buscar información específica sobre proyectos integradores, alumnos y áreas. Esto facilita la revisión y el análisis de la información almacenada en la aplicación..

2.2. Retos del proyecto

Realizar el proyecto de la aplicación de gestión para proyectos integradores puede presentar diversos retos. Algunos posibles desafíos incluyen:

1.-Diseño de la interfaz de usuario: Es importante que la interfaz de usuario de la aplicación sea fácil de usar y comprendida por los usuarios. El diseño y la disposición de los elementos deben ser claros y eficientes para facilitar la navegación y el uso de las funcionalidades por parte de los usuarios.

2.-Integración de tecnologías: Es necesario integrar una base de datos alojada en un servidor externo, para que cualquier usuario, simplemente con el programa en local, pueda acceder a la información guardada

3.-Validación y seguridad de los datos: Dado que la aplicación almacenará información sensible, es fundamental implementar mecanismos de validación y seguridad adecuados

para garantizar la integridad y confidencialidad de los datos. Por ello, nuestro servidor de sql está alojado en un servidor seguro donde exclusivamente está abierto el puerto 3306 de escucha con el protocolo tcp

4.-Pruebas exhaustivas: Es esencial realizar pruebas exhaustivas en la aplicación para identificar y corregir posibles errores y garantizar su correcto funcionamiento. Las pruebas las realizaremos con un sistema unitario de pruebas llamado JUnit.

5.-Mantenimiento y evolución: Una vez finalizada la implementación inicial, es probable que se requieran actualizaciones, mejoras y correcciones a lo largo del tiempo. Mantener y evolucionar la aplicación para satisfacer las necesidades cambiantes puede suponer un reto en términos de gestión de versiones, control de cambios y actualización de funcionalidades.

Superar estos retos requerirá una planificación adecuada, una buena organización del equipo de desarrollo, así como el uso de buenas prácticas de desarrollo de código y un enfoque iterativo para abordar y resolver los problemas que puedan surgir durante el desarrollo del proyecto.

3. Recursos

Para la correcta funcionalidad del proyecto, necesitaremos utilizar herramientas adecuadas para tener éxito en la creación.

Para la creación de los diagramas de clases, y los diagramas de casos de uso, utilizaremos la herramienta Software Ideas Modeler. Con la versión gratuita de esta herramienta seremos capaces de satisfacer los requisitos obligatorios.

Para el diseño de las interfaces gráficas, utilizaremos el recurso Online Canva ya que tiene una gran biblioteca de opciones para personalizar al máximo nuestro diseño. Con Canva también generaremos diferentes recursos gráficos como botones, fondos y el logo.

También, disponemos de un Ubuntu Server 22.04 que aloja MYSQL Server 8.0 para nuestra base de datos. Con esto lograremos que cualquier persona que tenga el código del programa, pueda acceder a la base de datos sin necesidad de tenerla en local. Para administrar la base de datos podemos usar el MySQL WorkBench o el DBeaver.

Nuestro proyecto requerirá también de un control de versiones. Este estará alojado en un repositorio de GitHub donde llevaremos un seguimiento de los cambios del código y poder colaborar todos los integrantes del grupo.

Es importante el proceso de documentación del proyecto. Utilizaremos la herramienta de generación de documentación Javadoc para generar documentación técnica a partir del código fuente.

Cómo entorno de programación usaremos Eclipse (con el plugin Window Builder). Esta herramienta será la que más usemos y la que nos permitirá conectar la base de datos con nuestro programa.

Por último, tendremos un gestor de dependencias que nos ayudará a administrar las dependencias de tu proyecto y facilitará la descarga y actualización de bibliotecas externas necesarias. Para este proyecto usaremos Maven

4. Procedimiento

4.1. Diagramas

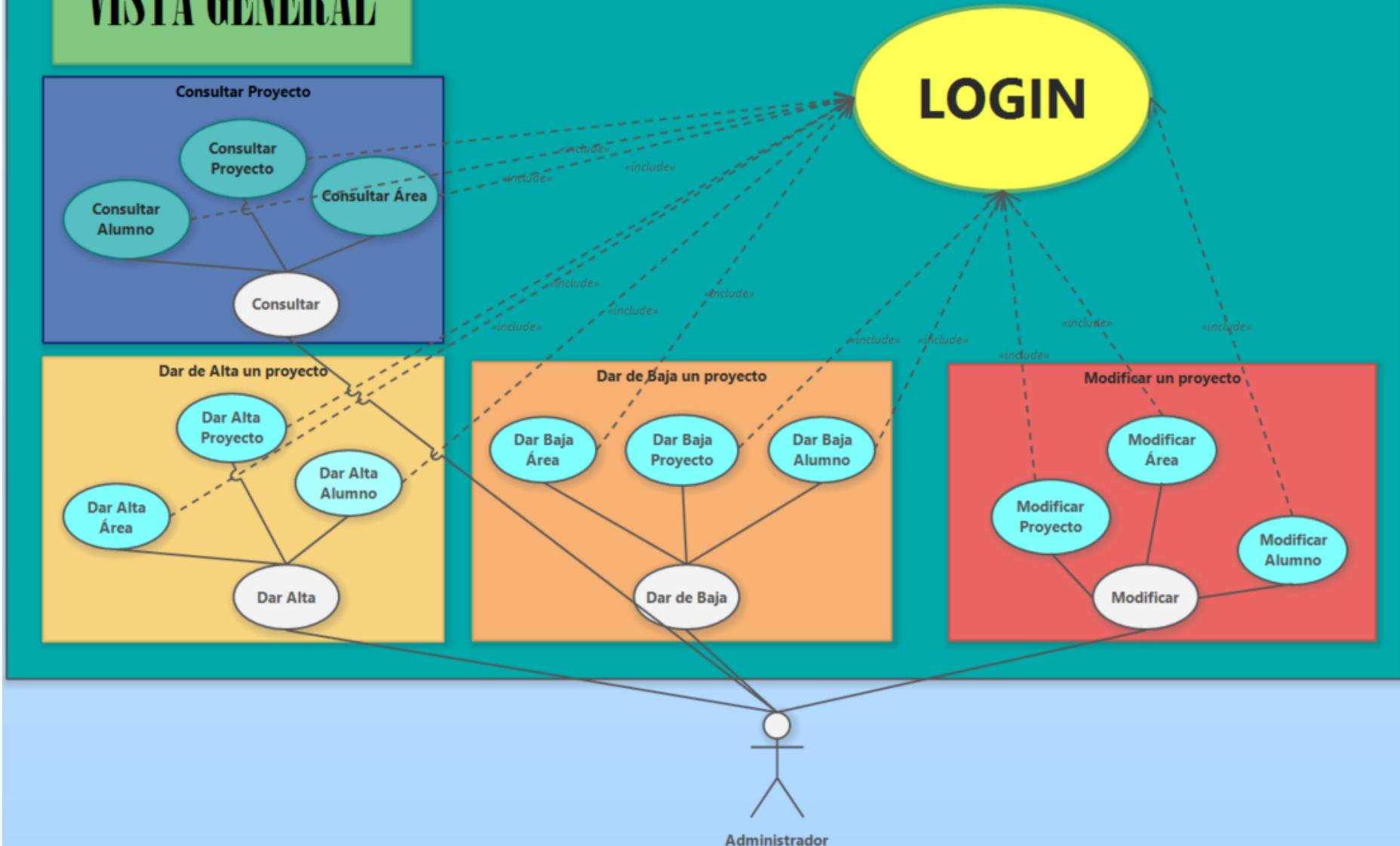
4.1.1. Diagrama de casos de uso.

Para realizar el diagrama de casos de uso, hemos aplicado unos pasos lógicos. Estos son:

1. Identificar los actores principales. Tendremos que identificar de una forma correcta a los usuarios externos que interactuarán con el sistema.
2. Identificar los casos de uso. Debemos de identificar las principales funcionalidades que el sistema debe proporcionar a los actores. Los casos de uso representan las interacciones entre los actores y el sistema.
3. Nombrar esos casos de uso de una forma coherente.
4. Determinar las relaciones de include y extend. El include lo encontramos cuando un caso de uso depende de otro caso de uso para su ejecución, mientras que el extend lo usamos cuando un caso de uso es opcional y agrega funcionalidades adicionales a otro caso de uso.
5. Diseñar el diagrama UML que represente visualmente a los actores, casos de uso y relaciones que existen entre ellos.

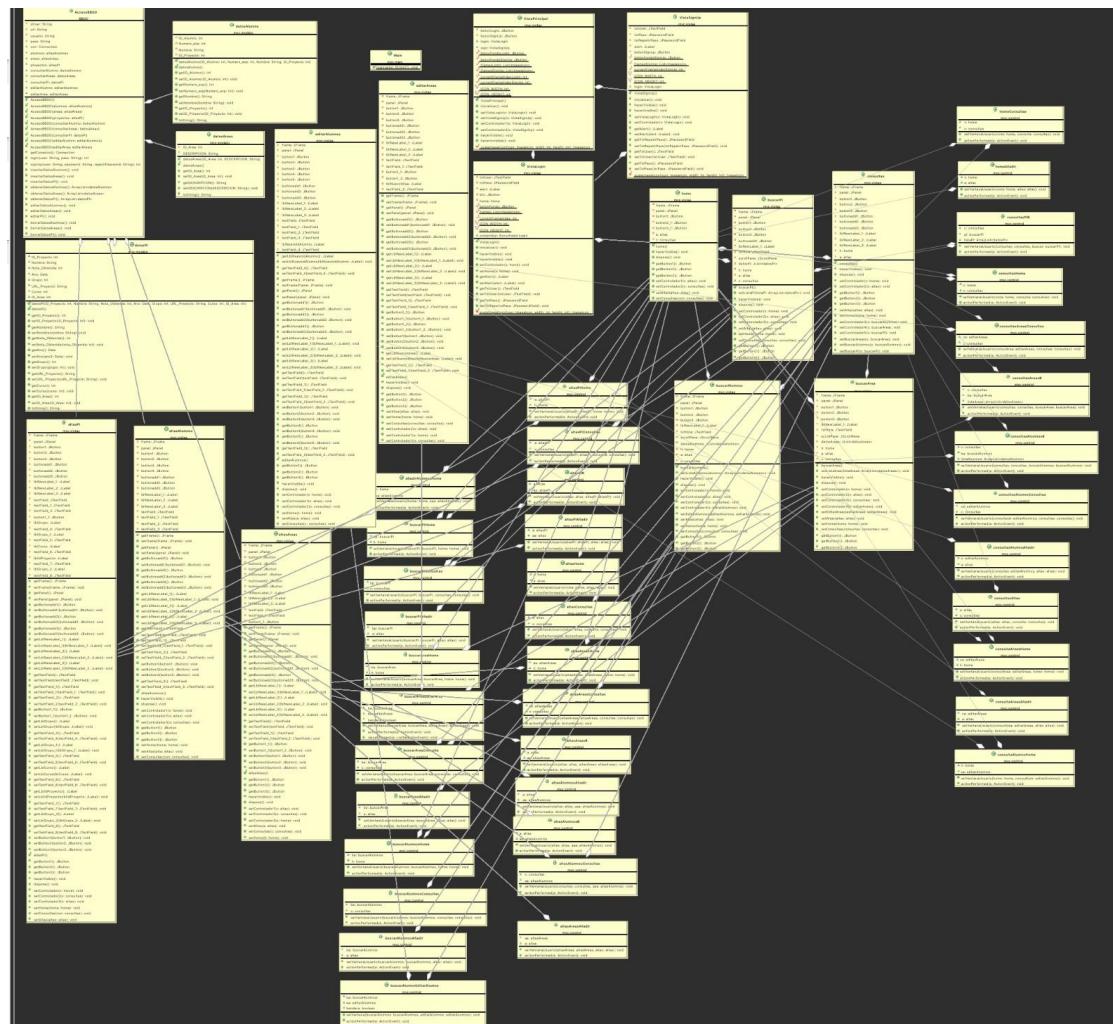
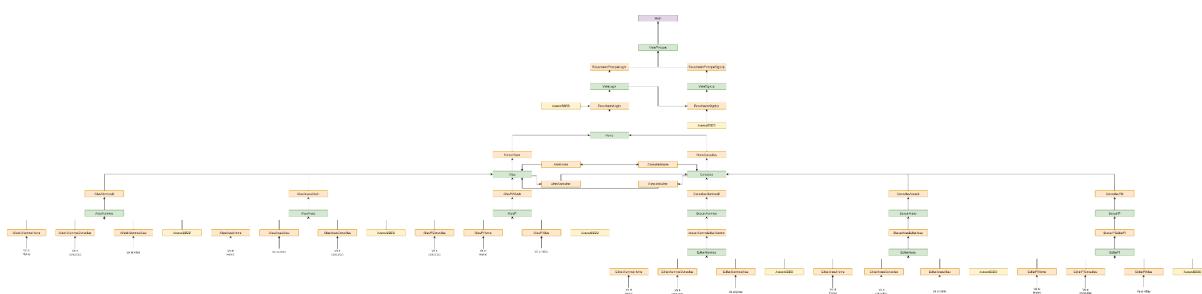
En nuestro caso, el caso de uso "Login" está incluido en todos los casos de uso. Esto significa que el inicio de sesión es una parte necesaria para acceder a las funcionalidades de gestión de proyectos integradores, alumnos y áreas.

VISTA GENERAL



4.1.2. Diagrama de clases.

Es una parte fundamental del lenguaje de modelado unificado (UML) y se utiliza para mostrar las clases, sus atributos, métodos y las relaciones entre ellas. Primero tenemos que identificar las clases principales. Luego identificaremos los atributos y métodos para cada clase. Tendremos que establecer las relaciones entre clases e identificar si son relaciones uno a uno, uno a muchos y muchos a muchos, así como la composiciones y agregaciones. Además, deberemos analizar si existe algún tipo de herencia entre las clases. El diagrama tiene que mantener la simplicidad y claridad, se trata de capturar los aspectos fundamentales de la estructura y las relaciones de clases para una comprensión efectiva.



[Link al diagrama en mejor resolución](#)

4.2. Diseño de interfaces

Para realizar el diseño gráfico de las interfaces es necesario seguir estos criterios generales:

- ★ **Jerarquía clara:** Establecer una estructura visual que destaque los elementos principales y transmita un orden de ejecución.
- ★ **Foco en elementos clave:** Resaltar las zonas relevantes y agrupar los elementos principales para facilitar la comprensión y navegación del usuario.
- ★ **Homogeneidad:** Mantener consistencia en la estructura de la interfaz en todas las ventanas para una experiencia coherente.
- ★ **Relaciones visuales:** Colocar cerca los elementos relacionados funcionalmente y evitar interfaces sobrecargadas.
- ★ **Estructura visual y flujo de trabajo:** Mantener una estructura clara que permita una navegación fluida según el flujo de trabajo de la aplicación.
- ★ **Alineación de elementos:** Alinear visualmente los elementos para mejorar la experiencia de usuario y usar una rejilla como guía.
- ★ **Acceso rápido:** Utilizar atajos de teclado para acceder a opciones frecuentemente utilizadas.
- ★ **Navegación intuitiva:** Usar iconos y convenciones de navegación estándar para facilitar la ubicación y acceso a las funciones.
- ★ **Visualización de información:** Presentar información relevante de forma clara y concisa, utilizando gráficos y esquemas.
- ★ **Entrada de datos simplificada:** Minimizar las acciones de entrada de datos y proporcionar ayuda y validación de datos cuando sea necesario.
- ★ **Interacción efectiva:** Orientar al usuario, solicitar confirmación en acciones importantes y brindar una comunicación clara.
- ★ **Uso efectivo del color:** Utilizar el color para representar información compleja de manera consistente.
- ★ **Diseño de ventanas y menús:** Dividir el espacio en áreas funcionales y utilizar una barra de menús organizada.

- ★ **Botones claros:** Utilizar botones para acciones importantes con contenido legible y accesibilidad mediante teclas rápidas.
- ★ **Sistemas de ayuda:** Proporcionar asistencia contextual y mensajes de error claros.

Estos principios de diseño de UI garantizan una experiencia de usuario mejorada y una mayor usabilidad en las aplicaciones.

[Link al diseño de interfaces](#)

4.3. Creación e implementación de una base de datos

En el proyecto, está contemplada una Base de datos donde alojar todos los datos de los alumnos, áreas y proyectos integradores. En nuestro caso, también implementaremos un inicio de sesión para que sea más segura la aplicación, por lo tanto, en esa base de datos alojaremos en una tabla los usuarios y contraseñas del inicio de sesión. También está contemplada la opción de creación de nuevos usuarios con un formulario de registro.

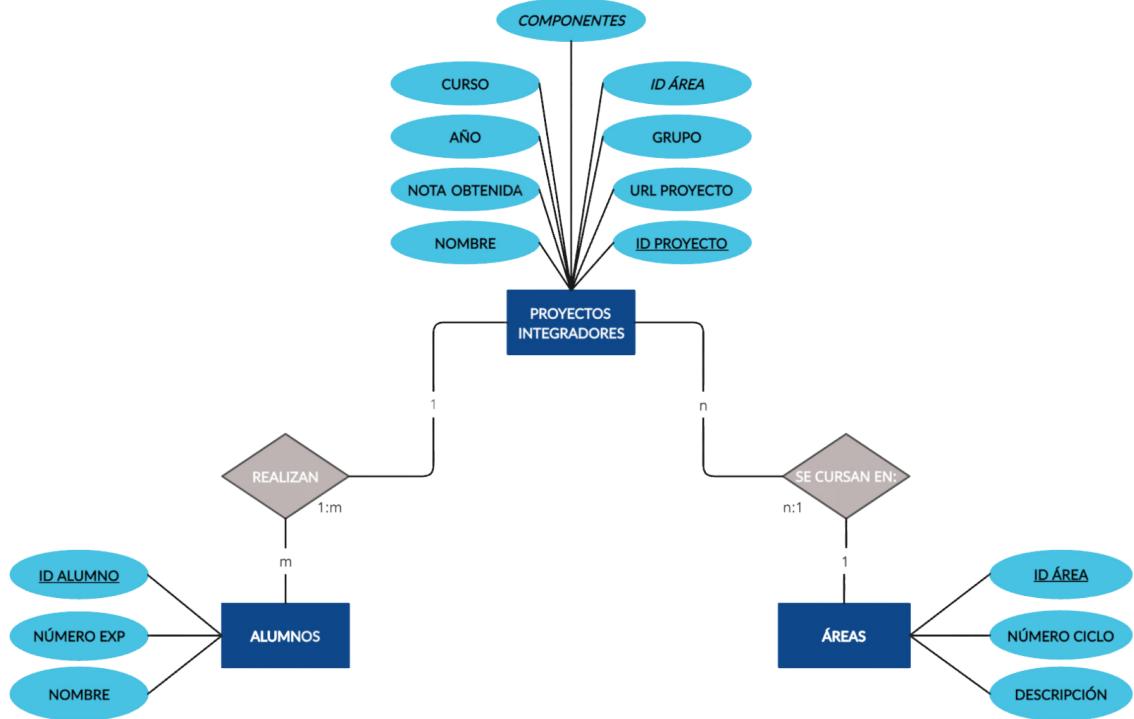
Para que el programa sea más accesible, hemos creado un servidor Ubuntu server 22.04 donde se aloja un servicio de MySQL. Esto nos permite tener a todos los creadores del proyecto la misma base de datos, lo que hace evitar errores. Esta acción está pensada para que el programa se pueda ejecutar desde cualquier dispositivo que soporte Java sin necesidad de trasladar con ella toda la base de datos.

```
glabs@glabs:~$ sudo systemctl status mysql
[sudo] password for glabs:
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:disabled)
   Active: active (running) since Mon 2023-05-22 16:39:58 UTC; 52s ago
     Process: 785 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 864 (mysqld)
      Status: "Server is operational"
        Tasks: 38 (limit: 3205)
       Memory: 424.5M
          CPU: 3.130s
         CGroup: /system.slice/mysql.service
                   └─864 /usr/sbin/mysqld

may 22 16:39:42 glabs systemd[1]: Starting MySQL Community Server...
may 22 16:39:58 glabs systemd[1]: Started MySQL Community Server.
```

Lo primero de todo, es realizar un diagrama Entidad Relación. Primero, localizamos las tres grandes entidades principales: Área, Alumno y Proyecto Integrador. La entidad Área tiene los atributos ID_Area(clave primaria) y Descripción. La entidad Alumno tiene los atributos ID_Alumno (clave primaria), Número de Expediente, Nombre y ID_Proyecto (relación con la entidad de proyecto). Y La entidad Proyecto Integrador tiene los atributos ID_Proyecto(clave primaria), Nombre, URL, Nota, Año, Curso, Grupo y Área (relación con la entidad Área).

El diagrama de forma gráfica sería así:



Del modelo Entidad relación anterior, sacamos el modelo relacional normalizado:

La tabla “**Proyectos integradores**” está en 0FN, porque la columna componentes puede tener varios valores en la misma tupla.

ID Proyecto	Nombre	Nota Obtenida	Año	Grupo	URL Proyecto	Curso	ID Área	Componentes
001	Proyecto A	8.5	2022	Grupo 1	www.proyectoa.com	Ingeniería de Software	101	002, 003, 005
002	Proyecto B	9.2	2023	Grupo 2	www.proyectob.com	Bases de Datos	103	001, 008, 007
003	Proyecto C	7.8	2021	Grupo 3	www.proyectoc.com	Redes de Computadoras	102	004, 006

La tabla “**Alumnos**” está en 1FN, porque no tiene atributos multivaluados

ID_Alumno	Número Exp	Nombre
001	12345	Juan Pérez
002	23456	Ana García
003	34567	Luis Hernandez
004	45678	Sofia Diaz

005	56789	Carlos Martinez
006	67890	Maria Rodriguez
007	78901	Pablo Gomez
008	89012	Andrea Fernandez

La tabla “Áreas” está en 1FN, porque no tiene atributos multivaluados

<u>Id Área</u>	Número Ciclo	Descripción
101	1	Ingeniería de Software
102	2	Sistemas Operativos
103	3	Programación Web

Normalizamos la tabla “Proyectos integradores”

Para pasarlo a 1FN, eliminamos los atributos multivaluados creando nuevas tablas

PROYECTO (ID Proyecto, Nombre, Nota Obtenida, Año, Grupo, Url Proyecto, Curso, *ID Área)
PK ID Proyecto
FK ID Área → ÁREAS

La tablas “Áreas y Alumnos” como ya están en 1FN, quedarían así;

ÁREAS (ID Área, número ciclo, descripción)

PK ID Área

ALUMNOS (ID Alumnos, número exp, nombre, *ID Proyecto*)

PK ID Alumnos

FK ID Proyecto → Proyecto

Para pasar las tablas a 2FN, es necesario separar los atributos que dependen de una parte de la clave en distintas tablas.

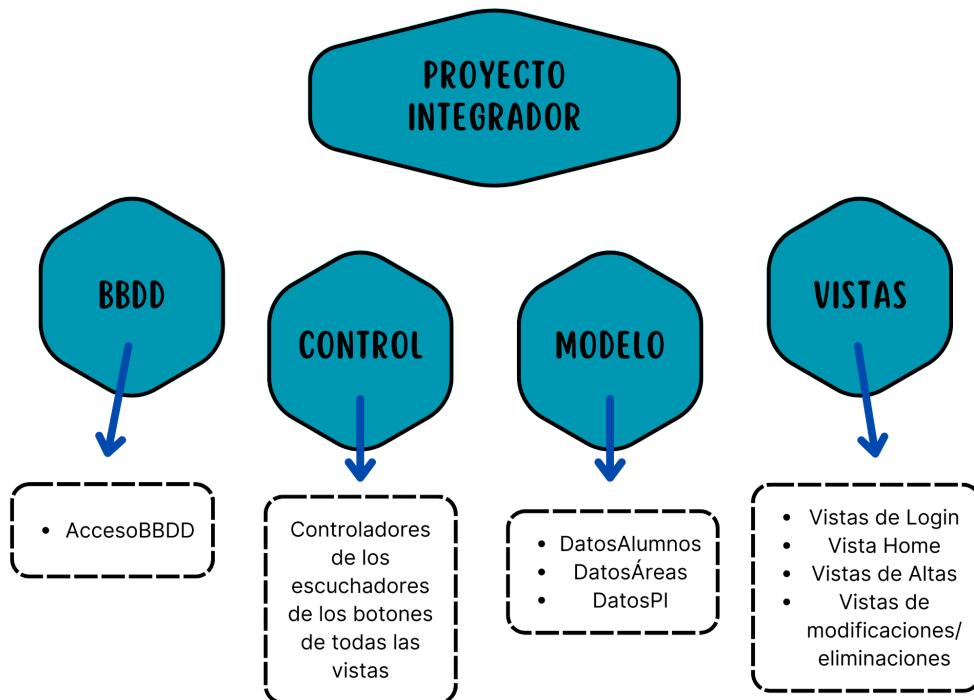
En este caso todas las tablas ya están en 2FN.

Ahora pasamos a 3FN. Para esto, los atributos que tienen dependencia de la PK de su tabla de forma transitiva se separan y pasan a una tabla nueva.

4.4. Implementación del proyecto en Java

Tras analizar todos los diagramas y el diseño de interfaces, podemos empezar a generar código para hacer realidad el proyecto. Primero, tendremos que crear un modelo en maven que nos proporciona el controlador JDBC (estándar de conectividad de bases de datos de Java). Al tener el proyecto en maven, el controlador se actualiza automáticamente. El proyecto está alojado en un repositorio en github.

El proyecto sigue el patrón MVC (Modelo Vista Controlador) dividido en paquetes. La estructura de paquetes, sigue el siguiente esquema:

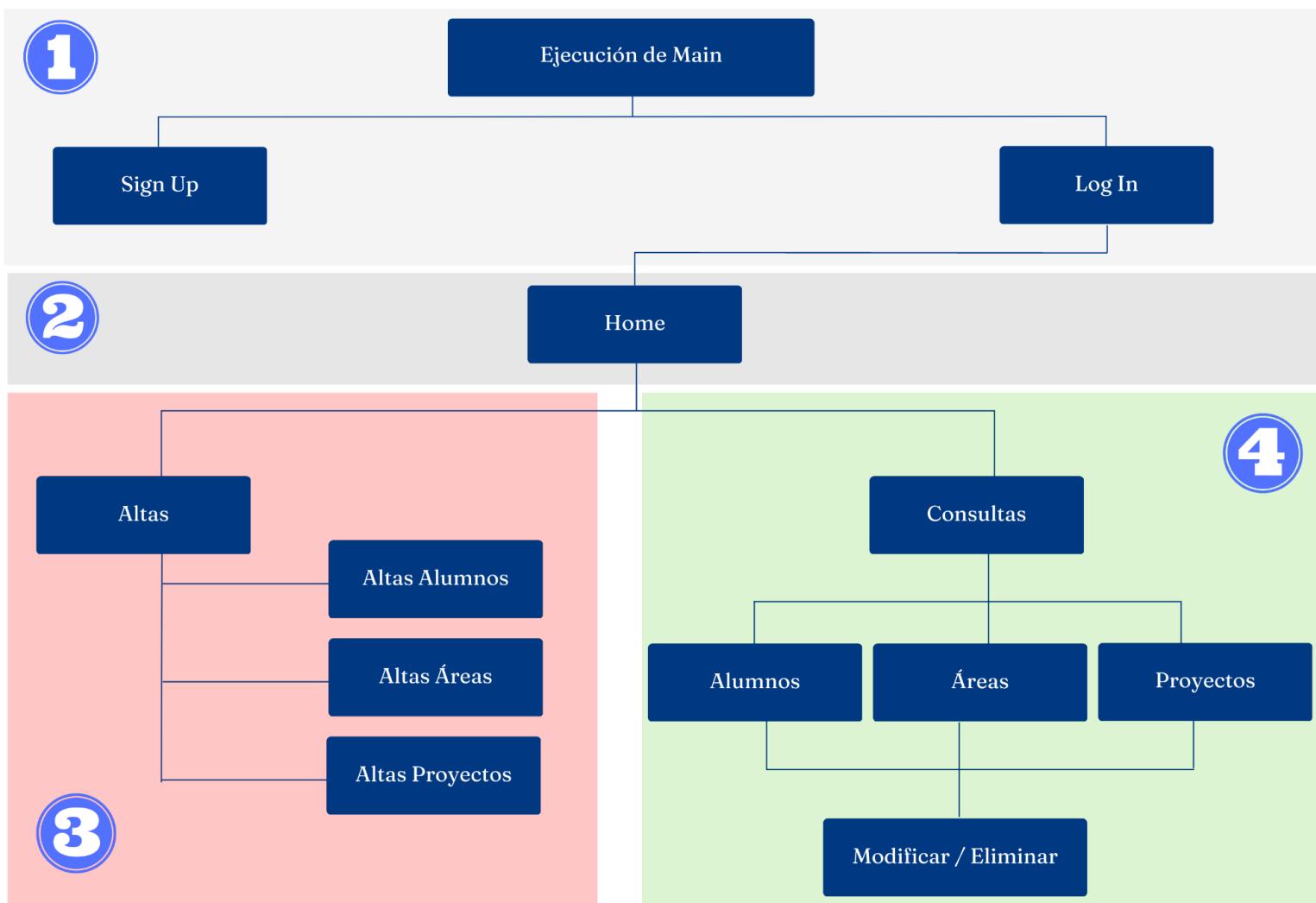


En el acceso a la base de datos, localizamos la conexión a la base de datos y los diferentes comandos de sql (querys) para hacer las altas, modificaciones, consultas y supresiones de datos.

En el paquete de control, localizamos toda la lógica de los botones de todo el programa. En total tenemos 40 botones. La mayoría de estos simplemente realizan el cambio de una vista a otra, sin embargo, hay otros que mandan la orden de ejecutar una query en la base de datos.

Dentro del paquete de modelo encontramos los datos que se transfieren del programa a la base de datos y viceversa.

Por último, en el paquete de vistas ubicamos todas las ventanas de la interfaz gráfica del programa. Si desglosamos el paquete, mantendría esta forma:



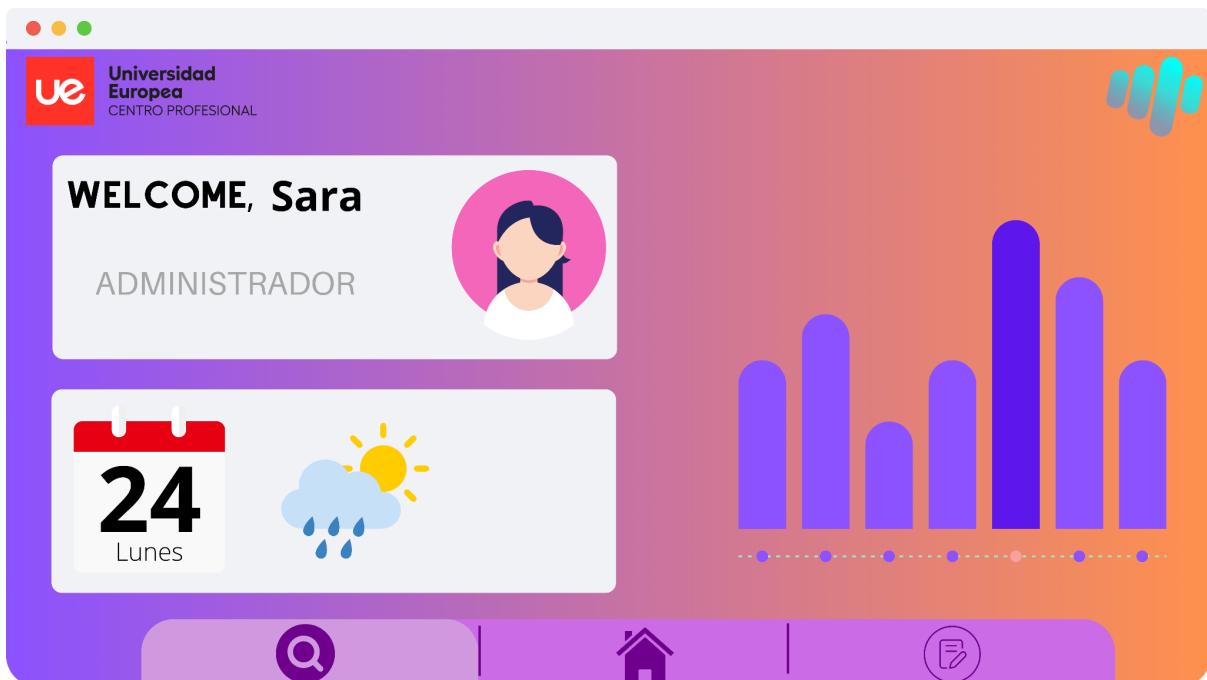
[Link para consultar todo el código](#)

Cómo añadido al proyecto y siguiendo la línea del mismo, hemos creado una página web con certificado ssl para que los usuarios puedan acceder de forma segura:

www.glabs.gay

5. Anexos

5.1. Diseño de interfaces





ALUMNO

Nombre: [REDACTED]

ID Alumno: [REDACTED]

Nº Expediente: [REDACTED]



ÁREAS

ID Área: [REDACTED]

Nº ciclo: [REDACTED]

Descripción: [REDACTED]



The screenshot shows a search interface for 'PROYECTOS INTEGRADORES'. It includes fields for 'Nombre' (Name), 'Componentes' (Components), 'ID Área' (Area ID), 'Grupo' (Group), 'Año' (Year) with a dropdown menu, 'Curso' (Course), 'ID Proyecto' (Project ID), 'Nota Obtenida' (Grade Obtained), and 'URL'. Below the search bar are three icons: a magnifying glass, a house, and a circular edit/pencil icon.

gLabs

ALTAS

Universidad Europea
CENTRO PROFESIONAL

PROYECTOS INTEGRADORES

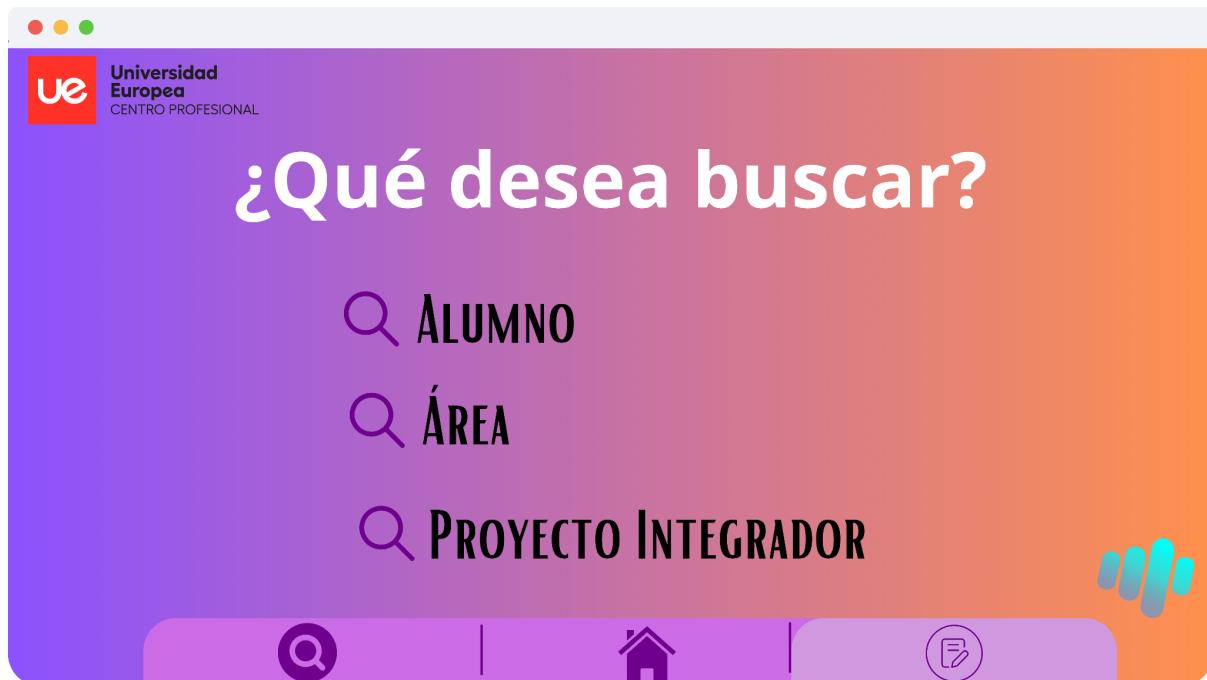
Nombre:

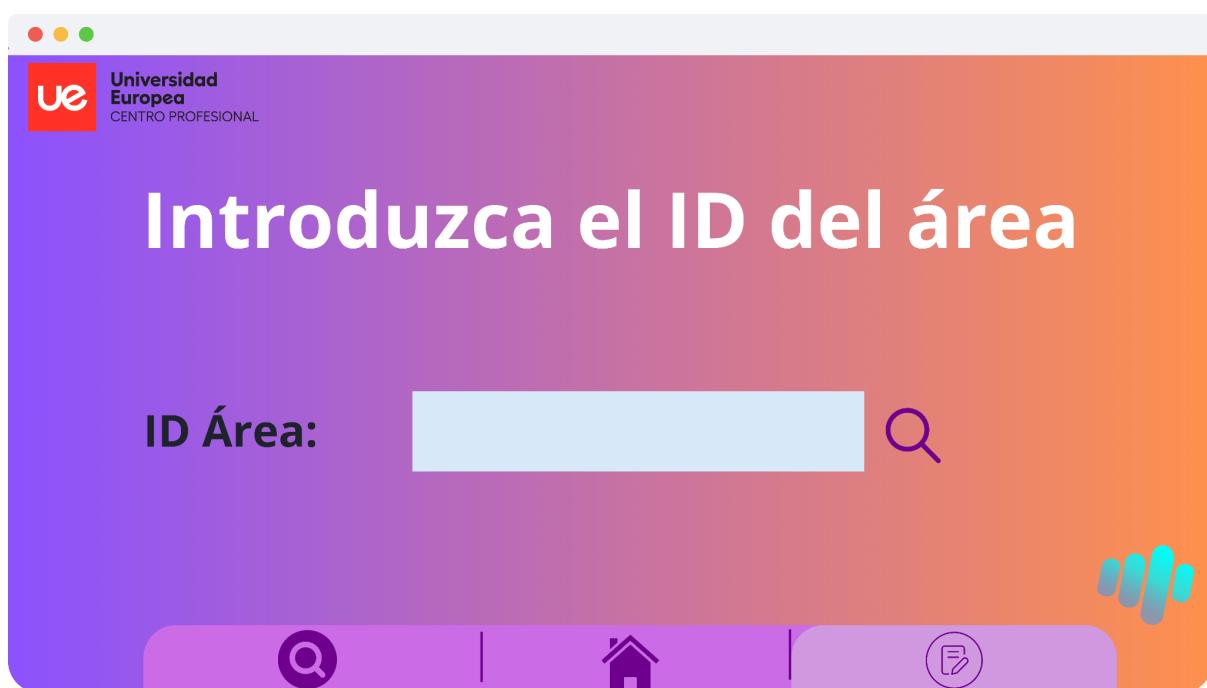
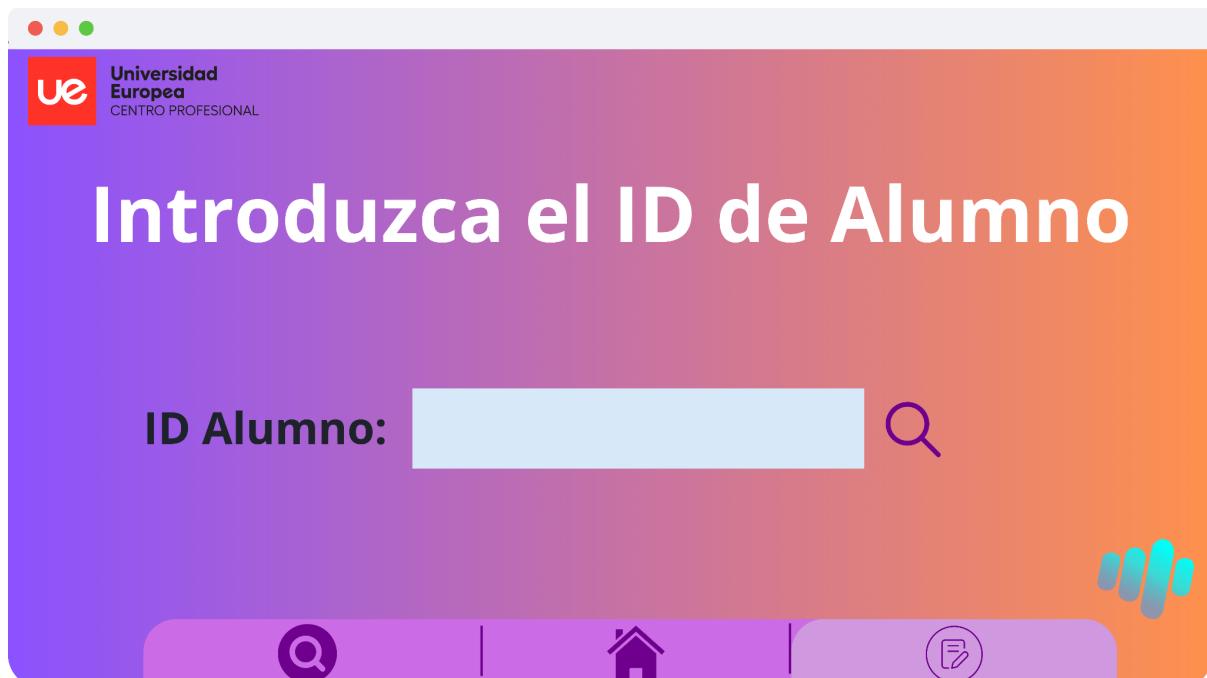
Componentes:

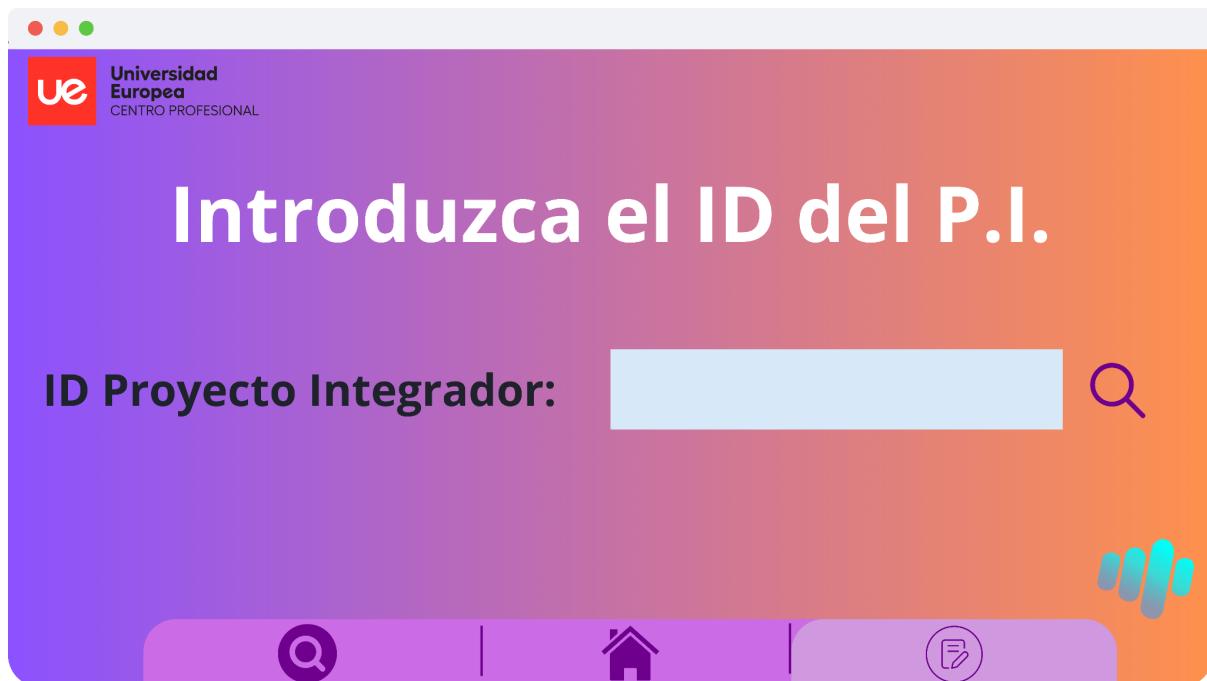
ID Área: Grupo: Año:

Curso: ID Proyecto: Nota Obtenida:

URL:







The screenshot shows a web application interface. At the top, there's a header with the word 'CONSULTA' in large white letters on a blue background. To the right of the header is the logo for 'Universidad Europea CENTRO PROFESIONAL'. Below the header, there's a dark blue navigation bar with the word 'ÁREAS' in white. The main content area has a light blue background. On the left, there's a form field labeled 'ID Área:' with the value '76549'. Next to it is another form field labeled 'Nº ciclo:' with the value '65'. Below these fields is a section labeled 'Descripción:' containing the text 'Administración de Sistemas Operativos'. To the right of this text are two icons: a blue pencil icon and a red circular icon with a white 'X' inside.



This screenshot shows a consultation page for 'CONSULTA' under the 'gLabs' brand. The header includes the 'gLabs' logo, the word 'CONSULTA', and the 'Universidad Europea CENTRO PROFESIONAL' logo. Below the header, there's a dark blue navigation bar with the text 'PROYECTOS INTEGRADORES' in white. The main content area contains several form fields: 'Nombre:' with the value 'Proyecto Integrador de Administración de Sistemas Informáticos en Red', 'Componentes:' with the value 'Aaron, Alejandro, Daniel y Javier', 'ID Área:' with the value 'STEAM', 'Grupo:' with the value '7', 'Año:' with the values '2', '0', '2', and '3' separated by dots, 'Curso:' with the value '1', 'ID Proyecto:' with the value '543', 'Nota Obtenida:' with the value '95.43', and 'URL:' with the value '<https://github.com/Alexjun09/proyecto-integrador>'. To the right of these fields are two large icons: a blue pencil inside a circle and a red circular icon with a white 'X' inside.



5.2. Script Sql

```
-- MySQL dump 10.13 Distrib 8.0.33, for Win64 (x86_64)
--
-- Host: 127.0.0.1 Database: loginpi
-----
-- Server version     8.0.33

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET      @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `alumnos`
--

DROP TABLE IF EXISTS `alumnos`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `alumnos` (
  `ID_Alumno` int NOT NULL,
  `Numero_exp` int DEFAULT NULL,
  `Nombre` varchar(255) DEFAULT NULL,
  `ID_Proyecto` int DEFAULT NULL,
  PRIMARY KEY (`ID_Alumno`),
  UNIQUE KEY `Numero_exp_UNIQUE` (`Numero_exp`),
  KEY `ID_Proyecto` (`ID_Proyecto`),
  CONSTRAINT `alumnos_ibfk_1` FOREIGN KEY (`ID_Proyecto`) REFERENCES `proyecto`(`ID_Proyecto`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `alumnos`
--

LOCK TABLES `alumnos` WRITE;
/*!40000 ALTER TABLE `alumnos` DISABLE KEYS */;
```

```

INSERT INTO `alumnos` VALUES (1,22030041,'Aaron Escudero',2);
/*!40000 ALTER TABLE `alumnos` ENABLE KEYS */;
UNLOCK TABLES;

-- 
-- Table structure for table `areas` 

-- 

DROP TABLE IF EXISTS `areas`;
/*!40101 SET @saved_cs_client    = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `areas` (
  `ID_Area` int NOT NULL,
  `DESCRIPCION` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`ID_Area`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `areas` 

-- 

LOCK TABLES `areas` WRITE;
/*!40000 ALTER TABLE `areas` DISABLE KEYS */;
INSERT INTO `areas` VALUES (1,'Entornos de Desarrollo');
/*!40000 ALTER TABLE `areas` ENABLE KEYS */;
UNLOCK TABLES;

-- 
-- Table structure for table `proyecto` 

-- 

DROP TABLE IF EXISTS `proyecto`;
/*!40101 SET @saved_cs_client    = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `proyecto` (
  `ID_Proyecto` int NOT NULL,
  `Nombre` varchar(255) DEFAULT NULL,
  `Nota_Obtendida` int DEFAULT NULL,
  `Ano` date DEFAULT NULL,
  `Grupo` char(1) DEFAULT NULL,
  `URL_Proyecto` varchar(255) DEFAULT NULL,
  `Curso` int DEFAULT NULL,
  `ID_Area` int DEFAULT NULL,
  PRIMARY KEY (`ID_Proyecto`),
  KEY `ID_Area` (`ID_Area`),
  CONSTRAINT `proyecto_ibfk_1` FOREIGN KEY (`ID_Area`) REFERENCES `areas` (`ID_Area`)
)

```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `proyecto`
-- 

LOCK TABLES `proyecto` WRITE;
/*!40000 ALTER TABLE `proyecto` DISABLE KEYS */;
INSERT INTO `proyecto` VALUES (1,'Proyecto Integrador gLabs',10,'2023-05-17','1','gLabs.com',1,1),(2,'Proyecto Integrador lolo',10,'2001-07-20','2','lolo.com',2,1);
/*!40000 ALTER TABLE `proyecto` ENABLE KEYS */;
UNLOCK TABLES;

-- 
-- Table structure for table `usuarios`
-- 

DROP TABLE IF EXISTS `usuarios`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `usuarios` (
  `usuario` varchar(15) NOT NULL,
  `contrasena` varchar(45) DEFAULT NULL,
  `admin` int DEFAULT NULL,
  PRIMARY KEY (`usuario`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `usuarios`
-- 

LOCK TABLES `usuarios` WRITE;
/*!40000 ALTER TABLE `usuarios` DISABLE KEYS */;
INSERT INTO `usuarios` VALUES ('admin','admin',1);
/*!40000 ALTER TABLE `usuarios` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

-- Dump completed on 2023-05-22 12:04:27

5.3. Java

5.3.1. Paquete Modelo

5.3.2. DatosAlumno

```
package mvc.modelo;

/**
 * Clase que almacena todos los datos de los alumnos
 *
 * @author alexjun09
 *
 */
public class datosAlumno {
    private int ID_Alumno;
    private int Numero_exp;
    private String Nombre;
    private int ID_Proyecto;

    /**
     * Constructor del objeto datosAlumno
     *
     * @param ID_Alumno el Identificador del alumno
     * @param Numero_exp el numero de expediente del alumno
     * @param Nombre el nombre del alumno
     * @param ID_Proyecto el id del proyecto integrador que ha hecho
     */
    public datosAlumno(int ID_Alumno, int Numero_exp, String Nombre, int ID_Proyecto) {
        this.ID_Alumno = ID_Alumno;
        this.Numero_exp = Numero_exp;
        this.Nombre = Nombre;
        this.ID_Proyecto = ID_Proyecto;
    }

    /**
     * constructor vacio del objeto datos alumno
     */
    public datosAlumno() {
    }

    /**
     * getter del id del alumno
     *
     * @return ID_Alumno devuelve el id del alumno
     */
    public int getID_Alumno() {
        return ID_Alumno;
    }

    /**
     * setter del id del alumno
     *
     * @param ID_Alumno el id del alumno
     */
    public void setID_Alumno(int ID_Alumno) {
        this.ID_Alumno = ID_Alumno;
    }

    /**
     * getter del numero de expediente del alumno
     *
     * @return Numero_exp el numero de expediente
     */
}
```

```

public int getNumero_exp() {
    return Numero_exp;
}

/**
 * setter del numero de expediente del alumno
 *
 * @param Numero_exp el numero de expediente
 */
public void setNumero_exp(int Numero_exp) {
    this.Numero_exp = Numero_exp;
}

/**
 * getter del nombre del alumno
 *
 * @return Nombre el nombre del alumno
 */
public String getNombre() {
    return Nombre;
}

/**
 * setter del nombre del alumno
 *
 * @param Nombre el nombre del alumno
 */
public void setNombre(String Nombre) {
    this.Nombre = Nombre;
}

/**
 * getter del id del proyecto que ha hecho el alumno
 *
 * @return ID_Proyecto el id del proyecto
 */
public int getID_Proyecto() {
    return ID_Proyecto;
}

/**
 * setter del id del proyecto que ha hecho el alumno
 *
 * @param ID_Proyecto el id del proyecto
 */
public void setID_Proyecto(int ID_Proyecto) {
    this.ID_Proyecto = ID_Proyecto;
}

/**
 * metodo tostring con todos los datos del alumno
 *
 * @return todos los datos del alumno en forma de string
 */
public String toString() {
    return "ID ALUMNO: " + ID_Alumno + " | NºEXPEDIENTE: " + Numero_exp + " | NOMBRE: " + Nombre + " | ID PROYECTO: " + ID_Proyecto ;
}
}

```

5.3.3. DatosAreas

```

package mvc.modelo;
/***
 * Clase que almacenará todos los datos de las areas
 *
 * @author alexjun09
 */
public class datosAreas {
    private int ID_Area;
    private String DESCRIPCION;

    /**

```

```

        * constructor del objeto datosAreas
        *
        * @param ID_Area      el id del area
        * @param DESCRIPCION la description del area
        */
    public datosAreas(int ID_Area, String DESCRIPCION) {
        this.ID_Area = ID_Area;
        this.DESCRIPCION = DESCRIPCION;
    }

    /**
     * constructor del objeto datos area vacio
     */
    public datosAreas() {

    }

    /**
     * getter del id del area
     *
     * @return ID_Area el id del area
     */
    public int getID_Area() {
        return ID_Area;
    }

    /**
     * setter del id del area del proyecto
     *
     * @param iD_Area el id del area
     */
    public void setID_Area(int iD_Area) {
        ID_Area = iD_Area;
    }

    /**
     * getter de la description
     *
     * @return DESCRIPCION la description
     */
    public String getDESCRIPCION() {
        return DESCRIPCION;
    }

    /**
     * setter de la descripcion
     *
     * @param dESCRIPCION la descripcion
     */
    public void setDESCRIPCION(String dESCRIPCION) {
        DESCRIPCION = dESCRIPCION;
    }

    /**
     * metodo tostring que devolverá todos los datos del area
     *
     * @return todos los datos del area en forma de string
     */
    public String toString() {
        return "ID_AREA: " + ID_Area + " | DESCRIPCION: " + DESCRIPCION;
    }
}

5.3.4. DatosPI

package mvc.modelo;
import java.sql.Date;

/**
 * Clase que almacenará todos los datos de los proyectos integradores
 *
 * @author alexjun09
 *
 */

```

```

public class datosPI {
    int ID_Proyecto;
    String Nombre;
    int Nota_Obttenida;
    Date Ano;
    int Grupo;
    String URL_Proyecto;
    int Curso;
    int ID_Area;

    /**
     * Constructor del objeto datosPI
     *
     * @param ID_Proyecto el id del proyecto al que pertenece
     * @param Nombre el nombre del proyecto
     * @param Nota_Obttenida la nota obtenida
     * @param Ano el año del proyecto
     * @param Grupo el grupo al que pertenece el proyecto
     * @param URL_Proyecto el link al proyecto
     * @param Curso el curso del proyecto
     * @param ID_Area el id del area del proyecto
     */
    public datosPI(int ID_Proyecto, String Nombre, int Nota_Obttenida, Date Ano, int Grupo, String URL_Proyecto,
                   int Curso, int ID_Area) {
        this.ID_Proyecto = ID_Proyecto;
        this.Nombre = Nombre;
        this.Nota_Obttenida = Nota_Obttenida;
        this.Ano = Ano;
        this.Grupo = Grupo;
        this.URL_Proyecto = URL_Proyecto;
        this.Curso = Curso;
        this.ID_Area = ID_Area;
    }

    /**
     * constructor del objeto vacío
     */
    public datosPI() {
    }

    /**
     * getter del id del proyecto
     *
     * @return ID_Proyecto el id del proyecto
     */
    public int getID_Proyecto() {
        return ID_Proyecto;
    }

    /**
     * setter del id del proyecto
     *
     * @param iD_Proyecto el id del proyecto
     */
    public void setID_Proyecto(int iD_Proyecto) {
        ID_Proyecto = iD_Proyecto;
    }

    /**
     * getter del nombre del proyecto
     *
     * @return Nombre el nombre del proyecto
     */
    public String getNombre() {
        return Nombre;
    }

    /**
     * setter del nombre del proyecto
     *
     * @param nombre el nombre del proyecto
     */
    public void setNombre(String nombre) {

```

```

        Nombre = nombre;
    }

    /**
     * getter de la nota obtenida por el proyecto
     *
     * @return Nota_Obttenida la nota obtenida
     */
    public int getNota_Obttenida() {
        return Nota_Obttenida;
    }

    /**
     * setter de la nota obtenida en el proyecto
     *
     * @param nota_Obttenida la nota obtenida
     */
    public void setNota_Obttenida(int nota_Obttenida) {
        Nota_Obttenida = nota_Obttenida;
    }

    /**
     * getter del año del proyecto
     *
     * @return Ano el año del proyecto
     */
    public Date getAno() {
        return Ano;
    }

    /**
     * setter del año del proyecto
     *
     * @param ano2 el año del proyecto
     */
    public void setAno(Date ano2) {
        Ano = ano2;
    }

    /**
     * getter del grupo del proyecto
     *
     * @return Grupo el grupo del proyecto
     */
    public int getGrupo() {
        return Grupo;
    }

    /**
     * setter del grupo del proyecto
     *
     * @param grupo el grupo
     */
    public void setGrupo(int grupo) {
        Grupo = grupo;
    }

    /**
     * getter del link del proyecto
     *
     * @return URL_Proyecto el link del proyecto
     */
    public String getURL_Proyecto() {
        return URL_Proyecto;
    }

    /**
     * setter del URL del proyecto
     *
     * @param uRL_Proyecto el url del proyecto
     */
    public void setURL_Proyecto(String uRL_Proyecto) {
        URL_Proyecto = uRL_Proyecto;
    }
}

```

```

/**
 * getter del curso del proyecto
 *
 * @return Curso el curso del proyecto
 */
public int getCurso() {
    return Curso;
}

/**
 * setter del curso
 *
 * @param curso el curso
 */
public void setCurso(int curso) {
    Curso = curso;
}

/**
 * getter del id del area
 *
 * @return ID_Area el id del area
 */
public int getID_Area() {
    return ID_Area;
}

/**
 * setter del id del area del proyecto
 *
 * @param iD_Area el id del area
 */
public void setID_Area(int iD_Area) {
    ID_Area = iD_Area;
}

/**
 * metodo tostring que devolverá todos los datos del area
 *
 * @return todos los datos del area en forma de string
 */
public String toString() {
    return "ID_PROYECTO: " + ID_Proyecto + " | NOMBRE: " + Nombre + " | NOTA: " + Nota_Obtendida + " | AÑO: "
+ Ano + " | GRUPO: " + Grupo + " | URL: " + URL_Proyecto + " | CURSO: " + Curso + " | ID AREA: " + ID_Area ;
}
}

```

5.3.5. Paquete Vista

Altas

```

package mvc.vistas;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;
import mvc.control.altasAlumnosB;
import mvc.control.altasAreasB;
import mvc.control.altasConsultas;
import mvc.control.altasHome;

```

```

import mvc.control.altasPIB;
import mvc.control.homeAñadir;

/**
 * Esta clase es la vista de altas (menu)
 *
 * @author alexjun09
 */
public class altas {
    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;
    private JButton buttonadd1;
    private JButton buttonadd2;
    private JButton buttonadd3;
    private JLabel lblNewLabel_1;
    private JLabel lblNewLabel_2;
    private JLabel lblNewLabel_3;

    private home h;
    private consultas c;

    public altas() {
        // this.h = home;
        // this.c = consulta;

        //////////////// MENÚ Y FONDO///////////////////////
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension screenSize = toolkit.getScreenSize();

        frame = new JFrame("Añadir Alumnos, áreas o proyectos");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(1350, 820);

        panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {

                // Carga y dibuja la imagen de fondo
                ImageIcon imageIcon = new ImageIcon("Imagenes/fondo.png");
                Image image = imageIcon.getImage();
                g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
                setVisible(true);
                super.paintComponent(g);
            }
        };

        // Cambia el icono de la ventana de Java
        ImageIcon icon = new ImageIcon("Imagenes/7.png");
        frame.setIconImage(icon.getImage());

        ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
        Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
        ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
        Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
        ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
        Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

        button1 = new JButton("");
        button1.setBorderPainted(false);
        button1.setBounds(115, 677, 90, 78);
        button1.setIcon(new ImageIcon(resizedImage));
        button1.setContentAreaFilled(false);
        button1.setFocusPainted(false);
        button1.setOpaque(false);

        button2 = new JButton("");
        button2.setBorderPainted(false);
        button2.setBounds(600, 677, 90, 78);
        button2.setIcon(new ImageIcon(resizedAdd));
    }
}

```

```

button2.setContentAreaFilled(false);
button2.setFocusPainted(false);
button2.setOpaque(false);

button3 = new JButton("");
button3.setBorderPainted(false);
button3.setBounds(1085, 677, 90, 78);
button3.setIcon(new ImageIcon(resizedConsultar));
button3.setContentAreaFilled(false);
button3.setFocusPainted(false);
button3.setOpaque(false);

frame.setContentPane(panel);
panel.setLayout(null);

// BOTONES DE BUSCAR
ImageIcon imagenAnadir = new ImageIcon("Imagenes/anadir.png");
Image resizedAnadir = imagenAnadir.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

buttonadd1 = new JButton("");
buttonadd1.setBorderPainted(false);
buttonadd1.setBounds(426, 199, 90, 78);
buttonadd1.setIcon(new ImageIcon(resizedAnadir));
buttonadd1.setContentAreaFilled(false);
buttonadd1.setFocusPainted(false);
buttonadd1.setOpaque(false);

panel.add(buttonadd1);

buttonadd2 = new JButton("");
buttonadd2.setBorderPainted(false);
buttonadd2.setBounds(426, 330, 90, 78);
buttonadd2.setIcon(new ImageIcon(resizedAnadir));
buttonadd2.setContentAreaFilled(false);
buttonadd2.setFocusPainted(false);
buttonadd2.setOpaque(false);
panel.add(buttonadd2);

buttonadd3 = new JButton("");
buttonadd3.setBorderPainted(false);
buttonadd3.setBounds(426, 465, 90, 78);
buttonadd3.setIcon(new ImageIcon(resizedAnadir));
buttonadd3.setContentAreaFilled(false);
buttonadd3.setFocusPainted(false);
buttonadd3.setOpaque(false);
panel.add(buttonadd3);
panel.add(button1);
panel.add(button2);
panel.add(button3);

////////////////////////////// LABELS////////////////////////////

JLabel lblNewLabel = new JLabel("{Qué desea añadir?}");
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setForeground(new Color(247, 254, 255));
lblNewLabel.setFont(new Font("Kefa", Font.PLAIN, 63));
lblNewLabel.setBounds(301, 11, 728, 119);
panel.add(lblNewLabel);

lblNewLabel_1 = new JLabel("Alumnos");
lblNewLabel_1.setFont(new Font("Academy Engraved LET", Font.PLAIN, 38));
lblNewLabel_1.setBounds(528, 188, 179, 89);
panel.add(lblNewLabel_1);

lblNewLabel_2 = new JLabel("Áreas");
lblNewLabel_2.setFont(new Font("Academy Engraved LET", Font.PLAIN, 38));
lblNewLabel_2.setBounds(526, 319, 179, 89);
panel.add(lblNewLabel_2);

lblNewLabel_3 = new JLabel("Proyectos Integradores");
lblNewLabel_3.setFont(new Font("Academy Engraved LET", Font.PLAIN, 38));
lblNewLabel_3.setBounds(528, 454, 504, 89);
panel.add(lblNewLabel_3);

```

```

}

/**
 * Metodo que hará la vista visible
 */
public void hacerVisible() {
    frame.setVisible(true);
}

/**
 * metodo que hará la vista invisible
 */
public void hacerInVisible() {
    frame.setVisible(false);
}

/**
 * metodo que se deshará de la vista
 */
public void dispose() {
    frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
    frame.setVisible(false);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá al home
 *
 * @param x la vista home
 */
public void setControlador(home x) {
    altasHome escuchador = new altasHome();
    escuchador.setVentanaUsuario(x, this);
    button1.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá al consultas
 *
 * @param x la vista consultas
 */
public void setControlador2(consultas x) {
    altasConsultas escuchador = new altasConsultas();
    escuchador.setVentanaUsuario(this, x);
    button3.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a altasAlumnos
 *
 * @param x la vista altasAlumnos
 */
public void setControlador3(alertasAlumnos x) {
    altasAlumnosB escuchador = new altasAlumnosB();
    escuchador.setVentanaUsuario(this, x);
    buttonadd1.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a altasAreas
 *
 * @param x la vista altasAreas
 */
public void setControlador5(alertasAreas x) {
    altasAreasB escuchador = new altasAreasB();
    escuchador.setVentanaUsuario(this, x);
    buttonadd2.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a altasPI
 *
 * @param x la vista altasPI
 */
public void setControlador4(alertasPI x) {
}

```

```

        altasPIB escuchador = new altasPIB();
        escuchador.setVentanaUsuario(this, x);
        buttonadd3.addActionListener(escuchador);
    }

    /**
     * getter del boton 1
     *
     * @return button1 el boton 1
     */
    public JButton getButton1() {
        return button1;
    }

    /**
     * getter del boton2
     *
     * @return button2 el boton 2
     */
    public JButton getButton2() {
        return button2;
    }

    /**
     * getter del boton 3
     *
     * @return button3 el boton 3
     */
    public JButton getButton3() {
        return button3;
    }

    /**
     * metodo que cambia la vista al home
     *
     * @param home la vista home
     */
    public void setHome(home home) {
        h = home;
        setControlador(h);
    }

    /**
     * metodo que cambia la vista a consultas
     *
     * @param con la vista consultas
     */
    public void setConsultas(consultas con) {
        c = con;
        // boton que va a consultas
        setControlador2(c);
    }

    /**
     * metodo que cambia la vista a altasAlumnos
     *
     * @param altaAlum la vista altasAlumnos
     */
    public void setAltasAlumnos(altasAlumnos altaAlum) {
        // Boton que va a altasAlumnos
        setControlador3(altaAlum);
    }

    /**
     * metodo que cambia la vista a altasPI
     *
     * @param altasPI la vista altasPI
     */
    public void setAltasPI(altasPI altasPI) {
        // Boton que va a altasPI
        setControlador4(altasPI);
    }

    /**

```

```

    * metodo que cambia la vista a altasArea
    *
    * @param altasArea la visya altasArea
    */
public void setAltasAreas(altasAreas altasArea) {
    // Boton que va a AltasAreas
    setControlador5(altasArea);
}
}

```

AltasAlumnos

```

package mvc.vistas;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement.ComponentPlacement;
import BBDD.AccesoBBDD;
import mvc.control.altasAlumnosAñadir;
import mvc.control.altasAlumnosB;
import mvc.control.altasAlumnosConsultas;
import mvc.control.añadirAlumnosHome;

/**
 * esta clase es la vista que dará de alta a los alumnos nuevos
 *
 * @author alexjun09
 *
 */
public class altasAlumnos {

    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;
    private JButton button4;
    private JButton buttonadd1;
    private JButton buttonadd2;
    private JButton buttonadd3;
    private JLabel lblNewLabel_1;
    private JLabel lblNewLabel_2;
    private JLabel lblNewLabel_3;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;

    // setter y getters habituales
    public JFrame getFrame() {
        return frame;
    }

    public void setFrame(JFrame frame) {
        this.frame = frame;
    }

    public JPanel getPanel() {
        return panel;
    }

    public void setPanel(JPanel panel) {
        this.panel = panel;
    }

    public JButton getButtonadd1() {
        return buttonadd1;
    }
}

```

```

public void setButtonadd1(JButton buttonadd1) {
    this.buttonadd1 = buttonadd1;
}

public JButton getButtonadd2() {
    return buttonadd2;
}

public void setButtonadd2(JButton buttonadd2) {
    this.buttonadd2 = buttonadd2;
}

public JButton getButtonadd3() {
    return buttonadd3;
}

public void setButtonadd3(JButton buttonadd3) {
    this.buttonadd3 = buttonadd3;
}

public JLabel getLblNewLabel_1() {
    return lblNewLabel_1;
}

public void setLblNewLabel_1(JLabel lblNewLabel_1) {
    this.lblNewLabel_1 = lblNewLabel_1;
}

public JLabel getLblNewLabel_2() {
    return lblNewLabel_2;
}

public void setLblNewLabel_2(JLabel lblNewLabel_2) {
    this.lblNewLabel_2 = lblNewLabel_2;
}

public JLabel getLblNewLabel_3() {
    return lblNewLabel_3;
}

public void setLblNewLabel_3(JLabel lblNewLabel_3) {
    this.lblNewLabel_3 = lblNewLabel_3;
}

public JTextField getTextField() {
    return textField;
}

public void setTextField(JTextField textField) {
    this.textField = textField;
}

public JTextField getTextField_1() {
    return textField_1;
}

public void setTextField_1(JTextField textField_1) {
    this.textField_1 = textField_1;
}

public JTextField getTextField_2() {
    return textField_2;
}

public void setTextField_2(JTextField textField_2) {
    this.textField_2 = textField_2;
}

public void setButton1(JButton button1) {
    this.button1 = button1;
}

public void setButton2(JButton button2) {

```

```

        this.button2 = button2;
    }

    public void setButton3(JButton button3) {
        this.button3 = button3;
    }

    public JTextField getTextField_3() {
        return textField_3;
    }

    public void setTextField_3(JTextField textField_3) {
        this.textField_3 = textField_3;
    }

    private JTextField textField_3;

    /**
     * constructor del objeto altas alumnos
     */
    public altasAlumnos() {

        //////////////// MENÚ Y FONDO///////////////////////////
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension screenSize = toolkit.getScreenSize();

        frame = new JFrame("Añadir Alumnos");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(1350, 820);

        panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {

                // Carga y dibuja la imagen de fondo
                ImageIcon imageIcon = new ImageIcon("Imagenes/AltasAlumnos.png");
                Image image = imageIcon.getImage();
                g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
                setVisible(true);
                super.paintComponent(g);
            }
        };

        // Cambia el icono de la ventana de Java
        ImageIcon icon = new ImageIcon("Imagenes/7.png");
        frame.setIconImage(icon.getImage());

        ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
        Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
        ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
        Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
        ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
        Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
        ImageIcon imagenAnadir = new ImageIcon("Imagenes/agregar.png");
        Image resizedAnadir = imagenAnadir.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

        button1 = new JButton("");
        button1.setBorderPainted(false);
        button1.setBounds(115, 677, 90, 78);
        button1.setIcon(new ImageIcon(resizedImage));
        button1.setContentAreaFilled(false);
        button1.setFocusPainted(false);
        button1.setOpaque(false);

        button2 = new JButton("");
        button2.setBorderPainted(false);
        button2.setBounds(600, 677, 90, 78);
        button2.setIcon(new ImageIcon(resizedAdd));
        button2.setContentAreaFilled(false);
        button2.setFocusPainted(false);
        button2.setOpaque(false);

        button3 = new JButton("");
    }
}

```

```

button3.setBorderPainted(false);
button3.setBounds(1085, 677, 90, 78);
button3.setIcon(new ImageIcon(resizedConsultar));
button3.setContentAreaFilled(false);
button3.setFocusPainted(false);
button3.setOpaque(false);

button4 = new JButton("");
button4.setBorderPainted(false);
button4.setOpaque(false);
button4.setFocusPainted(false);
button4.setIcon(new ImageIcon(resizedAnadir));
button4.setContentAreaFilled(false);

button4.setBounds(1236, 652, 90, 78);

frame.setContentPane(panel);
panel.setLayout(null);
button3.setIcon(new ImageIcon(resizedConsultar));

panel.add(button1);
panel.add(button2);
panel.add(button3);
panel.add(button4);

JLabel lblNewLabel = new JLabel("Nombre:");
lblNewLabel.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNewLabel.setBounds(59, 431, 174, 62);
panel.add(lblNewLabel);

JLabel lblNExpediente = new JLabel("Nº Expediente:");
lblNExpediente.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNExpediente.setBounds(59, 528, 229, 62);
panel.add(lblNExpediente);

JLabel lblIdAlumno = new JLabel("ID Alumno:");
lblIdAlumno.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblIdAlumno.setBounds(930, 431, 174, 62);
panel.add(lblIdAlumno);

textField = new JTextField();
textField.setBackground(new Color(211, 233, 250));
textField.setBounds(192, 445, 667, 39);
panel.add(textField);
textField.setColumns(10);

textField_1 = new JTextField();
textField_1.setColumns(10);
textField_1.setBackground(new Color(211, 233, 250));
textField_1.setBounds(283, 548, 319, 39);
panel.add(textField_1);

textField_2 = new JTextField();
textField_2.setColumns(10);
textField_2.setBackground(new Color(211, 233, 250));
textField_2.setBounds(1111, 445, 261, 39);
panel.add(textField_2);

JLabel lblIdProyectoExistente = new JLabel("ID Proyecto Existente:");
lblIdProyectoExistente.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblIdProyectoExistente.setBounds(687, 544, 299, 62);
panel.add(lblIdProyectoExistente);

textField_3 = new JTextField();
textField_3.setColumns(10);
textField_3.setBackground(new Color(211, 233, 250));
textField_3.setBounds(1006, 564, 299, 39);
panel.add(textField_3);

}

/**
 * metodo que hará visible la vista
 */

```

```

public void hacerVisible() {
    frame.setVisible(true);
}

/**
 * metodo que se deshará de la vista
 */
public void dispose() {
    frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
    frame.setVisible(false);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá al home
 *
 * @param x la vista home
 */
public void setControlador1(home x) {
    añadirAlumnosHome escuchador = new añadirAlumnosHome();
    escuchador.setVentanaUsuario(x, this);
    button1.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a altas
 *
 * @param x la vista altas
 */
public void setControlador2(alertas x) {
    altasAlumnosAñadir escuchador = new altasAlumnosAñadir();
    escuchador.setVentanaUsuario(x, this);
    button2.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a consultas
 *
 * @param x la vista consultas
 */
public void setControlador3(consultas x) {
    altasAlumnosConsultas escuchador = new altasAlumnosConsultas();
    escuchador.setVentanaUsuario(x, this);
    button3.addActionListener(escuchador);
}

/**
 * getter del botton 1
 *
 * @return button1 el boton 1
 */
public JButton getButton1() {
    return button1;
}

/**
 * getter del botton 2
 *
 * @return button2 el boton 2
 */
public JButton getButton2() {
    return button2;
}

/**
 * getter del botton 3
 *
 * @return button3 el boton 3
 */
public JButton getButton3() {
    return button3;
}

/**
 * metodo que cambia la vista al home

```

```

        *
        * @param home la vista home
        */
    public void setHome(home home) {
        // boton que vuelve a home
        setControlador1(home);
    }

    /**
     * metodo que cambia la vista a altas
     *
     * @param alta la vista altas
     */
    public void setAltas(altas alta) {
        // boton que va a altas
        setControlador2(alta);

    }

    /**
     * metodo que cambia la vista a consultas
     *
     * @param con la vista consultas
     */
    public void setConsultas(consultas con) {
        // boton que va a consultas
        setControlador3(con);

        final AccesoBBDD accesoBBDD = new AccesoBBDD(this);
        button4.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // Insertar los datos en la tabla ALUMNOS
                accesoBBDD.insertarDatosAlumnos();
            }
        });
    }
}

```

AltasAreas

```

package mvc.vistas;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;

import BBDD.AccesoBBDD;
import mvc.control.altasAreasAñadir;
import mvc.control.altasAreasConsultas;
import mvc.control.altasAreasHome;
import mvc.control.homeAñadir;
import mvc.control.homeConsultas;

/**
 * esta es la clase vista que dará de alta las areas
 *
 * @author alexjun09
 *
 */
public class altasAreas {

    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;

```

```

private JButton buttonadd1;
private JButton buttonadd2;
private JButton buttonadd3;
private JLabel lblNewLabel_1;
private JLabel lblNewLabel_2;
private JLabel lblNewLabel_3;
private JTextField textField;
private JTextField textField_1;
private JButton button1_1;

// setters y getters habituales
public JFrame getFrame() {
    return frame;
}

public void setFrame(JFrame frame) {
    this.frame = frame;
}

public JPanel getPanel() {
    return panel;
}

public void setPanel(JPanel panel) {
    this.panel = panel;
}

public JButton getButtonadd1() {
    return buttonadd1;
}

public void setButtonadd1(JButton buttonadd1) {
    this.buttonadd1 = buttonadd1;
}

public JButton getButtonadd2() {
    return buttonadd2;
}

public void setButtonadd2(JButton buttonadd2) {
    this.buttonadd2 = buttonadd2;
}

public JButton getButtonadd3() {
    return buttonadd3;
}

public void setButtonadd3(JButton buttonadd3) {
    this.buttonadd3 = buttonadd3;
}

public JLabel getLblNewLabel_1() {
    return lblNewLabel_1;
}

public void setLblNewLabel_1(JLabel lblNewLabel_1) {
    this.lblNewLabel_1 = lblNewLabel_1;
}

public JLabel getLblNewLabel_2() {
    return lblNewLabel_2;
}

public void setLblNewLabel_2(JLabel lblNewLabel_2) {
    this.lblNewLabel_2 = lblNewLabel_2;
}

public JLabel getLblNewLabel_3() {
    return lblNewLabel_3;
}

public void setLblNewLabel_3(JLabel lblNewLabel_3) {
    this.lblNewLabel_3 = lblNewLabel_3;
}

```

```

public JTextField getTextField() {
    return textField;
}

public void setTextField(JTextField textField) {
    this.textField = textField;
}

public JTextField getTextField_1() {
    return textField_1;
}

public void setTextField_1(JTextField textField_1) {
    this.textField_1 = textField_1;
}

public JButton getButton1_1() {
    return button1_1;
}

public void setButton1_1(JButton button1_1) {
    this.button1_1 = button1_1;
}

public void setButton1(JButton button1) {
    this.button1 = button1;
}

public void setButton2(JButton button2) {
    this.button2 = button2;
}

public void setButton3(JButton button3) {
    this.button3 = button3;
}

/**
 * constructor del objeto altasAreas
 */
public altasAreas() {

    //////////////// MENÚ Y FONDO///////////////////////////
    Toolkit toolkit = Toolkit.getDefaultToolkit();
    Dimension screenSize = toolkit.getScreenSize();

    frame = new JFrame("Añadir áreas");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1350, 820);

    panel = new JPanel() {
        @Override
        protected void paintComponent(Graphics g) {

            // Carga y dibuja la imagen de fondo
            ImageIcon imageIcon = new ImageIcon("Imagenes/AltasAreas.png");
            Image image = imageIcon.getImage();
            g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
            setOpaque(false);
            setVisible(true);
            super.paintComponent(g);
        }
    };

    // Cambia el icono de la ventana de Java
    ImageIcon icon = new ImageIcon("Imagenes/7.png");
    frame.setIconImage(icon.getImage());

    ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
    Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
    ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
    Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
    ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
    Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
}

```

```

button1 = new JButton("");
button1.setBorderPainted(false);
button1.setBounds(115, 677, 90, 78);
button1.setIcon(new ImageIcon(resizedImage));
button1.setContentAreaFilled(false);
button1.setFocusPainted(false);
button1.setOpaque(false);

button2 = new JButton("");
button2.setBorderPainted(false);
button2.setBounds(600, 677, 90, 78);
button2.setIcon(new ImageIcon(resizedAdd));
button2.setContentAreaFilled(false);
button2.setFocusPainted(false);
button2.setOpaque(false);

button3 = new JButton("");
button3.setBorderPainted(false);
button3.setBounds(1085, 677, 90, 78);
button3.setIcon(new ImageIcon(resizedConsultar));
button3.setContentAreaFilled(false);
button3.setFocusPainted(false);
button3.setOpaque(false);

ImageIcon imagenAnadirIcon = new ImageIcon("Imagenes/agregar.png");
Image      resizedAnadirImage      =      imagenAnadirIcon.getImage().getScaledInstance(60,      60,
Image.SCALE_SMOOTH);

button1_1 = new JButton("");
button1_1.setBorderPainted(false);
button1_1.setOpaque(false);
button1_1.setFocusPainted(false);
button1_1.setIcon(new ImageIcon(resizedAnadirImage));
button1_1.setContentAreaFilled(false);

button1_1.setBounds(1236, 556, 90, 78);

frame.setContentPane(panel);
panel.setLayout(null);

panel.add(button1);

panel.add(button1_1);
panel.add(button2);
panel.add(button3);

JLabel lblNewLabel = new JLabel("ID Área:");
lblNewLabel.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNewLabel.setBounds(59, 431, 174, 62);
panel.add(lblNewLabel);

JLabel lblNExpediente = new JLabel("Descripción:");
lblNExpediente.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNExpediente.setBounds(46, 528, 229, 62);
panel.add(lblNExpediente);

textField = new JTextField();
textField.setBackground(new Color(211, 233, 250));
textField.setBounds(192, 445, 319, 39);
panel.add(textField);
textField.setColumns(10);

textField_1 = new JTextField();
textField_1.setColumns(10);
textField_1.setBackground(new Color(211, 233, 250));
textField_1.setBounds(270, 539, 727, 108);
panel.add(textField_1);

}

/**
 * getter del boton 1
 *

```

```

    * @return button1 el boton 1
    */
public JButton getButton1() {
    return button1;
}

/**
 * getter del boton 2
 *
 * @return button2 el boton 2
 */
public JButton getButton2() {
    return button2;
}

/**
 * getter del boton 3
 *
 * @return button3 el boton 3
 */
public JButton getButton3() {
    return button3;
}

/**
 * metodo que hará la vista visible
 */
public void hacerVisible() {
    frame.setVisible(true);
}

/**
 * metodo que se deshará del la vista
 */
public void dispose() {
    frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
    frame.setVisible(false);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a altas
 *
 * @param x la vista altas
 */
public void setControlador1(alertas x) {
    altasAreasAñadir escuchador = new altasAreasAñadir();
    escuchador.setVentanaUsuario(this, x);
    button2.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a consultas
 *
 * @param x la vista home
 */
public void setControlador2(consultas x) {
    altasAreasConsultas escuchador2 = new altasAreasConsultas();
    escuchador2.setVentanaUsuario(this, x);
    button3.addActionListener(escuchador2);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá al home
 *
 * @param x la vista home
 */
public void setControlador3(home x) {
    altasAreasHome escuchador2 = new altasAreasHome();
    escuchador2.setVentanaUsuario(this, x);
    button1.addActionListener(escuchador2);
}

/**
 * metodo que cambia la vista a altas

```

```

        *
        * @param a la vista altas
        */
    public void setAltas(altas a) {
        // Boton que va a altas
        setControlador1(a);
    }

    /**
     * metodo que cambia la vista a consultas
     *
     * @param c la vista consultas
     */
    public void setConsultas(consultas c) {
        // Boton que va a consultas
        setControlador2(c);
    }

    /**
     * metodo que cambia la vista al gome
     *
     * @param h
     */
    public void setHome(home h) {
        // Boton que va a consultas
        setControlador3(h);

        final AccesoBBDD accesoBBDD = new AccesoBBDD(this);
        button1_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // Insertar los datos en la tabla AREAS
                accesoBBDD.insertarDatosAreas();
            }
        });
    }
}

```

AltasPI

```

package mvc.vistas;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.text.ParseException;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;
import BBDD.AccesoBBDD;
import mvc.control.altasAlumnosB;
import mvc.control.altasConsultas;
import mvc.control.altasHome;
import mvc.control.altasPIHome;
import mvc.control.altasPiAñadir;
import mvc.control.altasPiConsultas;

/**
 * Clase que contiene la vista de altasPI
 *
 * @author alexjun09
 *
 */
public class altasPI {

    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;

```

```

private JButton buttonadd1;
private JButton buttonadd2;
private JButton buttonadd3;
private JLabel lblNewLabel_1;
private JLabel lblNewLabel_2;
private JLabel lblNewLabel_3;
private JTextField textField;
private JTextField textField_1;
private JTextField textField_2;
private JButton button1_1;
private JLabel lblGrupo;
private JTextField textField_4;
private JLabel lblGrupo_1;
private JTextField textField_5;
private JLabel lblCurso;
private JTextField textField_6;
private JLabel lblIdProyecto;
private JTextField textField_7;
private JLabel lblGrupo_2;
private JTextField textField_8;

// setters y getters habituales
public JFrame getFrame() {
    return frame;
}

public void setFrame(JFrame frame) {
    this.frame = frame;
}

public JPanel getPanel() {
    return panel;
}

public void setPanel(JPanel panel) {
    this.panel = panel;
}

public JButton getButtonadd1() {
    return buttonadd1;
}

public void setButtonadd1(JButton buttonadd1) {
    this.buttonadd1 = buttonadd1;
}

public JButton getButtonadd2() {
    return buttonadd2;
}

public void setButtonadd2(JButton buttonadd2) {
    this.buttonadd2 = buttonadd2;
}

public JButton getButtonadd3() {
    return buttonadd3;
}

public void setButtonadd3(JButton buttonadd3) {
    this.buttonadd3 = buttonadd3;
}

public JLabel getLblNewLabel_1() {
    return lblNewLabel_1;
}

public void setLblNewLabel_1(JLabel lblNewLabel_1) {
    this.lblNewLabel_1 = lblNewLabel_1;
}

public JLabel getLblNewLabel_2() {
    return lblNewLabel_2;
}

```

```

public void setLblNewLabel_2(JLabel lblNewLabel_2) {
    this.lblNewLabel_2 = lblNewLabel_2;
}

public JLabel getLblNewLabel_3() {
    return lblNewLabel_3;
}

public void setLblNewLabel_3(JLabel lblNewLabel_3) {
    this.lblNewLabel_3 = lblNewLabel_3;
}

public JTextField getTextField() {
    return textField;
}

public void setTextField(JTextField textField) {
    this.textField = textField;
}

public JTextField getTextField_1() {
    return textField_1;
}

public void setTextField_1(JTextField textField_1) {
    this.textField_1 = textField_1;
}

public JTextField getTextField_2() {
    return textField_2;
}

public void setTextField_2(JTextField textField_2) {
    this.textField_2 = textField_2;
}

public JButton getButton1_1() {
    return button1_1;
}

public void setButton1_1(JButton button1_1) {
    this.button1_1 = button1_1;
}

public JLabel getLblGrupo() {
    return lblGrupo;
}

public void setLblGrupo(JLabel lblGrupo) {
    this.lblGrupo = lblGrupo;
}

public JTextField getTextField_4() {
    return textField_4;
}

public void setTextField_4(JTextField textField_4) {
    this.textField_4 = textField_4;
}

public JLabel getLblGrupo_1() {
    return lblGrupo_1;
}

public void setLblGrupo_1(JLabel lblGrupo_1) {
    this.lblGrupo_1 = lblGrupo_1;
}

public JTextField getTextField_5() {
    return textField_5;
}

public void setTextField_5(JTextField textField_5) {
    this.textField_5 = textField_5;
}

```

```

}

public JLabel getLblCurso() {
    return lblCurso;
}

public void setLblCurso(JLabel lblCurso) {
    this.lblCurso = lblCurso;
}

public JTextField getTextField_6() {
    return textField_6;
}

public void setTextField_6(JTextField textField_6) {
    this.textField_6 = textField_6;
}

public JLabel getLblIdProyecto() {
    return lblIdProyecto;
}

public void setLblIdProyecto(JLabel lblIdProyecto) {
    this.lblIdProyecto = lblIdProyecto;
}

public JTextField getTextField_7() {
    return textField_7;
}

public void setTextField_7(JTextField textField_7) {
    this.textField_7 = textField_7;
}

public JLabel getLblGrupo_2() {
    return lblGrupo_2;
}

public void setLblGrupo_2(JLabel lblGrupo_2) {
    this.lblGrupo_2 = lblGrupo_2;
}

public JTextField getTextField_8() {
    return textField_8;
}

public void setTextField_8(JTextField textField_8) {
    this.textField_8 = textField_8;
}

public void setButton1(JButton button1) {
    this.button1 = button1;
}

public void setButton2(JButton button2) {
    this.button2 = button2;
}

public void setButton3(JButton button3) {
    this.button3 = button3;
}

/**
 * constructor del objeto altasPI
 */
public altasPI() {

    //////////////// MENÚ Y FONDO/////////////////////////77
    Toolkit toolkit = Toolkit.getDefaultToolkit();
    Dimension screenSize = toolkit.getScreenSize();

    frame = new JFrame("Añadir áreas");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1350, 820);
}

```

```

panel = new JPanel() {
    @Override
    protected void paintComponent(Graphics g) {

        // Carga y dibuja la imagen de fondo
        ImageIcon imageIcon = new ImageIcon("Imagenes/AltasPI.png");
        Image image = imageIcon.getImage();
        g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
        setOpaque(false);
        setVisible(true);
        super.paintComponent(g);
    }
};

// Cambia el icono de la ventana de Java
ImageIcon icon = new ImageIcon("Imagenes/7.png");
frame.setIconImage(icon.getImage());

ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

button1 = new JButton("");
button1.setBorderPainted(false);
button1.setBounds(115, 677, 90, 78);
button1.setIcon(new ImageIcon(resizedImage));
button1.setContentAreaFilled(false);
button1.setFocusPainted(false);
button1.setOpaque(false);

button2 = new JButton("");
button2.setBorderPainted(false);
button2.setBounds(600, 677, 90, 78);
button2.setIcon(new ImageIcon(resizedAdd));
button2.setContentAreaFilled(false);
button2.setFocusPainted(false);
button2.setOpaque(false);

button3 = new JButton("");
button3.setBorderPainted(false);
button3.setBounds(1085, 677, 90, 78);
button3.setIcon(new ImageIcon(resizedConsultar));
button3.setContentAreaFilled(false);
button3.setFocusPainted(false);
button3.setOpaque(false);

ImageIcon imagenAnadirIcon = new ImageIcon("Imagenes/agregar.png");
Image     resizedAnadirImage      =     imagenAnadirIcon.getImage().getScaledInstance(60,       60,
Image.SCALE_SMOOTH);

button1_1 = new JButton("");
button1_1.setBorderPainted(false);
button1_1.setOpaque(false);
button1_1.setFocusPainted(false);
button1_1.setIcon(new ImageIcon(resizedAnadirImage));
button1_1.setContentAreaFilled(false);

button1_1.setBounds(1168, 628, 90, 78);

frame.setContentPane(panel);
panel.setLayout(null);

panel.add(button1);

panel.add(button1_1);
panel.add(button2);
panel.add(button3);

JLabel lblNewLabel = new JLabel("Nombre:");
lblNewLabel.setFont(new Font("SansSerif", Font.PLAIN, 30));

```

```

lblNewLabel.setBounds(46, 231, 174, 62);
panel.add(lblNewLabel);

JLabel lblNExpediente = new JLabel("URL:");
lblNExpediente.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNExpediente.setBounds(46, 528, 90, 62);
panel.add(lblNExpediente);

JLabel lblIdAlumno = new JLabel("ID Área:");
lblIdAlumno.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblIdAlumno.setBounds(46, 339, 174, 62);
panel.add(lblIdAlumno);

lblGrupo = new JLabel("Año:");
lblGrupo.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblGrupo.setBounds(441, 339, 83, 62);
panel.add(lblGrupo);

lblGrupo_1 = new JLabel("Grupo:");
lblGrupo_1.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblGrupo_1.setBounds(798, 332, 174, 62);
panel.add(lblGrupo_1);

textField = new JTextField();
textField.setBackground(new Color(211, 233, 250));
textField.setBounds(192, 249, 926, 39);
panel.add(textField);
textField.setColumns(10);

textField_1 = new JTextField();
textField_1.setColumns(10);
textField_1.setBackground(new Color(211, 233, 250));
textField_1.setBounds(128, 539, 1130, 39);
panel.add(textField_1);

textField_2 = new JTextField();
textField_2.setColumns(10);
textField_2.setBackground(new Color(211, 233, 250));
textField_2.setBounds(168, 350, 261, 39);
panel.add(textField_2);

textField_4 = new JTextField();
textField_4.setColumns(10);
textField_4.setBackground(new Color(211, 233, 250));
textField_4.setBounds(516, 350, 261, 39);
panel.add(textField_4);

textField_5 = new JTextField();
textField_5.setColumns(10);
textField_5.setBackground(new Color(211, 233, 250));
textField_5.setBounds(910, 350, 261, 39);
panel.add(textField_5);

lblCurso = new JLabel("Curso:");
lblCurso.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblCurso.setBounds(46, 420, 174, 62);
panel.add(lblCurso);

textField_6 = new JTextField();
textField_6.setColumns(10);
textField_6.setBackground(new Color(211, 233, 250));
textField_6.setBounds(168, 431, 261, 39);
panel.add(textField_6);

lblIdProyecto = new JLabel("ID Proyecto:");
lblIdProyecto.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblIdProyecto.setBounds(441, 420, 174, 62);
panel.add(lblIdProyecto);

textField_7 = new JTextField();
textField_7.setColumns(10);
textField_7.setBackground(new Color(211, 233, 250));
textField_7.setBounds(625, 437, 261, 39);
panel.add(textField_7);

```

```

        lblGrupo_2 = new JLabel("Nota:");
        lblGrupo_2.setFont(new Font("SansSerif", Font.PLAIN, 30));
        lblGrupo_2.setBounds(910, 420, 99, 62);
        panel.add(lblGrupo_2);

        textField_8 = new JTextField();
        textField_8.setColumns(10);
        textField_8.setBackground(new Color(211, 233, 250));
        textField_8.setBounds(1003, 431, 261, 39);
        panel.add(textField_8);

    }

    /**
     * getter del boton 1
     *
     * @return button1 boton 1
     */
    public JButton getButton1() {
        return button1;
    }

    /**
     * getter del boton 2
     *
     * @return button2
     */
    public JButton getButton2() {
        return button2;
    }

    /**
     * getter del boton 3
     *
     * @return button3
     */
    public JButton getButton3() {
        return button3;
    }

    /**
     * metodo que hace la vista visible
     */
    public void hacerVisible() {
        frame.setVisible(true);
    }

    /**
     * metodo que se deshace de la vista
     */
    public void dispose() {
        frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
        frame.setVisible(false);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a home
     *
     * @param x la vista home
     */
    public void setControlador(home x) {
        altasPIHome escuchador = new altasPIHome();
        escuchador.setVentanaUsuario(this, x);
        button1.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a consultas
     *
     * @param x la vista consultas
     */
    public void setControlador2(consultas x) {
        altasPiConsultas escuchador = new altasPiConsultas();

```

```

        escuchador.setVentanaUsuario(this, x);
        button3.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a altas
     *
     * @param x la vista altas
     */
    public void setControlador3(alertas x) {
        altasPiAñadir escuchador = new altasPiAñadir();
        escuchador.setVentanaUsuario(this, x);
        button2.addActionListener(escuchador);
    }

    /**
     * metodo que cambia la vista al home
     *
     * @param home la vista home
     */
    public void setHome(home home) {
        // boton que vuelve a home
        setControlador(home);
    }

    /**
     * metodo que cambia la vista a consultas
     *
     * @param home la vista consultas
     */
    public void setConsultas(consultas con) {
        // boton que va a consultas
        setControlador2(con);
    }

    /**
     * metodo que cambia la vista a altas
     *
     * @param altas la vista altas
     */
    public void setAltas(alertas altas) {
        // boton que va a altas
        setControlador3(altas);

        final AccesoBBDD accesoBBDD = new AccesoBBDD(this);
        button1_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // Insertar los datos en la tabla PROYECTO
                try {
                    accesoBBDD.insertarDatosPI();
                } catch (ParseException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }
            }
        });
    }
}

```

BuscarAlumnos

```

package mvc.vistas;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.util.ArrayList;
import java.awt.event.ActionEvent;

```

```

import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;

import BBDD.AccesoBBDD;
import mvc.control.buscarAlumnoEditarAlumno;
import mvc.control.buscarAlumnosAñadir;
import mvc.control.buscarAlumnosConsultas;
import mvc.control.buscarAlumnosHome;
import mvc.control.consultasAltas;
import mvc.control.consultasHome;
import mvc.modelo.datosAlumno;
import mvc.modelo.datosAreas;

/**
 * clase que contiene la vista de buscar alumnos
 *
 * @author alexjun09
 */
public class buscarAlumnos {
    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;
    private JLabel lblNewLabel_1;
    private JTextField txtHola;

    private JScrollPane scrollPane;
    public JList<datosAlumno> datosAlumnos;

    private home h;
    private altas a;
    private consultas c;

    /**
     * constructor del objeto altasAlumnos
     */
    public buscarAlumnos() {

        //////////////////////// MENÚ Y FONDO/////////////////////////////77
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension screenSize = toolkit.getScreenSize();

        frame = new JFrame("Consultar");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(1350, 820);

        panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {

                // Carga y dibuja la imagen de fondo
                ImageIcon imageIcon = new ImageIcon("Imagenes/FondoConsultarAlumnos.png");
                Image image = imageIcon.getImage();
                g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
                setVisible(true);
                super.paintComponent(g);
            }
        };

        // Cambia el icono de la ventana de Java
        ImageIcon icon = new ImageIcon("Imagenes/7.png");
        frame.setIconImage(icon.getImage());

        ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
        Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
        ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
        Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
        ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
        Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

        button1 = new JButton("");
    }
}

```

```

        button1.setBorderPainted(false);
        button1.setBounds(115, 677, 90, 78);
        button1.setIcon(new ImageIcon(resizedImage));
        button1.setContentAreaFilled(false);
        button1.setFocusPainted(false);
        button1.setOpaque(false);

        button2 = new JButton("");
        button2.setBorderPainted(false);
        button2.setBounds(600, 677, 90, 78);
        button2.setIcon(new ImageIcon(resizedAdd));
        button2.setContentAreaFilled(false);
        button2.setFocusPainted(false);
        button2.setOpaque(false);

        button3 = new JButton("");
        button3.setBorderPainted(false);
        button3.setBounds(1085, 677, 90, 78);
        button3.setIcon(new ImageIcon(resizedConsultar));
        button3.setContentAreaFilled(false);
        button3.setFocusPainted(false);
        button3.setOpaque(false);

        frame.setContentPane(panel);
        panel.setLayout(null);

        // BOTONES DE BUSCAR
        ImageIcon imagenAnadir = new ImageIcon("Imagenes/lupa.png");
        Image resizedAnadir = imagenAnadir.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
        panel.add(button1);
        panel.add(button2);
        panel.add(button3);

        JScrollPane panelScroll2 = new JScrollPane();
        panelScroll2.setBounds(46, 200, 1054, 437);
        frame.getContentPane().add(panelScroll2);

        datosAlumnos = new JList<datosAlumno>();
        panelScroll2.setViewportView(datosAlumnos);
        panelScroll2.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
        datosAlumnos.setFont(new Font("Verdana", Font.PLAIN, 18));

    }

    public void setListaAlumnos(ArrayList<datosAlumno> alumnos) {
        DefaultListModel<datosAlumno> modelo = new DefaultListModel<datosAlumno>();
        modelo.addAll(alumnos);
        // Esto no sé si puedes hacerlo o hay que hacer un foreach
        datosAlumnos.setModel(modelo);
    }

    /**
     * metodo que hace la vista invisible
     */
    public void hacerVisible() {
        frame.setVisible(true);
    }

    /**
     * metodo que se deshace de la vista
     */
    public void dispose() {
        frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
        frame.setVisible(false);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a home
     *
     * @param x la vista home
     */
    public void setControlador(home x) {
        buscarAlumnosHome escuchador = new buscarAlumnosHome();
        escuchador.setVentanaUsuario(this, x);
    }
}

```

```

        button1.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a altas
     *
     * @param x la vista altas
     */
    public void setControlador2(alertas x) {
        buscarAlumnosAñadir escuchador = new buscarAlumnosAñadir();
        escuchador.setVentanaUsuario(this, x);
        button2.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a consultas
     *
     * @param x la vista consultas
     */
    public void setControlador3(consultas x) {
        buscarAlumnosConsultas escuchador = new buscarAlumnosConsultas();
        escuchador.setVentanaUsuario(this, x);
        button3.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a editar alumnos
     *
     * @param x la vista editaralumnos
     */
    public void setControlador4(editarAlumnos x) {
        buscarAlumnoEditarAlumno escuchador = new buscarAlumnoEditarAlumno();
        escuchador.setVentana(this, x);
        datosAlumnos.addListSelectionListener(escuchador);
    }

    /**
     * metodo que cambia la vista a editar alumnos
     *
     * @param editarAlumnos la vista editarAlumnos
     */
    public void setEditarAlumnos(editarAlumnos editarAlumnos) {
        setControlador4(editarAlumnos);
    }

    public void setAltas(alertas altas) {
        a = altas;
        // boton que va a altas
        setControlador2(a);
    }

    /**
     * metodo que cambia la vista aal home
     *
     * @param altas la vista home
     */
    public void setHome(home home) {
        h = home;
        // boton que vuelve a home
        setControlador(h);
    }

    /**
     * metodo que cambia la vista a consultas
     *
     * @param consultas la vista consultas
     */
    public void setConsultas(consultas consultas) {
        c = consultas;
        setControlador3(c);
    }

    /**
     * getter del boton 1

```

```

        *
        * @return button1 el boton 1
        */
    public JButton getButton1() {
        return button1;
    }

    /**
     * getter del boton 2
     *
     * @return button2 el boton 2
     */
    public JButton getButton2() {
        return button2;
    }

    /**
     * getter del boton 3
     *
     * @return button3 el boton 3
     */
    public JButton getButton3() {
        return button3;
    }
}

```

BuscarArea

```

package mvc.vistas;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.util.ArrayList;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;

import mvc.control.buscarAreaAñadir;
import mvc.control.buscarAreaConsulta;
import mvc.control.buscarAreaEditarArea;
import mvc.control.buscarAreaHome;
import mvc.control.consultasAltas;
import mvc.control.consultasHome;
import mvc.modelo.datosAlumno;
import mvc.modelo.datosAreas;

/**
 * clase que contiene las vistas de buscarArea
 *
 * @author alexjun09
 */
public class buscarArea {
    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;
    private JLabel lblNewLabel_1;
    private JTextField txtHola;

    private JScrollPane scrollPane;
    public JList<datosAreas> datosAreas;

    private home h;
    private altas a;
    private consultas c;

    /**

```

```

* constructor del objeto buscarAreas
*/
public buscarArea() {

    //////////////// MENÚ Y FONDO///////////////////////////
    Toolkit toolkit = Toolkit.getDefaultToolkit();
    Dimension screenSize = toolkit.getScreenSize();

    frame = new JFrame("Consultar");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1350, 820);

    panel = new JPanel() {
        @Override
        protected void paintComponent(Graphics g) {

            // Carga y dibuja la imagen de fondo
            ImageIcon imageIcon = new ImageIcon("Imagenes/FondoConsultarAreas.png");
            Image image = imageIcon.getImage();
            g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
            setOpaque(false);
            setVisible(true);
            super.paintComponent(g);
        }
    };

    // Cambia el icono de la ventana de Java
    ImageIcon icon = new ImageIcon("Imagenes/7.png");
    frame.setIconImage(icon.getImage());

    ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
    Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
    ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
    Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
    ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
    Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

    button1 = new JButton("");
    button1.setBorderPainted(false);
    button1.setBounds(115, 677, 90, 78);
    button1.setIcon(new ImageIcon(resizedImage));
    button1.setContentAreaFilled(false);
    button1.setFocusPainted(false);
    button1.setOpaque(false);

    button2 = new JButton("");
    button2.setBorderPainted(false);
    button2.setBounds(600, 677, 90, 78);
    button2.setIcon(new ImageIcon(resizedAdd));
    button2.setContentAreaFilled(false);
    button2.setFocusPainted(false);
    button2.setOpaque(false);

    button3 = new JButton("");
    button3.setBorderPainted(false);
    button3.setBounds(1085, 677, 90, 78);
    button3.setIcon(new ImageIcon(resizedConsultar));
    button3.setContentAreaFilled(false);
    button3.setFocusPainted(false);
    button3.setOpaque(false);

    frame.setContentPane(panel);
    panel.setLayout(null);

    // BOTONES DE BUSCAR
    ImageIcon imagenAnadir = new ImageIcon("Imagenes/lupa.png");
    Image resizedAnadir = imagenAnadir.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
    panel.add(button1);
    panel.add(button2);
    panel.add(button3);

    JScrollPane panelScroll2 = new JScrollPane();
    panelScroll2.setBounds(46, 200, 1054, 437);
    frame.getContentPane().add(panelScroll2);
}

```

```

        datosAreas = new JList<datosAreas>();
        datosAreas.setFont(new Font("Verdana", Font.PLAIN, 18));
        panelScroll2.setViewportView(datosAreas);

        panelScroll2.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);

    }

    /**
     *
     * @param listaAreas
     */
    public void setListaArea(ArrayList<datosAreas> listaAreas) {
        DefaultListModel<datosAreas> modelo = new DefaultListModel<datosAreas>();
        modelo.addAll(listaAreas);
        // Esto no sé si puedes hacerlo o hay que hacer un foreach
        datosAreas.setModel(modelo);
    }

    /**
     * metodo que hace la vista visible
     */
    public void hacerVisible() {
        frame.setVisible(true);
    }

    /**
     * metodo que se deshace de la vista
     */
    public void dispose() {
        frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
        frame.setVisible(false);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a home
     *
     * @param x la vista home
     */
    public void setControlador(home x) {
        buscarAreaHome escuchador = new buscarAreaHome();
        escuchador.setVentanaUsuario(this, x);
        button1.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a altas
     *
     * @param x la vista altas
     */
    public void setControlador2(alertas x) {
        buscarAreaAñadir escuchador = new buscarAreaAñadir();
        escuchador.setVentanaUsuario(this, x);
        button2.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a consultas
     *
     * @param x la vista consultas
     */
    public void setControlador3(consultas x) {
        buscarAreaConsulta escuchador = new buscarAreaConsulta();
        escuchador.setVentanaUsuario(this, x);
        button3.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a editarAreas
     *
     * @param x la vista editarAreas
     */

```

```

public void setControlador4(editarAreas x) {
    buscarAreaEditarArea escuchador = new buscarAreaEditarArea();
    escuchador.setVentana(this, x);
    datosAreas.addListSelectionListener(escuchador);
}

/**
 * metodo que cambiará la vista a editarAreas
 *
 * @param editarAreas la vista editarAreas
 */
public void setEditarAreas(editarAreas editarAreas) {
    setControlador4(editarAreas);
}

/**
 * metodo que cambiará la vista a altas
 *
 * @param altas la vista altas
 */
public void setAltas(alertas altas) {
    a = altas;
    // boton que va a altas
    setControlador2(a);
}

/**
 * metodo que cambiará a la vista home
 *
 * @param home la vista home
 */
public void setHome(home home) {
    h = home;
    // boton que vuelve a home
    setControlador(h);
}

/**
 * metodo que cambiará la vista a consultas
 *
 * @param consultas la vista consultas
 */
public void setConsultas(consultas consultas) {
    c = consultas;
    setControlador3(c);
}

/**
 * getter del boton 1
 *
 * @return button1 el boton 1
 */
public JButton getButton1() {
    return button1;
}

/**
 * getter del boton 2
 *
 * @return button2 el boton 2
 */
public JButton getButton2() {
    return button2;
}

/**
 * getter del boton 3
 *
 * @return button3 el boton 3
 */
public JButton getButton3() {
    return button3;
}
}

```

BuscarPI

```
package mvc.vistas;

import javax.swing.*;

import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.util.ArrayList;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.LayoutStyle.ComponentPlacement;

import mvc.control.buscarAlumnoEditarAlumno;
import mvc.control.buscarPIAñadir;
import mvc.control.buscarPIConsultas;
import mvc.control.buscarPIEditarPI;
import mvc.control.buscarPIHome;
import mvc.control.consultasAltas;
import mvc.control.consultasHome;
import mvc.modelo.datosAlumno;
import mvc.modelo.datosAreas;
import mvc.modelo.datosPI;

/**
 * clase que contiene las vistas de buscarPI
 *
 * @author alexjun09
 *
 */
public class buscarPI {
    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;
    private JLabel lblNewLabel_1;
    private JTextField txtHola;

    private JScrollPane scrollPane;
    public JList<datosPI> datosPI;

    private home h;
    private altas a;
    private consultas c;

    /**
     * constructor del objeto buscarPI
     */
    public buscarPI() {

        //////////////// MENÚ Y FONDO///////////////
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension screenSize = toolkit.getScreenSize();

        frame = new JFrame("Consultar");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(1350, 820);

        panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {

                // Carga y dibuja la imagen de fondo
                ImageIcon imageIcon = new ImageIcon("Imagenes/FondoConsultaProyectos.png");
                Image image = imageIcon.getImage();
                g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
            }
        };
    }
}
```

```

        setVisible(true);
        super.paintComponent(g);
    }
};

// Cambia el icono de la ventana de Java
ImageIcon icon = new ImageIcon("Imagenes/7.png");
frame.setIconImage(icon.getImage());

ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

button1 = new JButton("");
button1.setBorderPainted(false);
button1.setBounds(115, 677, 90, 78);
button1.setIcon(new ImageIcon(resizedImage));
button1.setContentAreaFilled(false);
button1.setFocusPainted(false);
button1.setOpaque(false);

button2 = new JButton("");
button2.setBorderPainted(false);
button2.setBounds(600, 677, 90, 78);
button2.setIcon(new ImageIcon(resizedAdd));
button2.setContentAreaFilled(false);
button2.setFocusPainted(false);
button2.setOpaque(false);

button3 = new JButton("");
button3.setBorderPainted(false);
button3.setBounds(1085, 677, 90, 78);
button3.setIcon(new ImageIcon(resizedConsultar));
button3.setContentAreaFilled(false);
button3.setFocusPainted(false);
button3.setOpaque(false);

frame.setContentPane(panel);
panel.setLayout(null);

// BOTONES DE BUSCAR
ImageIcon imagenAnadir = new ImageIcon("Imagenes/lupa.png");
Image resizedAnadir = imagenAnadir.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
panel.add(button1);
panel.add(button2);
panel.add(button3);

JScrollPane panelScroll2 = new JScrollPane();
panelScroll2.setBounds(46, 200, 1054, 437);
frame.getContentPane().add(panelScroll2);

}

datosPI = new JList<datosPI>();
panelScroll2.setViewportView(datosPI);
panelScroll2.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
datosPI.setFont(new Font("Verdana", Font.PLAIN, 18));

private DefaultListModel<datosPI> modeloPI; // Agrega esta variable de instancia

/**
 *
 * @param listaPI
 */
public void setListaPI(ArrayList<datosPI> listaPI) {
    if (modeloPI == null) {

```

```

        modeloPI = new DefaultListModel<datosPI>();
        datosPI.setModel(modeloPI);
    }

    modeloPI.clear();
    modeloPI.addAll(listaPI);
}

/***
 * metodo que hará la vista visible
 */
public void hacerVisible() {
    frame.setVisible(true);
}

/***
 * metodo que se deshará de la vista
 */
public void dispose() {
    frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
    frame.setVisible(false);
}

/***
 * metodo que creará una nuevo controlador (boton) que redigirá a home
 *
 * @param x la vista home
 */
public void setControlador(home x) {
    buscarPIHome escuchador = new buscarPIHome();
    escuchador.setVentanaUsuario(this, x);
    button1.addActionListener(escuchador);
}

/***
 * metodo que creará una nuevo controlador (boton) que redigirá a altas
 *
 * @param x la vista alta
 */
public void setControlador2(alertas x) {
    buscarPIAñadir escuchador = new buscarPIAñadir();
    escuchador.setVentanaUsuario(this, x);
    button2.addActionListener(escuchador);
}

/***
 * metodo que creará una nuevo controlador (boton) que redigirá a consultas
 *
 * @param x la vista consultas
 */
public void setControlador3(consultas x) {
    buscarPIConsultas escuchador = new buscarPIConsultas();
    escuchador.setVentanaUsuario(this, x);
    button3.addActionListener(escuchador);
}

/***
 * metodo que creará una nuevo controlador (boton) que redigirá a editarPI
 *
 * @param x la vista editarPI
 */
public void setControlador4(editarPI x) {
    buscarPIEditarPI escuchador = new buscarPIEditarPI();
    escuchador.setVentana(this, x);
    datosPI.addListSelectionListener(escuchador);
}

/***
 * metodo que cambia de vista a editar pi
 *
 * @param editarPI la vista editarPI
 */
public void setEditarPI(editarPI editarPI) {
    setControlador4(editarPI);
}

```

```

    }

    /**
     * metodo que cambia de vista a altas
     *
     * @param altas la vista altas
     */
    public void setAltas(altas altas) {
        a = altas;
        // boton que va a altas
        setControlador2(a);
    }

    /**
     * metodo que cambia la vista al home
     *
     * @param home la vista home
     */
    public void setHome(home home) {
        h = home;
        // boton que vuelve a home
        setControlador(h);
    }

    /**
     * metodo que cambia la vista a consultas
     *
     * @param consultas la vista consultas
     */
    public void setConsultas(consultas consultas) {
        c = consultas;
        setControlador3(c);
    }

    /**
     * getter del boton 1
     *
     * @return button1 el boton 1
     */
    public JButton getButton1() {
        return button1;
    }

    /**
     * getter del boton 2
     *
     * @return button2 el boton 2
     */
    public JButton getButton2() {
        return button2;
    }

    /**
     * getter del boton 3
     *
     * @return button3 el boton 3
     */
    public JButton getButton3() {
        return button3;
    }
}

```

Consultas

```
package mvc.vistas;
```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.LayoutStyle.ComponentPlacement;

import mvc.control.altasHome;
import mvc.control.consultasAltas;
import mvc.control.consultasAlumnosB;
import mvc.control.consultasAreasB;
import mvc.control.consultasHome;
import mvc.control.consultasPIB;

/**
 * clase que contiene las vistas de consultas y sus correspondientes metodos
 *
 * @author alexjun09
 *
 */
public class consultas {
    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;
    private JButton buttonadd1;
    private JButton buttonadd2;
    private JButton buttonadd3;
    private JLabel lblNewLabel_1;
    private JLabel lblNewLabel_2;
    private JLabel lblNewLabel_3;

    private home h;
    private altas a;

    /**
     * constructor del objeto consultas
     */
    public consultas() {
        // this.h = home;
        //////////////////////////////// MENÚ Y FONDO////////////////////////////77
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension screenSize = toolkit.getScreenSize();

        frame = new JFrame("Consultar");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(1350, 820);

        panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {

                // Carga y dibuja la imagen de fondo
                ImageIcon imageIcon = new ImageIcon("Imagenes/fondo.png");
                Image image = imageIcon.getImage();
                g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
                setVisible(true);
                super.paintComponent(g);
            }
        };

        // Cambia el icono de la ventana de Java
        ImageIcon icon = new ImageIcon("Imagenes/7.png");
        frame.setIconImage(icon.getImage());

        ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
        Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
        ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
        Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
        ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
    }
}

```

```

Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

button1 = new JButton("");
button1.setBorderPainted(false);
button1.setBounds(115, 677, 90, 78);
button1.setIcon(new ImageIcon(resizedImage));
button1.setContentAreaFilled(false);
button1.setFocusPainted(false);
button1.setOpaque(false);

button2 = new JButton("");
button2.setBorderPainted(false);
button2.setBounds(600, 677, 90, 78);
button2.setIcon(new ImageIcon(resizedAdd));
button2.setContentAreaFilled(false);
button2.setFocusPainted(false);
button2.setOpaque(false);

button3 = new JButton("");
button3.setBorderPainted(false);
button3.setBounds(1085, 677, 90, 78);
button3.setIcon(new ImageIcon(resizedConsultar));
button3.setContentAreaFilled(false);
button3.setFocusPainted(false);
button3.setOpaque(false);

frame.setContentPane(panel);
panel.setLayout(null);

// BOTONES DE BUSCAR
ImageIcon imagenAnadir = new ImageIcon("Imagenes/lupa.png");
Image resizedAnadir = imagenAnadir.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

buttonadd1 = new JButton("");
buttonadd1.setBorderPainted(false);
buttonadd1.setBounds(426, 185, 90, 78);
buttonadd1.setIcon(new ImageIcon(resizedAnadir));
buttonadd1.setContentAreaFilled(false);
buttonadd1.setFocusPainted(false);
buttonadd1.setOpaque(false);

panel.add(buttonadd1);

buttonadd2 = new JButton("");
buttonadd2.setBorderPainted(false);
buttonadd2.setBounds(426, 315, 90, 78);
buttonadd2.setIcon(new ImageIcon(resizedAnadir));
buttonadd2.setContentAreaFilled(false);
buttonadd2.setFocusPainted(false);
buttonadd2.setOpaque(false);
panel.add(buttonadd2);

buttonadd3 = new JButton("");
buttonadd3.setBorderPainted(false);
buttonadd3.setBounds(426, 452, 90, 78);
buttonadd3.setIcon(new ImageIcon(resizedAnadir));
buttonadd3.setContentAreaFilled(false);
buttonadd3.setFocusPainted(false);
buttonadd3.setOpaque(false);
panel.add(buttonadd3);
panel.add(button1);
panel.add(button2);
panel.add(button3);

////////////////////////////// LABELS////////////////////////////

JLabel lblNewLabel = new JLabel("{Qué desea consultar?}");
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setForeground(new Color(247, 254, 255));
lblNewLabel.setFont(new Font("Kefa", Font.PLAIN, 63));
lblNewLabel.setBounds(260, 6, 917, 119);
panel.add(lblNewLabel);

lblNewLabel_1 = new JLabel("Alumnos");

```

```

        lblNewLabel_1.setFont(new Font("Academy Engraved LET", Font.PLAIN, 38));
        lblNewLabel_1.setBounds(528, 188, 285, 89);
        panel.add(lblNewLabel_1);

        lblNewLabel_2 = new JLabel("Áreas");
        lblNewLabel_2.setFont(new Font("Academy Engraved LET", Font.PLAIN, 38));
        lblNewLabel_2.setBounds(527, 304, 179, 89);
        panel.add(lblNewLabel_2);

        lblNewLabel_3 = new JLabel("Proyectos Integradores");
        lblNewLabel_3.setFont(new Font("Academy Engraved LET", Font.PLAIN, 38));
        lblNewLabel_3.setBounds(526, 441, 859, 89);
        panel.add(lblNewLabel_3);
        frame.setVisible(false);

    }

    /**
     * metodo que hace la vista visible
     */
    public void hacerVisible() {
        frame.setVisible(true);
    }

    /**
     * metodo que se deshace de la vista
     */
    public void dispose() {
        frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
        frame.setVisible(false);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a home
     *
     * @param x la vista home
     */
    public void setControlador(home x) {
        consultasHome escuchador = new consultasHome();
        escuchador.setVentanaUsuario(x, this);
        button1.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a altas
     *
     * @param x la vista altas
     */
    public void setControlador2(alertas x) {
        consultasAltas escuchador = new consultasAltas();
        escuchador.setVentanaUsuario(x, this);
        button2.addActionListener(escuchador);
    }

    /**
     * getter del boton 1
     *
     * @return button1 el boton 1
     */
    public JButton getButton1() {
        return button1;
    }

    /**
     * getter del boton 2
     *
     * @return button2 el boton 2
     */
    public JButton getButton2() {
        return button2;
    }

    /**
     * getter del boton 3
     */

```

```

*
 * @return button3 el boton 3
 */
public JButton getButton3() {
    return button3;
}

/**
 * metodo que cambia la vista a altas
 *
 * @param altas la vista altas
 */
public void setAltas(altas altas) {
    a = altas;
    // boton que va a altas
    setControlador2(a);
}

/**
 * metodo que cambia la vista al home
 *
 * @param home la vista home
 */
public void setHome(home home) {
    h = home;
    // boton que vuelve a home
    setControlador(h);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a buscarAlumnos
 *
 * @param x la vista buscar alumnos
 */
public void setControlador3(buscarAlumnos x) {
    consultasAlumnosB escuchador = new consultasAlumnosB();
    escuchador.setVentanaUsuario(this, x);
    buttonadd1.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a buscarAreas
 *
 * @param x la vista buscarArea
 */
public void setControlador4(buscarArea x) {
    consultasAreasB escuchador = new consultasAreasB();
    escuchador.setVentanaUsuario(this, x);
    buttonadd2.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a buscarPI
 *
 * @param x la vista buscarPI
 */
public void setControlador5(buscarPI x) {
    consultasPIB escuchador = new consultasPIB();
    escuchador.setVentanaUsuario(this, x);
    buttonadd3.addActionListener(escuchador);
}

/**
 * metodo que cambiará de vista a buscarArea
 *
 * @param x la vista buscarArea
 */
public void setBuscarAreas(buscarArea x) {
    // boton que vuelve a home
    setControlador4(x);
}

/**
 * metodo que cambiará de vista a buscarAlumnos

```

```

        *
        * @param x la vista buscarAlumnos
        */
    public void setBuscarAlumnos(buscarAlumnos x) {
        // boton que vuelve a home
        setControlador3(x);
    }

    /**
     * metodo que cambiará de vista a buscarPI
     *
     * @param x la vista buscarPI
     */
    public void setBuscarPI(buscarPI x) {
        // boton que vuelve a home
        setControlador5(x);
    }
}

```

EditarAlumnos

```

package mvc.vistas;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;
import BBDD.AccesoBBDD;
import mvc.control.altasAreasAñadir;
import mvc.control.altasAreasConsultas;
import mvc.control.buscarAlumnoEditarAlumno;
import mvc.control.consultaAlumnoHome;
import mvc.control.consultasAlumnoAñadir;
import mvc.control.consultasAlumnoConsultas;
import mvc.control.consultasHome;
import mvc.modelo.datosAlumno;

/**
 * clase que contiene las vistas de editar alumnos
 *
 * @author alexjun09
 *
 */
public class editarAlumnos {

    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;
    private JButton button4;
    private JButton button5;
    private JButton buttonadd1;
    private JButton buttonadd2;
    private JButton buttonadd3;
    private JLabel lblNewLabel_1;
    private JLabel lblNewLabel_2;
    private JLabel lblNewLabel_3;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JLabel lblNuevoIdAlumno;
    private JTextField textField_4;

    // setters y getters habituales
    public JLabel getLblNuevoIdAlumno() {

```

```

        return lblNuevoIdAlumno;
    }

    public void setLblNuevoIdAlumno(JLabel lblNuevoIdAlumno) {
        this.lblNuevoIdAlumno = lblNuevoIdAlumno;
    }

    public JTextField getTextField_4() {
        return textField_4;
    }

    public void setTextField_4(JTextField textField_4) {
        this.textField_4 = textField_4;
    }

    public JFrame getFrame() {
        return frame;
    }

    public void setFrame(JFrame frame) {
        this.frame = frame;
    }

    public JPanel getPanel() {
        return panel;
    }

    public void setPanel(JPanel panel) {
        this.panel = panel;
    }

    public JButton getButtonadd1() {
        return buttonadd1;
    }

    public void setButtonadd1(JButton buttonadd1) {
        this.buttonadd1 = buttonadd1;
    }

    public JButton getButtonadd2() {
        return buttonadd2;
    }

    public void setButtonadd2(JButton buttonadd2) {
        this.buttonadd2 = buttonadd2;
    }

    public JButton getButtonadd3() {
        return buttonadd3;
    }

    public void setButtonadd3(JButton buttonadd3) {
        this.buttonadd3 = buttonadd3;
    }

    public JLabel getLblNewLabel_1() {
        return lblNewLabel_1;
    }

    public void setLblNewLabel_1(JLabel lblNewLabel_1) {
        this.lblNewLabel_1 = lblNewLabel_1;
    }

    public JLabel getLblNewLabel_2() {
        return lblNewLabel_2;
    }

    public void setLblNewLabel_2(JLabel lblNewLabel_2) {
        this.lblNewLabel_2 = lblNewLabel_2;
    }

    public JLabel getLblNewLabel_3() {
        return lblNewLabel_3;
    }
}

```

```

public void setlblNewLabel_3(JLabel lblNewLabel_3) {
    this.lblNewLabel_3 = lblNewLabel_3;
}

public JTextField getTextField() {
    return textField;
}

public void setTextField(JTextField textField) {
    this.textField = textField;
}

public JTextField getTextField_1() {
    return textField_1;
}

public void setTextField_1(JTextField textField_1) {
    this.textField_1 = textField_1;
}

public JTextField getTextField_2() {
    return textField_2;
}

public void setTextField_2(JTextField textField_2) {
    this.textField_2 = textField_2;
}

public void setButton1(JButton button1) {
    this.button1 = button1;
}

public void setButton2(JButton button2) {
    this.button2 = button2;
}

public void setButton3(JButton button3) {
    this.button3 = button3;
}

public JButton getButton4() {
    return button4;
}

public void setButton4(JButton button4) {
    this.button4 = button4;
}

public JButton getButton5() {
    return button5;
}

public void setButton5(JButton button5) {
    this.button5 = button5;
}

public JTextField getTextField_3() {
    return textField_3;
}

public void setTextField_3(JTextField textField_3) {
    this.textField_3 = textField_3;
}

/**
 * constructor del objeto editarAlumnos
 */
public editarAlumnos() {

    //////////////// MENÚ Y FONDO///////////////////////////
    Toolkit toolkit = Toolkit.getDefaultToolkit();
    Dimension screenSize = toolkit.getScreenSize();
}

```

```

frame = new JFrame("Modificar Alumnos");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(1350, 820);

panel = new JPanel() {
    @Override
    protected void paintComponent(Graphics g) {

        // Carga y dibuja la imagen de fondo
        ImageIcon imageIcon = new ImageIcon("Imagenes/modAlum.png");
        Image image = imageIcon.getImage();
        g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
        setOpaque(false);
        setVisible(true);
        super.paintComponent(g);
    }
};

// Cambia el icono de la ventana de Java
ImageIcon icon = new ImageIcon("Imagenes/7.png");
frame.setIconImage(icon.getImage());

ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenAnadir = new ImageIcon("Imagenes/agregar.png");
Image resizedAnadir = imagenAnadir.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

button1 = new JButton("");
button1.setBorderPainted(false);
button1.setBounds(115, 677, 90, 78);
button1.setIcon(new ImageIcon(resizedImage));
button1.setContentAreaFilled(false);
button1.setFocusPainted(false);
button1.setOpaque(false);

button2 = new JButton("");
button2.setBorderPainted(false);
button2.setBounds(600, 677, 90, 78);
button2.setIcon(new ImageIcon(resizedAdd));
button2.setContentAreaFilled(false);
button2.setFocusPainted(false);
button2.setOpaque(false);

button3 = new JButton("");
button3.setBorderPainted(false);
button3.setBounds(1085, 677, 90, 78);
button3.setIcon(new ImageIcon(resizedConsultar));
button3.setContentAreaFilled(false);
button3.setFocusPainted(false);
button3.setOpaque(false);

ImageIcon imagenModificarIcon = new ImageIcon("Imagenes/lapiz.png");
Image     resizedModImage      =     imagenModificarIcon.getImage().getScaledInstance(70,      70,
Image.SCALE_SMOOTH);

button4 = new JButton("");
button4.setOpaque(false);
button4.setFocusPainted(false);
button4.setContentAreaFilled(false);
button4.setIcon(new ImageIcon(resizedModImage));
button4.setBorderPainted(false);
button4.setBounds(1166, 551, 43, 39);

ImageIcon imagenEliminarIcon = new ImageIcon("Imagenes/cancelar.png");
Image     resizedEliminarImage      =     imagenEliminarIcon.getImage().getScaledInstance(70,      70,
Image.SCALE_SMOOTH);

button5 = new JButton("");
button5.setBorderPainted(false);
button5.setOpaque(false);

```

```

button5.setFocusPainted(false);
button5.setIcon(new ImageIcon(resizedEliminarImage));
button5.setContentAreaFilled(false);
button5.setBounds(1229, 551, 43, 39);

frame.setContentPane(panel);
panel.setLayout(null);

panel.add(button1);
panel.add(button2);
panel.add(button3);
panel.add(button4);
panel.add(button5);

JLabel lblNewLabel = new JLabel("Nombre:");
lblNewLabel.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNewLabel.setBounds(46, 431, 174, 62);
panel.add(lblNewLabel);

JLabel lblNExpediente = new JLabel("Nº Expediente:");
lblNExpediente.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNExpediente.setBounds(46, 528, 229, 62);
panel.add(lblNExpediente);

JLabel lblIdAlumno = new JLabel("ID Alumno:");
lblIdAlumno.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblIdAlumno.setBounds(533, 431, 174, 62);
panel.add(lblIdAlumno);

textField = new JTextField();
textField.setBackground(new Color(211, 233, 250));
textField.setBounds(176, 451, 303, 39);
panel.add(textField);
textField.setColumns(10);

textField_1 = new JTextField();
textField_1.setColumns(10);
textField_1.setBackground(new Color(211, 233, 250));
textField_1.setBounds(270, 539, 319, 39);
panel.add(textField_1);

textField_2 = new JTextField();
textField_2.setColumns(10);
textField_2.setBackground(new Color(211, 233, 250));
textField_2.setBounds(717, 451, 123, 39);
panel.add(textField_2);

JLabel lblIdProyecto = new JLabel("ID Proyecto:");
lblIdProyecto.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblIdProyecto.setBounds(625, 528, 174, 62);
panel.add(lblIdProyecto);

textField_3 = new JTextField();
textField_3.setColumns(10);
textField_3.setBackground(new Color(211, 233, 250));
textField_3.setBounds(809, 548, 319, 39);
panel.add(textField_3);

lblNuevoIdAlumno = new JLabel("Nuevo ID Alumno:");
lblNuevoIdAlumno.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNuevoIdAlumno.setBounds(867, 431, 240, 62);
panel.add(lblNuevoIdAlumno);

textField_4 = new JTextField();
textField_4.setColumns(10);
textField_4.setBackground(new Color(211, 233, 250));
textField_4.setBounds(1117, 451, 123, 39);
panel.add(textField_4);

frame.setVisible(false);

}

/**

```

```

    * getter del boton 1
    *
    * @return button1 el boton 1
    */
public JButton getButton1() {
    return button1;
}

/**
 * getter del boton 2
 *
 * @return button2 el boton 2
 */
public JButton getButton2() {
    return button2;
}

/**
 * getter del boton 2
 *
 * @return button2 el boton 2
 */
public JButton getButton3() {
    return button3;
}

/**
 * metodo que hace la vista visible
 */
public void hacerVisible() {
    frame.setVisible(true);
}

/**
 * metodo que se deshace de la vista
 */
public void dispose() {
    frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
    frame.setVisible(false);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a editar home
 *
 * @param x vista home
 */
public void setControlador(home x) {
    consultaAlumnoHome escuchador = new consultaAlumnoHome();
    escuchador.setVentanaUsuario(x, this);
    button1.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a editar altas
 *
 * @param x vista altas
 */
public void setControlador1(alertas x) {
    consultasAlumnoAñadir escuchador = new consultasAlumnoAñadir();
    escuchador.setVentanaUsuario(this, x);
    button2.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a editar
 * consultas
 *
 * @param x vista consultas
 */
public void setControlador2(consultas x) {
    consultasAlumnoConsultas escuchador2 = new consultasAlumnoConsultas();
    escuchador2.setVentanaUsuario(this, x);
    button3.addActionListener(escuchador2);
}

```

```

    /**
     * cambia la vista a la vista home
     *
     * @param x vista home
     */
    public void setHome(home x) {
        setControlador(x);
    }

    /**
     * cambia la vista a la vista altas
     *
     * @param a vista altas
     */
    public void setAltas(alertas a) {
        // Boton que va a altas
        setControlador1(a);
    }

    /**
     * cambia la vista a la vista consultas
     *
     * @param c vista consultas
     */
    public void setConsultas(consultas c) {
        // Boton que va a consultas
        setControlador2(c);

        final AccesoBBDD accesoBBDD = new AccesoBBDD(this);
        button4.addActionListener(new ActionListener() {
            /**
             * metodo action performed que accede a la base de datos y edita los datos de
             * alumnos
             */
            public void actionPerformed(ActionEvent e) {
                // Insertar los datos en la tabla ALUMNOS
                accesoBBDD.editarDatosAlumnos();
            }
        });
        button5.addActionListener(new ActionListener() {
            /**
             * metodo que accede a la base de datos y borra alumnos
             */
            public void actionPerformed(ActionEvent e) {
                // Insertar los datos en la tabla ALUMNOS
                accesoBBDD.borrarDatosAlumnos();
            }
        });
    }
}

```

EditarAreas

```

package mvc.vistas;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;
import BBDD.AccesoBBDD;
import mvc.control.consultaAreasAñadir;
import mvc.control.consultaAreasHome;
import mvc.control.consultasAlumnosB;
import mvc.control.consultasAreasB;
import mvc.control.consultasAreasConsultas;

```

```

/**
 * clase que contiene las vistas de editar areas
 *
 * @author alexjun09
 *
 */
public class editarAreas {

    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;
    private JButton buttonadd1;
    private JButton buttonadd2;
    private JButton buttonadd3;
    private JLabel lblNewLabel_1;
    private JLabel lblNewLabel_2;
    private JLabel lblNewLabel_3;
    private JTextField textField;
    private JTextField textField_1;
    private JButton button1_1;
    private JButton button1_2;
    private JLabel lblNuevoIdrea;
    private JTextField textField_2;

    // setters y getters habituales
    public JFrame getFrame() {
        return frame;
    }

    public void setFrame(JFrame frame) {
        this.frame = frame;
    }

    public JPanel getPanel() {
        return panel;
    }

    public void setPanel(JPanel panel) {
        this.panel = panel;
    }

    public JButton getButtonadd1() {
        return buttonadd1;
    }

    public void setButtonadd1(JButton buttonadd1) {
        this.buttonadd1 = buttonadd1;
    }

    public JButton getButtonadd2() {
        return buttonadd2;
    }

    public void setButtonadd2(JButton buttonadd2) {
        this.buttonadd2 = buttonadd2;
    }

    public JButton getButtonadd3() {
        return buttonadd3;
    }

    public void setButtonadd3(JButton buttonadd3) {
        this.buttonadd3 = buttonadd3;
    }

    public JLabel getLblNewLabel_1() {
        return lblNewLabel_1;
    }

    public void setLblNewLabel_1(JLabel lblNewLabel_1) {
        this.lblNewLabel_1 = lblNewLabel_1;
    }
}

```

```

}

public JLabel getLblNewLabel_2() {
    return lblNewLabel_2;
}

public void setLblNewLabel_2(JLabel lblNewLabel_2) {
    this.lblNewLabel_2 = lblNewLabel_2;
}

public JLabel getLblNewLabel_3() {
    return lblNewLabel_3;
}

public void setLblNewLabel_3(JLabel lblNewLabel_3) {
    this.lblNewLabel_3 = lblNewLabel_3;
}

public JTextField getTextField() {
    return textField;
}

public void setTextField(JTextField textField) {
    this.textField = textField;
}

public JTextField getTextField_1() {
    return textField_1;
}

public void setTextField_1(JTextField textField_1) {
    this.textField_1 = textField_1;
}

public JButton getButton1_1() {
    return button1_1;
}

public void setButton1_1(JButton button1_1) {
    this.button1_1 = button1_1;
}

public JButton getButton1_2() {
    return button1_2;
}

public void setButton1_2(JButton button1_2) {
    this.button1_2 = button1_2;
}

public void setButton1(JButton button1) {
    this.button1 = button1;
}

public void setButton2(JButton button2) {
    this.button2 = button2;
}

public void setButton3(JButton button3) {
    this.button3 = button3;
}

public JLabel getLblNuevoIdrea() {
    return lblNuevoIdrea;
}

public void setLblNuevoIdrea(JLabel lblNuevoIdrea) {
    this.lblNuevoIdrea = lblNuevoIdrea;
}

public JTextField getTextField_2() {
    return textField_2;
}

```

```

public void setTextField_2(JTextField textField_2) {
    this.textField_2 = textField_2;
}

/**
 * constructor del objeto editarAreas
 */
public editarAreas() {
    //////////////////// MENÚ Y FONDO/////////////////////////////77
    Toolkit toolkit = Toolkit.getDefaultToolkit();
    Dimension screenSize = toolkit.getScreenSize();

    frame = new JFrame("Consultar áreas");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1350, 820);

    panel = new JPanel() {
        @Override
        protected void paintComponent(Graphics g) {

            // Carga y dibuja la imagen de fondo
            ImageIcon imageIcon = new ImageIcon("Imagenes/modArea.png");
            Image image = imageIcon.getImage();
            g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
            setOpaque(false);
            setVisible(true);
            super.paintComponent(g);
        }
    };
}

// Cambia el icono de la ventana de Java
ImageIcon icon = new ImageIcon("Imagenes/7.png");
frame.setIconImage(icon.getImage());

ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

button1 = new JButton("");
button1.setBorderPainted(false);
button1.setBounds(115, 677, 90, 78);
button1.setIcon(new ImageIcon(resizedImage));
button1.setContentAreaFilled(false);
button1.setFocusPainted(false);
button1.setOpaque(false);

button2 = new JButton("");
button2.setBorderPainted(false);
button2.setBounds(600, 677, 90, 78);
button2.setIcon(new ImageIcon(resizedAdd));
button2.setContentAreaFilled(false);
button2.setFocusPainted(false);
button2.setOpaque(false);

button3 = new JButton("");
button3.setBorderPainted(false);
button3.setBounds(1085, 677, 90, 78);
button3.setIcon(new ImageIcon(resizedConsultar));
button3.setContentAreaFilled(false);
button3.setFocusPainted(false);
button3.setOpaque(false);

ImageIcon imagenModificarIcon = new ImageIcon("Imagenes/lapiz.png");
Image     resizedModImage      =     imagenModificarIcon.getImage().getScaledInstance(70,      70,
Image.SCALE_SMOOTH);

button1_1 = new JButton("");
button1_1.setBorderPainted(false);
button1_1.setOpaque(false);
button1_1.setFocusPainted(false);
button1_1.setIcon(new ImageIcon(resizedModImage));

```

```

        button1_1.setContentAreaFilled(false);
        button1_1.setBounds(1072, 523, 99, 99);

        frame.setContentPane(panel);
        panel.setLayout(null);

        ImageIcon imagenEliminarIcon = new ImageIcon("Imagenes/cancelar.png");
        Image     resizedEliminarImage      =     imagenEliminarIcon.getImage().getScaledInstance(70,      70,
Image.SCALE_SMOOTH);

        button1_2 = new JButton("");
        button1_2.setOpaque(false);
        button1_2.setFocusPainted(false);
        button1_2.setContentAreaFilled(false);
        button1_2.setIcon(new ImageIcon(resizedEliminarImage));
        button1_2.setBorderPainted(false);
        button1_2.setBounds(1181, 523, 99, 99);

        panel.add(button1);
        panel.add(button1_1);
        panel.add(button1_2);
        panel.add(button2);
        panel.add(button3);

        JLabel lblNewLabel = new JLabel("ID Área:");
        lblNewLabel.setFont(new Font("SansSerif", Font.PLAIN, 30));
        lblNewLabel.setBounds(59, 431, 174, 62);
        panel.add(lblNewLabel);

        JLabel lblNExpediente = new JLabel("Descripción:");
        lblNExpediente.setFont(new Font("SansSerif", Font.PLAIN, 30));
        lblNExpediente.setBounds(59, 523, 229, 62);
        panel.add(lblNExpediente);

        textField = new JTextField();
        textField.setBackground(new Color(211, 233, 250));
        textField.setBounds(192, 445, 319, 39);
        panel.add(textField);
        textField.setColumns(10);

        textField_1 = new JTextField();
        textField_1.setColumns(10);
        textField_1.setBackground(new Color(211, 233, 250));
        textField_1.setBounds(314, 523, 727, 108);
        panel.add(textField_1);

        lblNuevoIdrea = new JLabel("Nuevo ID Área:");
        lblNuevoIdrea.setFont(new Font("SansSerif", Font.PLAIN, 30));
        lblNuevoIdrea.setBounds(575, 431, 241, 62);
        panel.add(lblNuevoIdrea);

        textField_2 = new JTextField();
        textField_2.setColumns(10);
        textField_2.setBackground(new Color(211, 233, 250));
        textField_2.setBounds(814, 445, 319, 39);
        panel.add(textField_2);

        frame.setVisible(false);

    }

    /**
     * metodo que hace las vista visible
     */
    public void hacerVisible() {
        frame.setVisible(true);
    }

    /**
     * metodo que se deshace de la vista
     */
    public void dispose() {
        frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
        frame.setVisible(false);
    }
}

```

```

}

/**
 * getter del boton 1
 *
 * @return button1 el boton 1
 */
public JButton getButton1() {
    return button1;
}

/**
 * getter del boton 2
 *
 * @return button2 el boton 2
 */
public JButton getButton2() {
    return button2;
}

/**
 * getter del boton 3
 *
 * @return button3 el boton 3
 */
public JButton getButton3() {
    return button3;
}

/**
 * cambia la vista a altas
 *
 * @param altas vista altas
 */
public void setAltas(altas altas) {
    // boton que va a altas
    setControlador2(altas);
}

/**
 * cambia la vista al home
 *
 * @param home vista home
 */
public void setHome(home home) {
    // boton que vuelve a home
    setControlador1(home);
}

/**
 * cambia la vista a consultas
 *
 * @param consultas vista consultas
 */
public void setConsultas(consultas consultas) {
    // boton que vuelve a home
    setControlador3(consultas);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a editar altas
 *
 * @param x vista altas
 */
public void setControlador2(altas x) {
    consultaAreasAñadir escuchador = new consultaAreasAñadir();
    escuchador.setVentanaUsuario(this, x);
    button2.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a editar home
 *
 * @param x vista home

```

```

        */
    public void setControlador1(home x) {
        consultaAreasHome escuchador = new consultaAreasHome();
        escuchador.setVentanaUsuario(this, x);
        button1.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a editar
     * consultas
     *
     * @param x vista consultas
     */
    public void setControlador3(consultas x) {
        consultasAreasConsultas escuchador = new consultasAreasConsultas();
        escuchador.setVentanaUsuario(this, x);
        button3.addActionListener(escuchador);

        final AccesoBBDD accesoBBDD = new AccesoBBDD(this);
        button1_1.addActionListener(new ActionListener() {
            /**
             * metodo que edita los datos de areas de la base de datos
             *
             * @param e
             */
            public void actionPerformed(ActionEvent e) {
                // Insertar los datos en la tabla ALUMNOS
                accesoBBDD.editarDatosAreas();
            }
        });

        button1_2.addActionListener(new ActionListener() {
            /**
             * metodo que borra los datos de las ares de la bbdd
             */
            public void actionPerformed(ActionEvent e) {
                // Insertar los datos en la tabla ALUMNOS
                accesoBBDD.borrarDatosAreas();
            }
        });
    }
}

```

EditarPI

```

package mvc.vistas;

import javax.swing.*;

import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.text.ParseException;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;

import BBDD.AccesoBBDD;

import mvc.control.consultaAreasAñadir;
import mvc.control.consultaAreasHome;
import mvc.control.consultasAlumnoAñadir;
import mvc.control.consultasAlumnoConsultas;
import mvc.control.consultasAlumnosB;
import mvc.control.consultasAreasB;
import mvc.control.consultasAreasConsultas;
import mvc.control.editarPIAñadir;
import mvc.control.editarPIConsultas;
import mvc.control.editarPIHome;

/**
 * clase que contiene la vista de editar pi y sus correspondientes accesos a la

```

```

* bbdd
*
* @author alexjun09
*
*/
public class editarPI {

    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2;
    private JButton button3;
    private JButton buttonadd1;
    private JButton buttonadd2;
    private JButton buttonadd3;
    private JLabel lblNewLabel_1;
    private JLabel lblNewLabel_2;
    private JLabel lblNewLabel_3;
    private JTextField textField;
    private JButton button1_1;
    private JButton button1_2;
    private JTextField textField_3;
    private JTextField textField_4;
    private JTextField textField_5;
    private JTextField textField_6;
    private JTextField textField_7;
    private JTextField textField_8;
    private JTextField textField_9;
    private JTextField textField_10;

    // setters y getters habituales
    public JFrame getFrame() {
        return frame;
    }

    public void setFrame(JFrame frame) {
        this.frame = frame;
    }

    public JPanel getPanel() {
        return panel;
    }

    public void setPanel(JPanel panel) {
        this.panel = panel;
    }

    public JButton getButtonadd1() {
        return buttonadd1;
    }

    public void setButtonadd1(JButton buttonadd1) {
        this.buttonadd1 = buttonadd1;
    }

    public JButton getButtonadd2() {
        return buttonadd2;
    }

    public void setButtonadd2(JButton buttonadd2) {
        this.buttonadd2 = buttonadd2;
    }

    public JButton getButtonadd3() {
        return buttonadd3;
    }

    public void setButtonadd3(JButton buttonadd3) {
        this.buttonadd3 = buttonadd3;
    }

    public JLabel getLblNewLabel_1() {
        return lblNewLabel_1;
    }
}

```

```

public void setLblNewLabel_1(JLabel lblNewLabel_1) {
    this.lblNewLabel_1 = lblNewLabel_1;
}

public JLabel getLblNewLabel_2() {
    return lblNewLabel_2;
}

public void setLblNewLabel_2(JLabel lblNewLabel_2) {
    this.lblNewLabel_2 = lblNewLabel_2;
}

public JLabel getLblNewLabel_3() {
    return lblNewLabel_3;
}

public void setLblNewLabel_3(JLabel lblNewLabel_3) {
    this.lblNewLabel_3 = lblNewLabel_3;
}

public JTextField getTextField() {
    return textField;
}

public void setTextField(JTextField textField) {
    this.textField = textField;
}

public JButton getButton1_1() {
    return button1_1;
}

public void setButton1_1(JButton button1_1) {
    this.button1_1 = button1_1;
}

public JButton getButton1_2() {
    return button1_2;
}

public void setButton1_2(JButton button1_2) {
    this.button1_2 = button1_2;
}

public JTextField getTextField_3() {
    return textField_3;
}

public void setTextField_3(JTextField textField_3) {
    this.textField_3 = textField_3;
}

public JTextField getTextField_4() {
    return textField_4;
}

public void setTextField_4(JTextField textField_4) {
    this.textField_4 = textField_4;
}

public JTextField getTextField_5() {
    return textField_5;
}

public void setTextField_5(JTextField textField_5) {
    this.textField_5 = textField_5;
}

public JTextField getTextField_6() {
    return textField_6;
}

public void setTextField_6(JTextField textField_6) {
}

```

```

        this.textField_6 = textField_6;
    }

    public JTextField getTextField_7() {
        return textField_7;
    }

    public void setTextField_7(JTextField textField_7) {
        this.textField_7 = textField_7;
    }

    public JTextField getTextField_8() {
        return textField_8;
    }

    public void setTextField_8(JTextField textField_8) {
        this.textField_8 = textField_8;
    }

    public JTextField getTextField_9() {
        return textField_9;
    }

    public void setTextField_9(JTextField textField_9) {
        this.textField_9 = textField_9;
    }

    public JTextField getTextField_10() {
        return textField_10;
    }

    public void setTextField_10(JTextField textField_10) {
        this.textField_10 = textField_10;
    }

    public void setButton1(JButton button1) {
        this.button1 = button1;
    }

    public void setButton2(JButton button2) {
        this.button2 = button2;
    }

    public void setButton3(JButton button3) {
        this.button3 = button3;
    }

    /**
     * constructor del objeto editarPI
     */
    public editarPI() {

        ///////////////// MENÚ Y FONDO///////////////
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension screenSize = toolkit.getScreenSize();

        frame = new JFrame("Consultar áreas");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(1350, 820);

        panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {

                // Carga y dibuja la imagen de fondo
                ImageIcon imageIcon = new ImageIcon("Imagenes/modPI.png");
                Image image = imageIcon.getImage();
                g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
                setVisible(true);
                super.paintComponent(g);
            }
        };
    };
}

```

```

// Cambia el icono de la ventana de Java
ImageIcon icon = new ImageIcon("Imagenes/7.png");
frame.setIconImage(icon.getImage());

ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

button1 = new JButton("");
button1.setBorderPainted(false);
button1.setBounds(115, 677, 90, 78);
button1.setIcon(new ImageIcon(resizedImage));
button1.setContentAreaFilled(false);
button1.setFocusPainted(false);
button1.setOpaque(false);

button2 = new JButton("");
button2.setBorderPainted(false);
button2.setBounds(600, 677, 90, 78);
button2.setIcon(new ImageIcon(resizedAdd));
button2.setContentAreaFilled(false);
button2.setFocusPainted(false);
button2.setOpaque(false);

button3 = new JButton("");
button3.setBorderPainted(false);
button3.setBounds(1085, 677, 90, 78);
button3.setIcon(new ImageIcon(resizedConsultar));
button3.setContentAreaFilled(false);
button3.setFocusPainted(false);
button3.setOpaque(false);

ImageIcon imagenModificarIcon = new ImageIcon("Imagenes/lapiz.png");
Image resizedModImage = imagenModificarIcon.getImage().getScaledInstance(70, 70,
Image.SCALE_SMOOTH);

button1_1 = new JButton("");
button1_1.setBorderPainted(false);
button1_1.setOpaque(false);
button1_1.setFocusPainted(false);
button1_1.setIcon(new ImageIcon(resizedModImage));
button1_1.setContentAreaFilled(false);
button1_1.setBounds(1104, 523, 75, 71);

frame.setContentPane(panel);
panel.setLayout(null);

ImageIcon imagenEliminarIcon = new ImageIcon("Imagenes/cancelar.png");
Image resizedEliminarImage = imagenEliminarIcon.getImage().getScaledInstance(70, 70,
Image.SCALE_SMOOTH);

button1_2 = new JButton("");
button1_2.setOpaque(false);
button1_2.setFocusPainted(false);
button1_2.setContentAreaFilled(false);
button1_2.setIcon(new ImageIcon(resizedEliminarImage));
button1_2.setBorderPainted(false);
button1_2.setBounds(1200, 523, 75, 71);

panel.add(button1);
panel.add(button1_1);
panel.add(button1_2);
panel.add(button2);
panel.add(button3);

JLabel lblNewLabel = new JLabel("ID Proyecto:");
lblNewLabel.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNewLabel.setBounds(57, 250, 174, 62);
panel.add(lblNewLabel);

textField = new JTextField();

```

```

textField.setBackground(new Color(211, 233, 250));
textField.setBounds(230, 270, 283, 39);
panel.add(textField);
textField.setColumns(10);

JLabel lblNombre = new JLabel("Nombre");
lblNombre.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNombre.setBounds(57, 394, 174, 62);
panel.add(lblNombre);

JLabel lblNotaObtenida = new JLabel("Nota :");
lblNotaObtenida.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblNotaObtenida.setBounds(57, 487, 225, 62);
panel.add(lblNotaObtenida);

JLabel lblAo = new JLabel("Año:");
lblAo.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblAo.setBounds(57, 575, 174, 62);
panel.add(lblAo);

textField_3 = new JTextField();
textField_3.setColumns(10);
textField_3.setBackground(new Color(211, 233, 250));
textField_3.setBounds(230, 414, 283, 39);
panel.add(textField_3);

textField_4 = new JTextField();
textField_4.setColumns(10);
textField_4.setBackground(new Color(211, 233, 250));
textField_4.setBounds(230, 507, 283, 39);
panel.add(textField_4);

textField_5 = new JTextField();
textField_5.setColumns(10);
textField_5.setBackground(new Color(211, 233, 250));
textField_5.setBounds(230, 583, 243, 39);
panel.add(textField_5);

JLabel lblGrupo = new JLabel("Grupo:");
lblGrupo.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblGrupo.setBounds(650, 314, 174, 62);
panel.add(lblGrupo);

JLabel lblUrl = new JLabel("URL:");
lblUrl.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblUrl.setBounds(650, 394, 174, 62);
panel.add(lblUrl);

JLabel lblCurso = new JLabel("Curso:");
lblCurso.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblCurso.setBounds(650, 487, 174, 62);
panel.add(lblCurso);

JLabel lblIdArea = new JLabel("ID Area:");
lblIdArea.setFont(new Font("SansSerif", Font.PLAIN, 30));
lblIdArea.setBounds(650, 575, 158, 62);
panel.add(lblIdArea);

textField_6 = new JTextField();
textField_6.setColumns(10);
textField_6.setBackground(new Color(211, 233, 250));
textField_6.setBounds(780, 334, 298, 39);
panel.add(textField_6);

textField_7 = new JTextField();
textField_7.setColumns(10);
textField_7.setBackground(new Color(211, 233, 250));
textField_7.setBounds(780, 414, 298, 39);
panel.add(textField_7);

textField_8 = new JTextField();
textField_8.setColumns(10);
textField_8.setBackground(new Color(211, 233, 250));
textField_8.setBounds(780, 507, 298, 39);

```

```

        panel.add(textField_8);

        textField_9 = new JTextField();
        textField_9.setColumns(10);
        textField_9.setBackground(new Color(211, 233, 250));
        textField_9.setBounds(780, 583, 298, 39);
        panel.add(textField_9);

        JLabel lblNuevoIdProyecto = new JLabel("Nuevo ID Proyecto:");
        lblNuevoIdProyecto.setFont(new Font("SansSerif", Font.PLAIN, 30));
        lblNuevoIdProyecto.setBounds(57, 322, 283, 46);
        panel.add(lblNuevoIdProyecto);

        textField_10 = new JTextField();
        textField_10.setColumns(10);
        textField_10.setBackground(new Color(211, 233, 250));
        textField_10.setBounds(333, 336, 243, 39);
        panel.add(textField_10);

        frame.setVisible(false);

    }

    /**
     * metodo que hace visible la vista
     */
    public void hacerVisible() {
        frame.setVisible(true);
    }

    /**
     * metodo que se desace de la vista
     */
    public void dispose() {
        frame.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
        frame.setVisible(false);
    }

    /**
     * getter del boton 1
     *
     * @return button1 el boton 1
     */
    public JButton getButton1() {
        return button1;
    }

    /**
     * getter del boton 2
     *
     * @return button2 el boton 2
     */
    public JButton getButton2() {
        return button2;
    }

    /**
     * getter del boton 2
     *
     * @return button2 el boton 2
     */
    public JButton getButton3() {
        return button3;
    }

    /**
     * metodo que cambia de vista al home
     *
     * @param x vista home
     */
    public void setHome(home x) {
        setControlador(x);
    }
}

```

```

/**
 * metodo que cambia la vista a altas
 *
 * @param a vista altas
 */
public void setAltas(altas a) {
    // Boton que va a altas
    setControlador2(a);
}

/**
 * metodo que cambia la vista a consultas
 *
 * @param c vista consultas
 */
public void setConsultas(consultas c) {
    // Boton que va a consultas
    setControlador1(c);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a editar home
 *
 * @param x vista home
 */
public void setControlador(home x) {
    editarPIHome escuchador = new editarPIHome();
    escuchador.setVentanaUsuario(x, this);
    button1.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a editar
 * consultas
 *
 * @param x vista consultas
 */
public void setControlador1(consultas x) {
    editarPIConsultas escuchador = new editarPIConsultas();
    escuchador.setVentanaUsuario(x, this);
    button2.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a editar altas
 *
 * @param x vista altas
 */
public void setControlador2(altas x) {
    editarPIAñadir escuchador2 = new editarPIAñadir();
    escuchador2.setVentanaUsuario(x, this);
    button3.addActionListener(escuchador2);

    final AccesoBBDD accesoBBDD = new AccesoBBDD(this);
    button1_1.addActionListener(new ActionListener() {
        /**
         * metodo que accede a la base de datos y edita los datos de los proyectos
         * integradores
         */
        public void actionPerformed(ActionEvent e) {
            // Insertar los datos en la tabla ALUMNOS
            try {
                accesoBBDD.editarDatosPI();
            } catch (ParseException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
        }
    });
}

button1_2.addActionListener(new ActionListener() {
    /**
     * metodo que accede a la base de datos y borra los datos del proyecto
     * integrador
     */
}

```

```
        *
        * @param e
        */
    public void actionPerformed(ActionEvent e) {
        // Insertar los datos en la tabla ALUMNOS
        accesoBBDD.borrarDatosPI();
    }
});
```

Home

```
package mvc.vistas;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.awt.event.ActionEvent;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;

import mvc.control.homeConsultas;
import mvc.control.homeAñadir;

/**
 * clase que contiene la vista del home y sus correspondientes accesos
 *
 * @author alexjun09
 *
 */
public class home {
    private JFrame frame;
    private JPanel panel;
    private JButton button1;
    private JButton button2_1;
    private JButton button3_1;
    private altas a;
    private consultas c;

    /**
     * constructor del objeto home
     */
    public home() {
        // Creacion de clases

        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension screenSize = toolkit.getScreenSize();

        frame = new JFrame("Home Page");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(1350, 820);

        panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {

                // Carga y dibuja la imagen de fondo
                ImageIcon imageIcon = new ImageIcon("Imagenes/Sara.png");
                Image image = imageIcon.getImage();
                g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
                setVisible(true);
                super.paintComponent(g);
            }
        };
    }
}
```

```

};

// Cambia el icono de la ventana de Java
ImageIcon icon = new ImageIcon("Imagenes/7.png");
frame.setIconImage(icon.getImage());

// Iconos de los botones
ImageIcon imagenHome = new ImageIcon("Imagenes/home.png");
Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenAdd = new ImageIcon("Imagenes/mas.png");
Image resizedAdd = imagenAdd.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
ImageIcon imagenConsultar = new ImageIcon("Imagenes/buscar.png");
Image resizedConsultar = imagenConsultar.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);

button1 = new JButton("");
button1.setBorderPainted(false);
button1.setBounds(115, 677, 90, 78);
button1.setIcon(new ImageIcon(resizedImage));
button1.setContentAreaFilled(false);
button1.setFocusPainted(false);
button1.setOpaque(false);

button2_1 = new JButton("");
button2_1.setBorderPainted(false);
button2_1.setBounds(600, 677, 90, 78);
button2_1.setIcon(new ImageIcon(resizedAdd));
button2_1.setContentAreaFilled(false);
button2_1.setFocusPainted(false);
button2_1.setOpaque(false);

button3_1 = new JButton("");
button3_1.setBorderPainted(false);
button3_1.setBounds(1085, 677, 90, 78);
button3_1.setIcon(new ImageIcon(resizedConsultar));
button3_1.setContentAreaFilled(false);
button3_1.setFocusPainted(false);
button3_1.setOpaque(false);

frame.setContentPane(panel);
panel.setLayout(null);
panel.add(button1);
panel.add(button2_1);
panel.add(button3_1);

}

/**
 * metodo que hace la vista visible
 */
public void hacerVisible() {
    frame.setVisible(true);
}

/**
 * metodo que se deshace de la vista
 */
public void dispose() {
    frame.setVisible(false);
}

/**
 * getter del boton 1
 *
 * @return button1 el boton 1
 */
public JButton getButton1() {
    return button1;
}

/**
 * getter del boton 2
 *
 * @return button2 el boton 2
 */

```

```

public JButton getButton2() {
    return button2_1;
}

/**
 * getter del boton 3
 *
 * @return button3 el boton 3
 */
public JButton getButton3() {
    return button3_1;
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a editar altas
 *
 * @param x vista de altas
 */
public void setControlador1(alertas x) {
    homeAñadir escuchador = new homeAñadir();
    escuchador.setVentanaUsuario(this, x);
    button2_1.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a editar
 * consultas
 *
 * @param x vista de consultas
 */
public void setControlador2(consultas x) {
    homeConsultas escuchador2 = new homeConsultas();
    escuchador2.setVentanaUsuario(this, x);
    button3_1.addActionListener(escuchador2);
}

/**
 * metodo que cambia de vista a altas
 *
 * @param altas vista de altas
 */
public void setAltas(alertas altas) {
    a = altas;
    // Boton que va a altas
    setControlador1(a);
}

/**
 * metodo que cambia de vista a consultas
 *
 * @param con vista de consultas
 */
public void setConsultas(consultas con) {
    c = con;
    // Boton que va a consultas
    setControlador2(c);
}
}

```

VistaLogin

```

package mvc.vistas;

import java.awt.*;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

import java.util.ArrayList;
import java.util.List;
import javax.swing.*;

import mvc.control.EscuchadorLogIn;
import mvc.control.EscuchadorSignUp;
import mvc.control.loginBack;
import mvc.vistas.*;

/**
 * clase que contiene la vista del login y sus correspondientes accesos
 *
 * @author alexjun09
 *
 */
public class VistaLogin extends JFrame {
    private JTextField txtUser;
    private JPasswordField txtPass;
    private JLabel alert;
    private JButton btn;
    private JButton btn_atras;
    private home home;
    private VistaPrincipal VistaPrincipal;
    private static JButton botonFondo;
    private static List<ImageIcon> frames;
    private static int currentFrameIndex;
    private static final int ICON_WIDTH = 167;
    private static final int ICON_HEIGHT = 79;
    EscuchadorLogIn comprobar;

    /**
     * constructor del objeto vistalogin
     */
    public VistaLogin() {
        inicializar();
    }

    /**
     * metodo con toda la funcionalidad de la vista
     */
    public void inicializar() {
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                // Carga y dibuja la imagen de fondo
                ImageIcon imageIcon = new ImageIcon("Imagenes/FondoLogin.png");
                Image image = imageIcon.getImage();
                g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
                super.paintComponent(g);
            }
        };
        setContentPane(panel);
        setSize(1350, 820);
        getContentPane().setLayout(null);

        // Cambia el icono de la ventana de Java
        ImageIcon icon = new ImageIcon("Imagenes/7.png");
        setIconImage(icon.getImage());

        JLabel labelUser = new JLabel("Usuario");
        labelUser.setFont(new Font("Tw Cen MT Condensed", Font.PLAIN, 50));
        labelUser.setBounds(517, 374, 153, 46);
        getContentPane().add(labelUser);

        JLabel labelPass = new JLabel("Contraseña");
        labelPass.setFont(new Font("Tw Cen MT Condensed", Font.PLAIN, 50));
        labelPass.setBounds(474, 482, 200, 62);
        getContentPane().add(labelPass);

        txtUser = new JTextField();
        txtUser.setFont(new Font("Tahoma", Font.PLAIN, 30));

```

```

txtUser.setBounds(659, 375, 252, 45);
getContentPane().add(txtUser);
txtUser.setColumns(10);

txtPass = new JPasswordField();
txtPass.setFont(new Font("Tahoma", Font.PLAIN, 30));
txtPass.setBounds(659, 494, 252, 45);
getContentPane().add(txtPass);

//Botón atrás
btn_atras = new JButton("");
btn_atras.setBorderPainted(false);
btn_atras.setBounds(115, 677, 90, 78);
btn_atras.setContentAreaFilled(false);
btn_atras.setFocusPainted(false);
btn_atras.setOpaque(false);
ImageIcon imagenHome = new ImageIcon("Imagenes/back.png");
Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
btn_atras.setIcon(new ImageIcon(resizedImage));
getContentPane().add(btn_atras);

// BOTÓN PRINCIPAL
btn = new JButton("");
btn.setBounds(601, 613, 167, 79);
getContentPane().add(btn);
// Establecer la opacidad del botón
btn.setOpaque(false);
btn.setContentAreaFilled(false);
btn.setBorderPainted(false);

// Establecer el fondo del botón como transparente
btn.setBackground(new Color(0, 0, 0));

// BOTÓN FONDO
frames = new ArrayList<>();

// Carga las imágenes del GIF en una lista
for (int i = 1; i <= 2; i++) {
    String imagePath = "ImagenesGifL/imagen (" + i + ").png";
    ImageIcon frame = new ImageIcon(imagePath);
    ImageIcon scaledFrame = scaleImageIcon(frame, ICON_WIDTH, ICON_HEIGHT);
    frames.add(scaledFrame);
}

// Configura el primer frame como ícono inicial del botón
currentFrameIndex = 0;

// Inicia un temporizador para cambiar automáticamente los frames
int delay = 3000; // Intervalo en milisegundos entre cada cambio de frame
Timer timer = new Timer(delay, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        botonFondo.doClick(); // Simula un clic en el botón para cambiar el frame
    }
});
timer.start();

botonFondo = new JButton();
botonFondo.setBounds(601, 613, 170, 70);
botonFondo.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Cambia la imagen del botón al siguiente frame
        currentFrameIndex = (currentFrameIndex + 1) % frames.size();
        botonFondo.setIcon(frames.get(currentFrameIndex));
    }
});
botonFondo.setIcon(frames.get(currentFrameIndex));

botonFondo.setOpaque(false);
botonFondo.setContentAreaFilled(false);
botonFondo.setBorderPainted(false);
botonFondo.setBackground(new Color(0, 0, 0));

```

```

        getContentPane().add(botonFondo);

        alert = new JLabel("");
        alert.setBounds(486, 450, 337, 16);
        getContentPane().add(alert);

    }

    /**
     * metodo que hace la clase visible
     */
    public void hacerVisible() {
        setVisible(true);
    }

    /**
     * metodo que hace la clase invisible
     */
    public void hacerInvisible() {
        setVisible(false);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá al home
     *
     * @param x vista home
     */
    public void setControlador(home x) {
        EscuchadorLogIn escuchador = new EscuchadorLogIn();
        escuchador.setVistaPrincipal(this, x);
        btn.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a la vista
     * principal
     *
     * @param x vista vistaprincipal
     */
    public void setControlador2(VistaPrincipal vp) {
        loginBack escuchador = new loginBack();
        escuchador.setVistaPrincipal(this, vp);
        btn_atras.addActionListener(escuchador);
    }

    /**
     * metodo que cambia de vista al home
     *
     * @param x vista home
     */
    public void setHome(home x) {
        home = x;
        setControlador(x);
    }

    /**
     * metodo que cambia la vista al la vista principal
     *
     * @param vp vista vistaprincipal
     */
    public void setVistaPrincipal(VistaPrincipal vp) {
        VistaPrincipal = vp;
        setControlador2(vp);
    }

    // setters y getters habituales
    public JLabel getAlert() {
        return alert;
    }

    public void setAlert(JLabel alert) {
        this.alert = alert;
    }

```

```

public JTextField getTxtUser() {
    return txtUser;
}

public void setTxtUser(JTextField txtUser) {
    this.txtUser = txtUser;
}

public JPasswordField getTxtPass() {
    return txtPass;
}

public void setTxtPass(JPasswordField txtPass) {
    this.txtPass = txtPass;
}

public JButton getBtn_atras() {
    return btn_atras;
}

private static ImageIcon scaleImageIcon(ImageIcon icon, int width, int height) {
    Image image = icon.getImage();
    Image scaledImage = image.getScaledInstance(width, height, Image.SCALE_SMOOTH);
    return new ImageIcon(scaledImage);
}
}

```

VistaPrincipal

```

package mvc.vistas;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.List;

import javax.swing.*;

import mvc.control.*;

/**
 * clase que contiene la vista de vistaPrincinal y sus correspondientes accesos
 *
 * @author alexjun09
 *
 */
public class VistaPrincipal extends JFrame {

    private JButton botonLogIn;
    private JButton botonSignUp;
    VistaLogin login;
    VistaSignUp sign;
    private static JButton botonFondoLogIn;
    private static JButton botonFondoSignUp;
    private static List<ImageIcon> framesLogIn = new ArrayList<>();
    private static List<ImageIcon> framesSignUp = new ArrayList<>();
    private static int currentFrameIndexLogIn;
    private static int currentFrameIndexSignUp;
    private static final int ICON_WIDTH = 235;
    private static final int ICON_HEIGHT = 164;

    /**
     * constructor del objeto vistaprincipal
     */
    public VistaPrincipal() {
        inicializar();
    }
}

```

```

}

/**
 * metodo con toda la funcionalidad de la vista
 */
public void inicializar() {
    JPanel panel = new JPanel() {
        @Override
        protected void paintComponent(Graphics g) {
            // Carga y dibuja la imagen de fondo
            ImageIcon imageIcon = new ImageIcon("Imagenes/FondoVistaPrincipal.png");
            Image image = imageIcon.getImage();
            g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
            setOpaque(false);
            super.paintComponent(g);
        }
    };

    setContentPane(panel);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(1350, 820);
    getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("Bienvenido");
    lblNewLabel.setForeground(new Color(255, 255, 255));
    lblNewLabel.setFont(new Font("Agency FB", Font.BOLD | Font.ITALIC, 99));
    lblNewLabel.setBounds(468, 75, 402, 99);
    getContentPane().add(lblNewLabel);

    // Cambia el icono de la ventana de Java
    ImageIcon icon = new ImageIcon("Imagenes/7.png");
    setIconImage(icon.getImage());

    //BOTÓN LOGIN FUNCIONAL
    botonLogIn = new JButton("");
    botonLogIn.setBounds(364, 600, 235, 164);
    getContentPane().add(botonLogIn);
    // Establecer la opacidad del botón
    botonLogIn.setOpaque(false);
    botonLogIn.setContentAreaFilled(false);
    botonLogIn.setBorderPainted(false);
    // Establecer el fondo del botón como transparente
    botonLogIn.setBackground(new Color(0, 0, 0, 0));

    //BOTÓN DISEÑO LOGIN
    // Carga las imágenes del GIF en una lista
    for (int i = 1; i <= 2; i++) {
        String imagePath = "ImagenesGifL/imagen (" + i + ").png";
        ImageIcon frame = new ImageIcon(imagePath);
        ImageIcon scaledFrame = scaleImageIcon(frame, ICON_WIDTH, ICON_HEIGHT);
        framesLogIn.add(scaledFrame);
    }

    // Configura el primer frame como ícono inicial del botón
    currentFrameIndexLogIn = 0;

    // Inicia un temporizador para cambiar automáticamente los frames
    int delay = 3000; // Intervalo en milisegundos entre cada cambio de frame
    Timer timer = new Timer(delay, new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            botonFondoLogIn.doClick(); // Simula un clic en el botón para cambiar el frame
        }
    });
    timer.start();

    botonFondoLogIn = new JButton();
    botonFondoLogIn.setBounds(364, 600, 235, 164);
    botonFondoLogIn.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Cambia la imagen del botón al siguiente frame
            currentFrameIndexLogIn = (currentFrameIndexLogIn + 1) % framesLogIn.size();
            botonFondoLogIn.setIcon(framesLogIn.get(currentFrameIndexLogIn));
        }
    });
}

```

```

    });
    botonFondoLogIn.setIcon(framesLogIn.get(currentFrameIndexLogIn));

    botonFondoLogIn.setOpaque(false);
    botonFondoLogIn.setContentAreaFilled(false);
    botonFondoLogIn.setBorderPainted(false);
    botonFondoLogIn.setBackground(new Color(0, 0, 0, 0));

    getContentPane().add(botonFondoLogIn);

//BOTÓN SIGNUP FUNCIONAL
    botonSignUp = new JButton("");
    botonSignUp.setBounds(686, 600, 235, 164);
    getContentPane().add(botonSignUp);
    // Establecer la opacidad del botón
    botonSignUp.setOpaque(false);
    botonSignUp.setContentAreaFilled(false);
    botonSignUp.setBorderPainted(false);
    // Establecer el fondo del botón como transparente
    botonSignUp.setBackground(new Color(0, 0, 0, 0));

//BOTÓN DISEÑO SIGNUP
    // Carga las imágenes del GIF en una lista
    for (int i = 1; i <= 2; i++) {
        String imagePath = "ImagenesGifs/imagen (" + i + ").png";
        ImageIcon frame = new ImageIcon(imagePath);
        ImageIcon scaledFrame = scaleImageIcon(frame, ICON_WIDTH, ICON_HEIGHT);
        framesSignUp.add(scaledFrame);
    }

    // Configura el primer frame como ícono inicial del botón
    currentFrameIndexSignUp = 0;

    // Inicia un temporizador para cambiar automáticamente los frames
    int delayS = 3000; // Intervalo en milisegundos entre cada cambio de frame
    Timer timerS = new Timer(delayS, new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            botonFondoSignUp.doClick(); // Simula un clic en el botón para cambiar el frame
        }
    });
    timerS.start();

    botonFondoSignUp = new JButton();
    botonFondoSignUp.setBounds(686, 600, 235, 164);
    botonFondoSignUp.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Cambia la imagen del botón al siguiente frame
            currentFrameIndexSignUp = (currentFrameIndexSignUp + 1) % framesSignUp.size();
            botonFondoSignUp.setIcon(framesSignUp.get(currentFrameIndexSignUp));
        }
    });
    botonFondoSignUp.setIcon(framesSignUp.get(currentFrameIndexSignUp));

    botonFondoSignUp.setOpaque(false);
    botonFondoSignUp.setContentAreaFilled(false);
    botonFondoSignUp.setBorderPainted(false);
    botonFondoSignUp.setBackground(new Color(0, 0, 0, 0));

    getContentPane().add(botonFondoSignUp);
}

/**
 * cambia la vista a vistalogin
 *
 * @param x vista vistalogin
 */
public void setVistaLogin(VistaLogin x) {
    login = x;
    setControlador1(x);
}

```

```

    /**
     * cambia la vista a vist_signup
     *
     * @param x vista vist_signup
     */
    public void setVistaSignUp(VistaSignUp x) {
        sign = x;
        setControlador2(x);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a vist_login
     *
     * @param x vista vist_login
     */
    public void setControlador1(VistaLogin x) {
        EscuchadorPrincipalLogIn escuchador = new EscuchadorPrincipalLogIn();
        escuchador.setEscuchadorPrincipalLogIn(this, x);
        botonLogIn.addActionListener(escuchador);
    }

    /**
     * metodo que creará una nuevo controlador (boton) que redigirá a vist_signup
     *
     * @param x vista vist_signup
     */
    public void setControlador2(VistaSignUp x) {
        EscuchadorPrincipalSignUp escuchador2 = new EscuchadorPrincipalSignUp();
        escuchador2.setEscuchadorPrincipalSignUp(this, x);
        botonSignUp.addActionListener(escuchador2);
    }

    /**
     * metodo que hace la vista visible
     */
    public void hacerVisible() {
        setVisible(true);
    }

    /**
     * metodo que hace la vista invisible
     */
    public void hacerInvisible() {
        setVisible(false);
    }

    private static ImageIcon scaleImageIcon(ImageIcon icon, int width, int height) {
        Image image = icon.getImage();
        Image scaledImage = image.getScaledInstance(width, height, Image.SCALE_SMOOTH);
        return new ImageIcon(scaledImage);
    }
}

```

VistaSignUp

```

package mvc.vistas;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.List;

import javax.swing.*;

import mvc.control.*;
import java.awt.Font;

/**
 * clase que contiene las vistas y funcionalidad del signup
 *
 * @author alexjun09

```

```

*
*/
public class VistaSignUp extends JFrame {
    private JTextField txtUser;
    private JPasswordField txtPass;
    private JPasswordField txtRepetirPass;
    private JLabel alert;
    private JButton botonSignup;
    private JButton btn_atras;
    private VistaPrincipal VistaPrincipal;
    private static JButton botonFondoSignUp;
    private static List<ImageIcon> framesSignUp = new ArrayList<>();
    private static int currentFrameIndexSignUp;
    private static final int ICON_WIDTH = 235;
    private static final int ICON_HEIGHT = 164;
    VistaLogin login;

    /**
     * constructor del objeto vistsignup
     */
    public VistaSignUp() {
        inicializar();
    }

    /**
     * metodo que contiene la funcionalidad y las vistas dels ignum
     */
    public void inicializar() {
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                // Carga y dibuja la imagen de fondo
                ImageIcon imageIcon = new ImageIcon("Imagenes/FondoLogin.png");
                Image image = imageIcon.getImage();
                g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
                super.paintComponent(g);
            }
        };
        // Cambia el icono de la ventana de Java
        ImageIcon icon = new ImageIcon("Imagenes/7.png");
        setIconImage(icon.getImage());

        setContentPane(panel);
        setSize(1350, 820);
        getContentPane().setLayout(null);

        JLabel labelUser = new JLabel("Usuario");
        labelUser.setFont(new Font("Tw Cen MT Condensed", Font.PLAIN, 50));
        labelUser.setBounds(281, 238, 130, 55);
        getContentPane().add(labelUser);

        JLabel labelPass = new JLabel("Contraseña");
        labelPass.setFont(new Font("Tw Cen MT Condensed", Font.PLAIN, 50));
        labelPass.setBounds(279, 300, 184, 69);
        getContentPane().add(labelPass);

        txtUser = new JTextField();
        txtUser.setFont(new Font("Tahoma", Font.PLAIN, 30));
        txtUser.setBounds(511, 238, 252, 45);
        getContentPane().add(txtUser);
        txtUser.setColumns(10);

        txtPass = new JPasswordField();
        txtPass.setBounds(511, 300, 252, 45);
        getContentPane().add(txtPass);

        // Botón atras
        btn_atras = new JButton("");
        btn_atras.setBorderPainted(false);
        btn_atras.setBounds(115, 677, 90, 78);
        btn_atras.setContentAreaFilled(false);
    }
}

```

```

btn_atras.setFocusPainted(false);
btn_atras.setOpaque(false);
ImageIcon imagenHome = new ImageIcon("Imagenes/back.png");
Image resizedImage = imagenHome.getImage().getScaledInstance(60, 60, Image.SCALE_SMOOTH);
btn_atras.setIcon(new ImageIcon(resizedImage));
getContentPane().add(btn_atras);

//BOTÓN CON FUNCIONALIDAD
botonSignup = new JButton("");
botonSignup.setFont(new Font("Tahoma", Font.PLAIN, 30));
botonSignup.setBounds(931, 438, 235, 164);
getContentPane().add(botonSignup);
// Establecer la opacidad del botón
botonSignup.setOpaque(false);
botonSignup.setContentAreaFilled(false);
botonSignup.setBorderPainted(false);
// Establecer el fondo del botón como transparente
botonSignup.setBackground(new Color(0, 0, 0, 0));

//BOTÓN DISEÑO SIGNUP
// Carga las imágenes del GIF en una lista
for (int i = 1; i <= 2; i++) {
    String imagePath = "ImagenesGifS/imagen (" + i + ").png";
    ImageIcon frame = new ImageIcon(imagePath);
    ImageIcon scaledFrame = scaleImageIcon(frame, ICON_WIDTH, ICON_HEIGHT);
    framesSignUp.add(scaledFrame);
}

// Configura el primer frame como ícono inicial del botón
currentFrameIndexSignUp = 0;

// Inicia un temporizador para cambiar automáticamente los frames
int delayS = 3000; // Intervalo en milisegundos entre cada cambio de frame
Timer timerS = new Timer(delayS, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        botonFondoSignUp.doClick(); // Simula un clic en el botón para cambiar el frame
    }
});
timerS.start();

botonFondoSignUp = new JButton();
botonFondoSignUp.setBounds(931, 438, 235, 164);
botonFondoSignUp.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Cambia la imagen del botón al siguiente frame
        currentFrameIndexSignUp = (currentFrameIndexSignUp + 1) % framesSignUp.size();
        botonFondoSignUp.setIcon(framesSignUp.get(currentFrameIndexSignUp));
    }
});
botonFondoSignUp.setIcon(framesSignUp.get(currentFrameIndexSignUp));

botonFondoSignUp.setOpaque(false);
botonFondoSignUp.setContentAreaFilled(false);
botonFondoSignUp.setBorderPainted(false);
botonFondoSignUp.setBackground(new Color(0, 0, 0, 0));

getContentPane().add(botonFondoSignUp);

JLabel labelRepetirPass = new JLabel("Repetir Contraseña");
labelRepetirPass.setFont(new Font("Tw Cen MT Condensed", Font.PLAIN, 40));
labelRepetirPass.setBounds(244, 360, 286, 65);
getContentPane().add(labelRepetirPass);

txtRepetirPass = new JPasswordField();
txtRepetirPass.setFont(new Font("Tahoma", Font.PLAIN, 30));
txtRepetirPass.setBounds(511, 360, 252, 45);
getContentPane().add(txtRepetirPass);

alert = new JLabel("");
alert.setBounds(446, 397, 413, 16);
getContentPane().add(alert);

```

```

}

/**
 * metodo que hace visible la vista
 */
public void hacerVisible() {
    setVisible(true);
}

/**
 * metodo que hace invisible la vista
 */
public void hacerInvisible() {
    setVisible(false);
}

/**
 * metodo que cambia la vista al login
 *
 * @param x vista login
 */
public void setVistaLogin(VistaLogin x) {
    login = x;
    setControlador(x);
}

/**
 * metodo que cambia la vista a la vista principal
 *
 * @param vp vista vistaprincipal
 */
public void setVistaPrincipal(VistaPrincipal vp) {
    VistaPrincipal = vp;
    setControlador2(vp);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a la vista login
 *
 * @param x vista vistalogin
 */
public void setControlador(VistaLogin x) {
    EscuchadorSignUp escuchador = new EscuchadorSignUp();
    escuchador.setVistaPrincipal(this, login);
    botonSignup.addActionListener(escuchador);
}

/**
 * metodo que creará una nuevo controlador (boton) que redigirá a la vista
 * principal
 *
 * @param x vista vistaprincipal
 */
public void setControlador2(VistaPrincipal vp) {
    SignUpBack escuchador = new SignUpBack();
    escuchador.setVistaPrincipal(this, vp);
    btn_atras.addActionListener(escuchador);
}

//setters y getters habituales
public JLabel getAlert() {
    return alert;
}

public void setAlert(JLabel alert) {
    this.alert = alert;
}

public JPasswordField getTxtRepetirPass() {
    return txtRepetirPass;
}

public void setTxtRepetirPass(JPasswordField txtRepetirPass) {

```

```

        this.txtRepetirPass = txtRepetirPass;
    }

    public JTextField getTxtUser() {
        return txtUser;
    }

    public void setTxtUser(JTextField txtUser) {
        this.txtUser = txtUser;
    }

    public JPasswordField getTxtPass() {
        return txtPass;
    }

    public void setTxtPass(JPasswordField txtPass) {
        this.txtPass = txtPass;
    }

    private static ImageIcon scaleImageIcon(ImageIcon icon, int width, int height) {
        Image image = icon.getImage();
        Image scaledImage = image.getScaledInstance(width, height, Image.SCALE_SMOOTH);
        return new ImageIcon(scaledImage);
    }
}

```

5.3.6. Paquete Controlador

- 5.3.7. altasAlumnosAadir
- 5.3.8. altasAlumnosB
- 5.3.9. altasAlumnosConsultas
- 5.3.10. altasAreasAadir
- 5.3.11. altasAreasB
- 5.3.12. altasAreasConsultas
- 5.3.13. altasAreasHome
- 5.3.14. altasConsultas
- 5.3.15. altasHome
- 5.3.16. altasPiAadir
- 5.3.17. altasPIB
- 5.3.18. altasPiConsultas
- 5.3.19. altasPIHome
- 5.3.20. archivos
- 5.3.21. anadirAlumnosHome
- 5.3.22. buscarAlumnoEditarAlumno
- 5.3.23. buscarAlumnosAnadir
- 5.3.24. buscarAlumnosConsultas
- 5.3.25. buscarAlumnosHome
- 5.3.26. buscarAreaAadir
- 5.3.27. buscarAreaConsulta
- 5.3.28. buscarAreaEditarArea
- 5.3.29. buscarAreaHome
- 5.3.30. buscarPIAadir
- 5.3.31. buscarPIConsultas
- 5.3.32. buscarPIEditarPI
- 5.3.33. buscarPIHome
- 5.3.34. consultaAlumnoHome

- 5.3.35. consultaAreasAnadir
- 5.3.36. consultaAreasHome
- 5.3.37. consultasAltas
- 5.3.38. consultasAlumnoAñadir
- 5.3.39. consultasAlumnoConsultas
- 5.3.40. consultasAlumnosB
- 5.3.41. consultasAreasB
- 5.3.42. consultasAreasConsultas
- 5.3.43. consultasHome
- 5.3.44. consultasPIB
- 5.3.45. editarPIAnadir
- 5.3.46. editarPIConsultas
- 5.3.47. editarPIHome
- 5.3.48. EscuchadorLogIn
- 5.3.49. EscuchadorPrincipalLogIn
- 5.3.50. EscuchadorPrincipalSignUp
- 5.3.51. EscuchadorSignUp
- 5.3.52. homeAnadir
- 5.3.53. homeConsultas
- 5.3.54. loginBack
- 5.3.55. SignUpBack

5.4. Paquete Main

5.4.1. Main

```
package mvc.main;

import java.awt.EventQueue;

import mvc.vistas.*;

/**
 * Clase main que ejecutará el resto de clases. Esta clase declara todas las
 * vista e incluye acceso a todas ellas
 *
 * @author alexjun09
 *
 */
public class Main {
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {

            /**
             * metodo que declara todas las clases y las pasa por parametros a otras clases
             */
            public void run() {

                VistaPrincipal principal = new VistaPrincipal();
                VistaLogin login = new VistaLogin();
                VistaSignUp signup = new VistaSignUp();

                home vistaHome = new home();
                consultas co = new consultas();
                altas al = new altas();
                altasAlumnos aa = new altasAlumnos();
                altasPI ap = new altasPI();
                altasAreas aar = new altasAreas();
                editarAlumnos ca = new editarAlumnos();
                editarAreas car = new editarAreas();
                buscarAlumnos ba = new buscarAlumnos();
                buscarArea bar = new buscarArea();
                buscarPI bp = new buscarPI();
                editarAlumnos ea = new editarAlumnos();
            }
        });
    }
}
```

```

editarPI ei = new editarPI();

principal.hacerVisible();

// Seteo de los botones de la vista principal
principal.setVistaLogin(login);
principal.setVistaSignUp(signup);

// Seteo de los botones de Vista Login
login.setHome(vistaHome);
login.setVistaPrincipal(principal);

// Seteo de los botones de VistaSingUp
signup.setVistaLogin(login);
signup.setVistaPrincipal(principal);

// Seteo de los botones del home
vistaHome.setAltas(al);
vistaHome.setConsultas(co);

// Seteo de los botones de la vista Consultas
co.setAltas(al);
co.setHome(vistaHome);
// Botones secundarios
co.setBuscarAlumnos(ba);
co.setBuscarAreas(bar);
co.setBuscarPI(bp);

// Seteo de los botones de la vista Altas
al.setConsultas(co);
al.setHome(vistaHome);
// Botones secundarios
al.setAltasAlumnos(aa);
al.setAltasPI(ap);
al.setAltasAreas(aar);

// Seteo de los botones de la vista altasAlumnos
aa.setHome(vistaHome);
aa.setAltas(al);
aa.setConsultas(co);

// Seteo de los botones de la vista altasAreas
aar.setAltas(al);
aar.setConsultas(co);
aar.setHome(vistaHome);

// Seteo de los botones de la vista altasPI
ap.setAltas(al);
ap.setHome(vistaHome);
ap.setConsultas(co);

// Seteo de los botones de la vista consultaAlumno
ca.setHome(vistaHome);
ca.setAltas(al);
ca.setConsultas(co);

// Seteo de los botones de la vista consultaAreas
car.setAltas(al);
car.setHome(vistaHome);
car.setConsultas(co);

// Seteo de los botones de la vista buscarAlumno
ba.setAltas(al);
ba.setHome(vistaHome);
ba.setConsultas(co);
ba.setEditarAlumnos(ea);

// Seteo de los botones de la vista editarAlumnos
ea.setAltas(al);
ea.setConsultas(co);
ea.setHome(vistaHome);

// Seteo de los botones de la vista buscarAreas
bar.setAltas(al);

```

```

        bar.setConsultas(co);
        bar.setHome(vistaHome);
        bar.setEditarAreas(car);

        // Seteo de los botones de la vista buscarPI
        bp.setHome(vistaHome);
        bp.setAltas(al);
        bp.setConsultas(co);
        bp.setEditarPI(ei);

        // Seteo de los botones de las vista editarPI
        ei.setAltas(al);
        ei.setHome(vistaHome);
        ei.setConsultas(co);

    }

}

}

```

5.5. Paquete Acceso a la base de datos

5.5.1. AccesoBBDD

```

package BBDD;
import java.sql.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import javax.swing.JOptionPane;
import mvc.modelo.datosAlumno;
import mvc.modelo.datosAreas;
import mvc.modelo.datosPI;
import mvc.vistas.*;

/**
 * Esta clase contiene todos los metodos que se comunican con la base de datos
 * (altas, bajas, queries y demás)
 */
public class AccesoBBDD {
    private String driver = "com.mysql.cj.jdbc.Driver";
    private String url = "jdbc:mysql://localhost/loginpi";
    private String usuario = "root";
    private String pass = "root";

    Connection con = null;

    /**
     * Constructor vacio para comunicarse con la base de datos
     */
    public AccesoBBDD() {
    }

    altasAlumnos alumnos;

    /**
     * Constructor de acceso a la base de datos
     *
     * @param alumnos Los vista que da de alta a los alumnos
     */
    public AccesoBBDD(altasAlumnos alumnos) {
        this.alumnos = alumnos;
    }

    altasAreas areas;

    /**
     * Constructor de acceso a la base de datos
     *

```

```

    * @param areas La vista que da de alta las areas
    */
public AccesoBBDD(altasAreas areas) {
    this.areas = areas;
}

altasPI proyectos;

/**
 * Constructor de acceso a la base de datos
 *
 * @param proyectos La vista que da de alta los proyectos integradores
 */
public AccesoBBDD(altasPI proyectos) {
    this.proyectos = proyectos;
}

datosAlumno consultarAlumno;

/**
 * Constructor de acceso a la base de datos
 *
 * @param consultarAlumno La clase que almacena los datos del alumno
 */
public AccesoBBDD(datosAlumno consultarAlumno) {
    this.consultarAlumno = consultarAlumno;
}

datosAreas consultarAreas;

/**
 * Constructor de acceso a la base de datos
 *
 * @param consultarAreas La clase que almacena los datos del area
 */
public AccesoBBDD(datosAreas consultarAreas) {
    this.consultarAreas = consultarAreas;
}

datosPI consultarPI;

/**
 * Constructor de acceso a la base de datos
 *
 * @param consultarPI La clase que almacena los datos del proyecto integrador
 */
public AccesoBBDD(datosPI consultarPI) {
    this.consultarPI = consultarPI;
}

editarAlumnos editarAlumno;

/**
 * Constructor de acceso a la base de datos
 *
 * @param editarAlumnos La vista que edita los datos alumnos
 */
public AccesoBBDD(editarAlumnos editarAlumno) {
    this.editarAlumno = editarAlumno;
}

editarAreas editarArea;

/**
 * Constructor de acceso a la base de datos
 *
 * @param editarArea La vista que edita los datos de las Areas
 */
public AccesoBBDD(editarAreas editarArea) {
    this.editarArea = editarArea;
}

editarPI editarPI;

```

```

/**
 * Constructor de acceso a la base de datos
 *
 * @param editarPI La vista edita los datos de los proyectos integradores
 */
public AccesoBBDD(editarPI editarPI) {
    this.editarPI = editarPI;
}

/**
 * metodo para conectarse con la base de datos, contiene control de errores
 *
 * @return con La conexion
 */
public Connection getConexion() {
    try {
        Class.forName(driver);
        con = DriverManager.getConnection(url, usuario, pass);

        System.out.println("Conexión establecida");
    } catch (Exception e) {
        System.out.println("Error de Conexión con la BBDD" + e.getMessage());
        JOptionPane.showMessageDialog(null, "Error de Conexión con la BBDD");
        e.printStackTrace();
    }
    return con;
}

/**
 * Metodo que comprobará que los datos introducidos en el login son correctos
 *
 * @param user El usuario que desea hacer login
 *
 * @param pass La contraseña del usuario que desea hacer login
 *
 * @throws SQLException Lanza exception ya que accede a una BBDD SQL
 *
 * @return answer dependiendo de si la contraseña es correcta, o no, o si el
 *         usuario existe, devuelve un numero u otro el cual es usado en la
 *         vista del login
 */
public int logIn(String user, String pass) throws SQLException {
    int answer = 0;

    Statement stmt = con.createStatement();

    // buscamos los usuarios en la BBDD
    String query = "SELECT usuario FROM usuarios WHERE usuario = '" + user + "'";
    ResultSet resultado = stmt.executeQuery(query);

    // verificamos que la BBDD ha devuelto algo (el usuario existe)
    if (resultado.next()) {
        // verificamos que lo que ha devuelto coincide con lo escrito
        if (resultado.getString("usuario").equals(user)) {
            // buscamos la contraseña
            String query2 = "SELECT contrasena FROM usuarios WHERE usuario = '" + user +
"'";
            ResultSet resultado2 = stmt.executeQuery(query2);
            resultado2.next();
            // verificamos que la contraseña coincide con el usuario
            if (resultado2.getString("contrasena").equals(pass)) {
                System.out.println("Acceso conseguido!");
                resultado.close();
                stmt.close();
                con.close();
                answer = 1;
                // contraseña incorrecta
            } else {
                System.out.println("Acceso denegado, Contraseña incorrecta");
                answer = 2;
            }
        }
        return answer;
        // usuario no existe
    }
}

```

```

        } else {
            System.out.println("Acceso denegado, usuario no existe");
            return 3;
        }
    }

/**
 * Metodo que comprobará que los datos introducidos en el signup son validos
 *
 * @param user           El nombre de usuario que desea ser creado
 *
 * @param password       La contraseña que le quiere asignar al nuevo usuario
 *
 * @param repetirPassword Repetir la contraseña que se le desea asignar al nuevo
 *                        usuario
 *
 * @throws SQLException Lanza exception ya que accede a una BBDD SQL
 *
 * @return answer dependiendo de si la contraseña es correcta, o no, o si el
 *         usuario existe, devuelve un numero u otro el cual es usado en la
 *         vista del login
 */
public int signUp(String user, String password, String repetirPassword) throws SQLException {
    int answer = 0;

    Statement stmt = con.createStatement();
    String queryUsuario = "SELECT * FROM USUARIOS WHERE usuario = '" + user + "'";
    ResultSet resultado1 = stmt.executeQuery(queryUsuario);

    // el usuario ya existe, hay que usar otro nombre
    if (resultado1.next()) {
        answer = 1;
        System.out.println("Error, usuario ya existe");
        JOptionPane.showMessageDialog(null, "Error, usuario ya existe");
    } else {
        // la contraseña no coincide
        if (!password.equals(repetirPassword)) {
            answer = 2;
            System.out.println("error, las contraseñas no coinciden");
            JOptionPane.showMessageDialog(null, "error, las contraseñas no coinciden");
        } else {
            // el usuario no existe y las contraseñas son iguales asi que almacenamos los
            // datos en la BBDD
            if (password.length() < 5) {
                answer = 3;
                System.out.println("Error, contraseña menor a 5 caracteres");
                JOptionPane.showMessageDialog(null, "Error, contraseña menor a 5 caracteres");
            } else {
                String queryInsert = "INSERT INTO usuarios VALUES ('" + user + "','" +
password + "',null)";
                int insertar = stmt.executeUpdate(queryInsert);
                answer = 4;
                System.out.println("Usuario añadido correctamente");
                JOptionPane.showMessageDialog(null, "Usuario añadido correctamente");
            }
        }
    }
    return answer;
}

/**
 * Metodo que insertará los datos de los alumnos en la BBDD
 */
public void insertarDatosAlumnos() {
    try {
        String nombre = alumnos.getTextField().getText();
        int numeroExp = Integer.parseInt(alumnos.getTextField_1().getText());
        int idAlumno = Integer.parseInt(alumnos.getTextField_2().getText());

        // Verificar si el campo textField_3 está vacío
    }
}

```

```

        String textField3Value = alumnos.getTextField_3().getText();
        Integer idProyecto = null;
        if (!textField3Value.isEmpty()) {
            idProyecto = Integer.parseInt(textField3Value);
        }

        Connection conexion = getConexion();

        // Consulta para insertar los datos en la tabla ALUMNOS
        String consulta = "INSERT INTO alumnos (ID_Alumno, Numero_exp, Nombre, ID_Proyecto)
VALUES (?, ?, ?, ?)";

        // Crear el objeto PreparedStatement
        PreparedStatement preparedStatement = conexion.prepareStatement(consulta);

        // Insertar los datos en la consulta
        preparedStatement.setInt(1, idAlumno);
        preparedStatement.setInt(2, numeroExp);
        preparedStatement.setString(3, nombre);

        if (idProyecto != null) {
            preparedStatement.setInt(4, idProyecto);
        } else {
            preparedStatement.setNull(4, java.sql.Types.INTEGER);
        }

        // Ejecutar la consulta
        preparedStatement.executeUpdate();

        // Cerrar el PreparedStatement y la conexión
        preparedStatement.close();
        conexion.close();

        System.out.println("Datos insertados correctamente en la tabla Alumnos.");
        JOptionPane.showMessageDialog(null, "Alumno insertado correctamente");
    } catch (SQLException e) {
        System.out.println("Error al insertar los datos: " + e.getMessage());
        JOptionPane.showMessageDialog(null, "Error al insertar los datos");

    } catch (NumberFormatException e) {
        System.out.println("Error de formato numérico: " + e.getMessage());
        JOptionPane.showMessageDialog(null, "Error de formato numérico");
    }
}

/**
 * Método que insertará los datos de las áreas en la BBDD
 */
public void insertarDatosAreas() {
    try {
        int idArea = Integer.parseInt(areas.getTextField().getText());
        String descripcion = areas.getTextField_1().getText();

        Connection conexion = getConexion();

        String consulta2 = "INSERT INTO areas (ID_Area, DESCRIPCION) VALUES (?, ?)";

        PreparedStatement preparedStatement = conexion.prepareStatement(consulta2);

        preparedStatement.setInt(1, idArea);
        preparedStatement.setString(2, descripcion);

        preparedStatement.executeUpdate();

        preparedStatement.close();
        conexion.close();

        System.out.println("Datos insertados correctamente en la tabla Areas.");
        JOptionPane.showMessageDialog(null, "Área insertada correctamente");
    } catch (SQLException e) {
        System.out.println("Error al insertar los datos: " + e.getMessage());
    }
}

```

```

        JOptionPane.showMessageDialog(null, "Error al insertar los datos: " + e.getMessage());
    }

}

/**
 * Metodo que insertará los datos de los proyectos integradores en la BBDD
 */
public void insertarDatosPI() throws ParseException {
    try {

        int idProyecto = Integer.parseInt(proyectos.getTextField_7().getText());
        String nombre = proyectos.getTextField().getText();
        int notaObtenida = Integer.parseInt(proyectos.getTextField_8().getText());
        String fechaTexto = proyectos.getTextField_4().getText();
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd");
        Date ano = dateFormat.parse(fechaTexto);

        try {
            ano = dateFormat.parse(fechaTexto);
        } catch (ParseException e) {
            System.out.println("Error al convertir la fecha: " + e.getMessage());
            JOptionPane.showMessageDialog(null, "Error al convertir la fecha: " +
e.getMessage());
        }

        return;
    }

    int grupo = Integer.parseInt(proyectos.getTextField_5().getText());
    String urlProyecto = proyectos.getTextField_1().getText();
    int curso = Integer.parseInt(proyectos.getTextField_6().getText());
    int idArea = Integer.parseInt(proyectos.getTextField_2().getText());

    Connection conexion = getConexion();

    String consulta3 = "INSERT INTO proyecto (ID_Proyecto, Nombre, Nota_Obtenerida, Ano, Grupo, URL_Proyecto, Curso, ID_Area) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";

    PreparedStatement preparedStatement = conexion.prepareStatement(consulta3);

    preparedStatement.setInt(1, idProyecto);
    preparedStatement.setString(2, nombre);
    preparedStatement.setInt(3, notaObtenida);
    preparedStatement.setDate(4, new java.sql.Date(ano.getTime()));
    preparedStatement.setInt(5, grupo);
    preparedStatement.setString(6, urlProyecto);
    preparedStatement.setInt(7, curso);
    preparedStatement.setInt(8, idArea);

    preparedStatement.executeUpdate();
    preparedStatement.close();
    conexion.close();

    System.out.println("Datos insertados correctamente en la tabla Proyectos.");
    JOptionPane.showMessageDialog(null, "Proyecto insertado correctamente");
} catch (SQLException e) {
    System.out.println("Error al insertar los datos: " + e.getMessage());
    JOptionPane.showMessageDialog(null, "Error al insertar los datos: " + e.getMessage());
}
}

/**
 * metodo que generará un arraylist con los datos de los alumnos
 *
 * @return listaAlumnos La lista con los alumnos y sus correspondientes datos
 */
public ArrayList<datosAlumno> obtenerDatosAlumnos() {
    ArrayList<datosAlumno> listaAlumnos = new ArrayList<>();

    try {
        Connection conexion = getConexion();

```

```

// Consulta para obtener los datos de los alumnos
String consulta = "SELECT ID_Alumno, Numero_exp, Nombre, ID_Proyecto FROM alumnos";

// Crear el objeto PreparedStatement
PreparedStatement preparedStatement = conexion.prepareStatement(consulta);

// Ejecutar la consulta y obtener el resultado
ResultSet resultSet = preparedStatement.executeQuery();

// Iterar sobre el resultado y procesar los datos
while (resultSet.next()) {
    int idAlumno = resultSet.getInt("ID_Alumno");
    int numeroExp = resultSet.getInt("Numero_exp");
    String nombre = resultSet.getString("Nombre");
    int idProyecto = resultSet.getInt("ID_Proyecto");

    // Crear un objeto datosAlumno y agregarlo a la lista
    datosAlumno alumno = new datosAlumno();
    alumno.setID_Alumno(idAlumno);
    alumno.setNumero_exp(numeroExp);
    alumno.setNombre(nombre);
    alumno.setID_Proyecto(idProyecto);

    listaAlumnos.add(alumno);
}

// Cerrar el ResultSet, el PreparedStatement y la conexión
resultSet.close();
preparedStatement.close();
conexion.close();

System.out.println("Consulta de datos de alumnos completada.");
} catch (SQLException e) {
    System.out.println("Error al consultar los datos de los alumnos: " + e.getMessage());
}

return listaAlumnos;
}

/**
 * metodo que generará un arraylist con los datos de las areas
 *
 * @return AREAS La lista con las areas y sus correspondientes datos
 */
public ArrayList<datosAreas> obtenerDatosAreas() {

    ArrayList<datosAreas> AREAS = new ArrayList<>();
    try {
        Connection conexion = getConexion();

        // Consulta para obtener los datos de los alumnos
        String consulta = "SELECT ID_Area, DESCRIPCION FROM areas";

        // Crear el objeto PreparedStatement
        PreparedStatement preparedStatement = conexion.prepareStatement(consulta);

        // Ejecutar la consulta y obtener el resultado
        ResultSet resultSet = preparedStatement.executeQuery();

        // Iterar sobre el resultado y procesar los datos
        while (resultSet.next()) {
            int idArea = resultSet.getInt("ID_Area");
            String descripcion = resultSet.getString("DESCRIPCION");

            // Crear un objeto datosAlumno y agregarlo a la lista
            datosAreas areasob = new datosAreas();
            areasob.setID_Area(idArea);
            areasob.setDESCRIPCION(descripcion);
            AREAS.add(areasob);
        }

        // Cerrar el ResultSet, el PreparedStatement y la conexión
        resultSet.close();
        preparedStatement.close();
        conexion.close();
    }
}

```

```

        System.out.println("Consulta de datos de alumnos completada.");
    } catch (SQLException e) {
        System.out.println("Error al consultar los datos de los alumnos: " + e.getMessage());
    }

    return AREAS;
}

/**
 * metodo que generará un arraylist con los datos de los proyectos integradores
 *
 * @return PI La lista con los proyectos integradores y sus correspondientes
 *         datos
 */
public ArrayList<datosPI> obtenerDatosPI() {

    ArrayList<datosPI> PI = new ArrayList<>();
    try {
        Connection conexion = getConexion();

        // Consulta para obtener los datos de los alumnos
        String consulta = "SELECT ID_Proyecto, nombre, Nota_obtenida, Ano, Grupo, URL_Proyecto,
Curso, ID_Area FROM proyecto";

        // Crear el objeto PreparedStatement
        PreparedStatement preparedStatement = conexion.prepareStatement(consulta);

        // Ejecutar la consulta y obtener el resultado
        ResultSet resultSet = preparedStatement.executeQuery();

        // Iterar sobre el resultado y procesar los datos
        while (resultSet.next()) {
            int ID_Proyecto = resultSet.getInt("ID_Proyecto");
            String nombre = resultSet.getString("nombre");
            int Nota_obtenida = resultSet.getInt("Nota_obtenida");
            Date Ano = resultSet.getDate("Ano");
            int Grupo = resultSet.getInt("Grupo");
            String URL_Proyecto = resultSet.getString("URL_Proyecto");
            int Curso = resultSet.getInt("Curso");
            int ID_Area = resultSet.getInt("ID_Area");

            // Crear un objeto datosAlumno y agregarlo a la lista
            datosPI datosPI = new datosPI();
            datosPI.setID_Proyecto(ID_Proyecto);
            datosPI.setNombre(nombre);
            datosPI.setAno((java.sql.Date) Ano);
            datosPI.setGrupo(Grupo);
            datosPI.setURL_Proyecto(URL_Proyecto);
            datosPI.setCurso(Curso);
            datosPI.setID_Area(ID_Area);
            PI.add(datosPI);
        }

        // Cerrar el ResultSet, el PreparedStatement y la conexión
        resultSet.close();
        preparedStatement.close();
        conexion.close();

        System.out.println("Consulta de datos de alumnos completada.");
    } catch (SQLException e) {
        System.out.println("Error al consultar los datos de los alumnos: " + e.getMessage());
    }

    return PI;
}

/**
 * Metodo que cogera los datos de los alumnos de la BBDD y los actualizará con
 * los nuevos datos proporcionados por el usuario
 */
public void editarDatosAlumnos() {

```

```

try {
    String nombre = editarAlumno.getTextField().getText();
    int numeroExp = Integer.parseInt(editarAlumno.getTextField_1().getText());
    int idAlumno = Integer.parseInt(editarAlumno.getTextField_2().getText());
    int idProyecto = Integer.parseInt(editarAlumno.getTextField_3().getText());
    String nuevoIdAlumnoString = editarAlumno.getTextField_4().getText();
    Integer nuevoIdAlumno = null; // Valor por defecto

    if (!nuevoIdAlumnoString.isEmpty()) {
        nuevoIdAlumno = Integer.parseInt(nuevoIdAlumnoString);
    }

    Connection conexion = getConexion();

    String consulta;
    PreparedStatement preparedStatement;

    if (nuevoIdAlumno != null) {
        consulta = "UPDATE alumnos SET Numero_exp = ?, Nombre = ?, ID_Proyecto = ?," +
        "ID_Alumno = ? WHERE ID_Alumno = ?";
        preparedStatement = conexion.prepareStatement(consulta);
        preparedStatement.setInt(1, numeroExp);
        preparedStatement.setString(2, nombre);
        preparedStatement.setInt(3, idProyecto);
        preparedStatement.setInt(4, nuevoIdAlumno);
        preparedStatement.setInt(5, idAlumno);
    } else {
        consulta = "UPDATE alumnos SET Numero_exp = ?, Nombre = ?, ID_Proyecto = ?" +
        "WHERE ID_Alumno = ?";
        preparedStatement = conexion.prepareStatement(consulta);
        preparedStatement.setInt(1, numeroExp);
        preparedStatement.setString(2, nombre);
        preparedStatement.setInt(3, idProyecto);
        preparedStatement.setInt(4, idAlumno);
    }

    int filasActualizadas = preparedStatement.executeUpdate();

    preparedStatement.close();
    conexion.close();

    if (filasActualizadas > 0) {
        System.out.println("Datos actualizados correctamente en la tabla Alumnos.");
        JOptionPane.showMessageDialog(null, "Datos actualizados correctamente");
    } else {
        System.out.println("No se encontró ningún alumno con el ID especificado.");
        JOptionPane.showMessageDialog(null, "No se encontró ningún alumno con el ID" +
        "especificado.");
    }
} catch (SQLException e) {
    System.out.println("Error al actualizar los datos: " + e.getMessage());
    JOptionPane.showMessageDialog(null, "Error al actualizar los datos: " + e.getMessage());
} catch (NumberFormatException e) {
    System.out.println("Error de formato numérico: " + e.getMessage());
    JOptionPane.showMessageDialog(null, "Error al actualizar los datos: " + e.getMessage());
}
}

/**
 * Método que cogera los datos de las áreas de la BBDD y los actualizará con los
 * nuevos datos proporcionados por el usuario
 */
public void editarDatosAreas() {
    try {
        int idArea = Integer.parseInt(editarArea.getTextField().getText());
        String descripcion = editarArea.getTextField_1().getText();

        // Obtener el nuevo valor de ID_Area
        String nuevoIdAreaString = editarArea.getTextField_2().getText();
        Integer nuevoIdArea = null; // Valor por defecto
    }
}

```

```

        if (!nuevoIdAreaString.isEmpty()) {
            nuevoIdArea = Integer.parseInt(nuevoIdAreaString);
        }

        Connection conexion = getConexion();

        String consulta;
        PreparedStatement preparedStatement;

        if (nuevoIdArea != null) {
            consulta = "UPDATE areas SET ID_Area = ?, DESCRIPCION = ? WHERE ID_Area = ?";
            preparedStatement = conexion.prepareStatement(consulta);
            preparedStatement.setInt(1, nuevoIdArea);
            preparedStatement.setString(2, descripcion);
            preparedStatement.setInt(3, idArea);
        } else {
            consulta = "UPDATE areas SET DESCRIPCION = ? WHERE ID_Area = ?";
            preparedStatement = conexion.prepareStatement(consulta);
            preparedStatement.setString(1, descripcion);
            preparedStatement.setInt(2, idArea);
        }

        preparedStatement.executeUpdate();

        preparedStatement.close();
        conexion.close();

        System.out.println("Datos actualizados correctamente en la tabla Areas.");
        JOptionPane.showMessageDialog(null, "Datos actualizados correctamente en la tabla Areas.");
    } catch (SQLException e) {
        System.out.println("Error al actualizar los datos: " + e.getMessage());
        JOptionPane.showMessageDialog(null, "Error al actualizar los datos: " + e.getMessage());
    } catch (NumberFormatException e) {
        System.out.println("Error de formato numérico: " + e.getMessage());
    }
}

/**
 * Método que cogera los datos los proyectos integradores de la BBDD y los
 * actualizará con los nuevos datos proporcionados por el usuario
 */
public void editarDatosPI() throws ParseException {
    try {
        String idProyectoString = editarPI.getTextField().getText();
        int idProyecto = Integer.parseInt(idProyectoString);
        String nombre = editarPI.getTextField_3().getText();
        int notaObtenida = Integer.parseInt(editarPI.getTextField_4().getText());
        String fechaTexto = editarPI.getTextField_5().getText();
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd");
        Date ano = dateFormat.parse(fechaTexto);
        String nuevoIdProyecto = editarPI.getTextField_10().getText();
        Integer nuevoIdProyectoInteger = null; // Valor por defecto

        try {
            ano = dateFormat.parse(fechaTexto);
        } catch (ParseException e) {
            System.out.println("Error al convertir la fecha: " + e.getMessage());
            return;
        }

        int grupo = Integer.parseInt(editarPI.getTextField_6().getText());
        String urlProyecto = editarPI.getTextField_7().getText();
        int curso = Integer.parseInt(editarPI.getTextField_8().getText());
        int idArea = Integer.parseInt(editarPI.getTextField_9().getText());

        Connection conexion = getConexion();

        String consulta;
        PreparedStatement preparedStatement;
    }
}

```

```

        if (!nuevoIdProyecto.isEmpty()) {
            nuevoIdProyectoInteger = Integer.parseInt(nuevoIdProyecto);
            consulta = "UPDATE PROYECTO SET ID_Proyecto = ?, Nombre = ?, Nota_Obttenida = ?,
Ano = ?, Grupo = ?, URL_Proyecto = ?, Curso = ?, ID_Area = ? WHERE ID_Proyecto = ?";
            preparedStatement = conexion.prepareStatement(consulta);
            preparedStatement.setInt(1, nuevoIdProyectoInteger);
            preparedStatement.setString(2, nombre);
            preparedStatement.setInt(3, notaObtenida);
            preparedStatement.setDate(4, new java.sql.Date(ano.getTime()));
            preparedStatement.setInt(5, grupo);
            preparedStatement.setString(6, urlProyecto);
            preparedStatement.setInt(7, curso);
            preparedStatement.setInt(8, idArea);
            preparedStatement.setInt(9, idProyecto);
        } else {
            consulta = "UPDATE PROYECTO SET Nombre = ?, Nota_Obttenida = ?, Ano = ?, Grupo =
?, URL_Proyecto = ?, Curso = ?, ID_Area = ? WHERE ID_Proyecto = ?";
            preparedStatement = conexion.prepareStatement(consulta);
            preparedStatement.setString(1, nombre);
            preparedStatement.setInt(2, notaObtenida);
            preparedStatement.setDate(3, new java.sql.Date(ano.getTime()));
            preparedStatement.setInt(4, grupo);
            preparedStatement.setString(5, urlProyecto);
            preparedStatement.setInt(6, curso);
            preparedStatement.setInt(7, idArea);
            preparedStatement.setInt(8, idProyecto);
        }
        preparedStatement.executeUpdate();
        preparedStatement.close();
        conexion.close();

        System.out.println("Los datos del proyecto se han actualizado correctamente.");
        JOptionPane.showMessageDialog(null, "Los datos del proyecto se han actualizado
correctamente");
    } catch (SQLException e) {
        System.out.println("Error al actualizar los datos: " + e.getMessage());
        JOptionPane.showMessageDialog(null, "Error al actualizar los datos: " + e.getMessage());
    }
}

/**
 * metodo que eliminará los datos de los alumnos de la BBDD
 */
public void borrarDatosAlumnos() {
    try {
        int idAlumno = Integer.parseInt(editarAlumno.getTextField_2().getText());

        Connection conexion = getConexion();

        // Consulta para eliminar los datos de la tabla ALUMNOS
        String consulta = "DELETE FROM alumnos WHERE ID_Alumno = ?";

        // Crear el objeto PreparedStatement
        PreparedStatement preparedStatement = conexion.prepareStatement(consulta);

        // Insertar los datos en la consulta
        preparedStatement.setInt(1, idAlumno);

        // Ejecutar la consulta
        int filasBorradas = preparedStatement.executeUpdate();

        // Cerrar el PreparedStatement y la conexión
        preparedStatement.close();
        conexion.close();

        if (filasBorradas > 0) {
            System.out.println("Datos eliminados correctamente de la tabla Alumnos.");
            System.out.println("Datos eliminados correctamente de la tabla Alumnos");
        } else {
            System.out.println("No se encontró ningún alumno con el ID especificado.");
            System.out.println("No se encontró ningún alumno con el ID especificado.");
        }
    }
}

```

```

        } catch (SQLException e) {
            System.out.println("Error al eliminar los datos: " + e.getMessage());
            JOptionPane.showMessageDialog(null, "Error al eliminar los datos: " + e.getMessage());
        } catch (NumberFormatException e) {
            System.out.println("Error de formato numérico: " + e.getMessage());
            JOptionPane.showMessageDialog(null, "Error de formato numérico: " + e.getMessage());
        }
    }

    /**
     * metodo que eliminará los datos de las areas de la BBDD
     */
    public void borrarDatosAreas() {
        try {
            int idArea = Integer.parseInt(editarArea.getTextField().getText());

            Connection conexion = getConexion();

            // Consulta para eliminar los datos de la tabla AREAS
            String consulta = "DELETE FROM areas WHERE ID_Area = ?";

            // Crear el objeto PreparedStatement
            PreparedStatement preparedStatement = conexion.prepareStatement(consulta);

            // Insertar los datos en la consulta
            preparedStatement.setInt(1, idArea);

            // Ejecutar la consulta
            int filasBorradas = preparedStatement.executeUpdate();

            // Cerrar el PreparedStatement y la conexión
            preparedStatement.close();
            conexion.close();

            if (filasBorradas > 0) {
                System.out.println("Datos eliminados correctamente de la tabla Areas.");
                JOptionPane.showMessageDialog(null, "Datos eliminados correctamente de la tabla Areas");
            } else {
                System.out.println("No se encontró ninguna área con el ID especificado.");
                JOptionPane.showMessageDialog(null, "No se encontró ninguna área con el ID especificado");
            }
        } catch (SQLException e) {
            System.out.println("Error al eliminar los datos: " + e.getMessage());
            JOptionPane.showMessageDialog(null, "Error al eliminar los datos: " + e.getMessage());
        } catch (NumberFormatException e) {
            System.out.println("Error de formato numérico: " + e.getMessage());
            JOptionPane.showMessageDialog(null, "Error de formato numérico: " + e.getMessage());
        }
    }

    /**
     * metodo que eliminará los datos de los proyectos integradores de la BBDD
     */
    public void borrarDatosPI() {
        try {
            String idProyectoString = editarPI.getTextField().getText();
            int idProyecto = Integer.parseInt(idProyectoString);

            Connection conexion = getConexion();

            String consulta = "DELETE FROM PROYECTO WHERE ID_Proyecto = ?";
            PreparedStatement preparedStatement = conexion.prepareStatement(consulta);
            preparedStatement.setInt(1, idProyecto);

            int filasBorradas = preparedStatement.executeUpdate();

            preparedStatement.close();
            conexion.close();

            if (filasBorradas > 0) {
                System.out.println("Los datos del proyecto se han borrado correctamente.");
            }
        }
    }
}

```

```
        JOptionPane.showMessageDialog(null, "Los datos del proyecto se han borrado  
correctamente");  
    } else {  
        System.out.println("No se encontró ningún proyecto con el ID especificado.");  
        JOptionPane.showMessageDialog(null, "No se encontró ningún proyecto con el ID  
especificado");  
    }  
  
} catch (SQLException e) {  
    System.out.println("Error al borrar los datos: " + e.getMessage());  
    JOptionPane.showMessageDialog(null, "Error al borrar los datos: " + e.getMessage());  
}  
}
```