**Configuración del Proyecto en Relación al Entorno de Desarrollo**

**1. Introducción:** El proyecto proporcionado es una aplicación de ajedrez desarrollada en Java, utilizando el JDK (Java Development Kit) y el entorno de desarrollo Eclipse. En este informe, se analizará la configuración del proyecto en relación con Eclipse IDE y el uso del JDK para su desarrollo.

**2. Configuración del Proyecto en Eclipse:**

**a) Estructura del Proyecto:** El proyecto está organizado en varios archivos Java, incluyendo las clases:

* **ChessBoardGUI.java**,
* **Registro.java**,
* **BaseDeDatos.java**
* **Main.java**.

Estas clases contienen la lógica del juego, la gestión del registro de movimientos y la interacción con la base de datos. Además, hay tres carpetas importantes en el proyecto:

* **"BD":** Contiene la base de datos Access utilizada por la aplicación para almacenar información relacionada con el juego de ajedrez.
* **"imágenes":** Almacena las imágenes del tablero de ajedrez y las piezas del juego, que son utilizadas por la aplicación para la representación visual en la interfaz gráfica.
* **"documentos":** Contiene un archivo de registro donde se registran los movimientos de ajedrez junto con las marcas de tiempo.

**b) Gestión de Dependencias:** El proyecto utiliza las librerías estándar de Java Swing y AWT para la creación de la interfaz gráfica por lo que no depende de bibliotecas externas aparte de las mencionadas del JDK estándar de Java. Todas las bibliotecas utilizadas son parte del Java Standard Library y son ampliamente utilizadas en el desarrollo de aplicaciones Java.

***Configuración del Sistema de Versionado*** El sistema de control de versiones utilizado para este proyecto ha sido Git que me ha permitiendo gestionar el código fuente de manera efectiva desde diferentes puestos de trabajo.

***Configuración del Build Path*** Eclipse ha gestiona automáticamente el Build Path para este proyecto. Las clases y recursos del proyecto están disponibles en el classpath, lo que permite la compilación y ejecución sin problemas.

**3. Uso del JDK:**

**a) JDK en el Proyecto:** El proyecto hace uso del JDK para compilar y ejecutar las clases Java. El código fuente se compila utilizando el compilador Java (javac) proporcionado por el JDK para generar archivos de clase (.class), que son ejecutables en la JVM.

**b) Características del JDK Utilizadas:** El JDK proporciona las herramientas necesarias para el desarrollo Java, incluyendo el compilador, las bibliotecas estándar, y las herramientas de depuración. Estas características son esenciales para el desarrollo del proyecto.

**4. Carpetas "BD", "imágenes" y "documentos":**

La carpeta "BD" contiene la base de datos Access utilizada por la aplicación para almacenar información sobre los movimientos de ajedrez y otros datos relacionados con el juego. La carpeta "imágenes" aloja las imágenes del tablero de ajedrez (las piezas del juego), mientras que la carpeta "documentos" contiene el archivo de registro de movimientos.

**5. Clases Específicas:**

**a) BaseDeDatos.java:** La clase **BaseDeDatos.java** se utiliza para interactuar con la base de datos Access alojada en la carpeta "BD". Utiliza JDBC (Java Database Connectivity) para realizar operaciones de base de datos, como la inserción y recuperación de movimientos de ajedrez desde la base de datos Access.

**b) Registro.java:** La clase **Registro.java** maneja el registro de los movimientos de ajedrez junto con las marcas de tiempo. Almacena esta información en un archivo de texto dentro de la carpeta "documentos".

**c) Main.java:** La clase **Main.java** contiene el método **main**, que sirve como punto de entrada para la aplicación. Inicia la aplicación de ajedrez invocando el constructor de **ChessBoardGUI.java** dentro del hilo de despacho de eventos de Swing (Swing Event Dispatch Thread) para garantizar una interfaz gráfica fluida y receptiva.

**6. Conclusiones:** El proyecto intenta hacer un uso efectivo del JDK y Eclipse IDE para el desarrollo del programa Java. Se intenta mostrar un código bien estructurado y organizado que aprovecha las funcionalidades estándar de Java y las herramientas proporcionadas por el JDK. La inclusión de las clases **BaseDeDatos.java** y **Registro.java** nos permite una interacción con una base de datos Access y un registro detallado de los movimientos del juego, respectivamente