**Descripción del Proyecto**

El proyecto "Academia de Ajedrez HerMar" es una aplicación de gestión que permite a una academia virtual gestionar a sus alumnos y profesores, así como registrar torneos y proporcionar análisis y planes de entrenamiento personalizados. Esta versión inicial del proyecto está desarrollada en Java y utiliza MySQL como base de datos para la persistencia de la información.

**Estructura del Proyecto**

El proyecto sigue el patrón de diseño **Modelo-Vista-Controlador (MVC)** para separar la lógica de negocio, la gestión de datos y la interfaz de usuario. La organización de las carpetas es la siguiente:

Tabla

Descripción generada automáticamente

**Requisitos del Sistema**

Para ejecutar el proyecto, asegúrate de tener instalados los siguientes componentes:

* **Java 8 o superior**
* **Apache Maven**
* **MySQL**
* **MySQL Workbench** (opcional, pero recomendado para la gestión de la base de datos)
* **Conector JDBC de MySQL** (incluido en la carpeta lib/)

**Instrucciones de Configuración**

**Paso 1: Clonar el Repositorio**

Primero, debes clonar el repositorio de GitHub en tu máquina local:

git clone https://github.com/zuckonit23/ProyectoAjedrez.git

cd ProyectoAjedrez

**Paso 2: Configurar la Base de Datos MySQL**

1. **Crear una nueva base de datos**: Abre **MySQL Workbench** o cualquier herramienta de gestión de MySQL y ejecuta el siguiente comando para crear la base de datos:

CREATE DATABASE academia\_ajedrez;

USE academia\_ajedrez;

1. **Importar el archivo database.sql**: Carga el script database.sql para crear las tablas necesarias e insertar los datos de prueba. Puedes hacerlo desde **MySQL Workbench** (usando la opción de importar) o desde la línea de comandos:

mysql -u root -p academia\_ajedrez < database.sql

**Paso 3: Configurar el Proyecto Java**

1. **Abrir el Proyecto en un IDE**:
   * Puedes abrir el proyecto en **IntelliJ IDEA**, **NetBeans** o cualquier otro IDE de tu preferencia.
2. **Añadir el Conector JDBC de MySQL**:
   * El archivo **mysql-connector-j-9.1.0.jar** ya está incluido en la carpeta lib/. Asegúrate de agregarlo al **classpath** de tu proyecto.
   * En **IntelliJ IDEA**, puedes hacerlo haciendo clic derecho en el archivo y seleccionando "Add as Library".
3. **Configurar Dependencias en Maven**:
   * El archivo pom.xml ya incluye la dependencia del conector JDBC. Si estás utilizando Maven, asegúrate de instalar todas las dependencias necesarias ejecutando:

mvn clean install

**Paso 4: Configurar la Conexión a la Base de Datos**

El archivo de configuración para la conexión a la base de datos (ConexionBD.java) está preparado para conectarse con **MySQL**. Asegúrate de modificar el archivo con tus credenciales si es necesario:

Texto

Descripción generada automáticamente

**Paso 5: Ejecutar el Proyecto**

Una vez configurada la base de datos y las dependencias, ya puedes ejecutar el proyecto:

* **Desde el IDE**: Abre la clase MainView.java en la carpeta src/main/java/com/academiaajedrez/view/ y ejecútala.
* **Desde la Terminal** (si usas Maven):

mvn exec:java -Dexec.mainClass="com.academiaajedrez.view.MainView"

**Credenciales de Prueba**

Para facilitar las pruebas, el sistema incluye algunos usuarios ya registrados. Usa las siguientes credenciales para iniciar sesión:

**Profesores**

1. **Nombre de Usuario**: profesor1
   * **Contraseña**: pass123
   * **Correo Electrónico**: juan.perez@example.com
   * **Experiencia**: 10 años
2. **Nombre de Usuario**: profesor2
   * **Contraseña**: pass456
   * **Correo Electrónico**: maria.lopez@example.com
   * **Experiencia**: 8 años

**Alumnos**

1. **Nombre de Usuario**: alumno1
   * **Contraseña**: alumno123
   * **Elo FIDE**: 1500
   * **Elo Virtual**: 1600
   * **Estado de Aprobación**: Aprobado
2. **Nombre de Usuario**: alumno2
   * **Contraseña**: alumno456
   * **Elo FIDE**: 1400
   * **Elo Virtual**: 1500
   * **Estado de Aprobación**: Aprobado

**Funcionalidades Principales**

El sistema incluye las siguientes funcionalidades que se pueden probar:

* **Inicio de Sesión**: Los usuarios pueden iniciar sesión con las credenciales proporcionadas.
* **Registro de Alumnos**: Los nuevos alumnos pueden registrarse y esperar la aprobación de un profesor.
* **Aprobación de Alumnos**: Los profesores pueden aprobar a los alumnos que están en espera.
* **Registro de Torneos**: Los alumnos pueden registrar los torneos en los que han participado.
* **Visualización de Perfil**: Los alumnos aprobados pueden ver su información personal detallada.
* **Visualización de Torneos por Parte del Profesor**: Los profesores pueden revisar los torneos registrados por los alumnos.

**Notas Adicionales**

* **Migración a MySQL**: El proyecto ha migrado de un enfoque de persistencia basado en serialización a uno basado en MySQL para mejorar la integridad de los datos y la escalabilidad.
* **Patrón DAO (Data Access Object)**: Todas las operaciones de acceso a la base de datos están encapsuladas mediante el patrón DAO, lo cual permite separar la lógica de persistencia de la lógica de negocio.