

Manual Técnico
Luis Regalado 3224630331326
Practica 1

Software:

- **Lenguaje:** JAVA 21 con OpenJDK 21

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems.

El lenguaje Java proporciona:

- El paradigma de la programación orientada a objetos.
- Ejecución de un mismo programa en múltiples sistemas operativos y plataformas.
- Es posible utilizarlo para múltiples propósitos, desde aplicaciones de escritorio hasta en servidores web.
- Tiene una curva de aprendizaje media pero también toma lo mejor de otros lenguajes orientados a objetos, como C++.

- **Sistema Operativo:** Linux Fedora

Fedora es una distribución de Linux basada en software libre y de código abierto, mantenida por la comunidad del Proyecto Fedora y patrocinada por Red Hat. Es conocida por estar a la vanguardia en la adopción de tecnologías nuevas y emergentes, lo que la convierte en una opción popular entre desarrolladores, administradores de sistemas y usuarios avanzados

- **IDE:** Apache NetBeans 22

NetBeans es un entorno de desarrollo integrado, de código abierto y sigue la filosofía del software libre, y está hecho principalmente para el uso del lenguaje de programación Java. También existen muchísimos módulos extras para extender su funcionamiento. NetBeans es un producto libre y gratuito sin restricciones de uso y es un proyecto de código abierto de gran éxito con una gran base de usuarios y una comunidad en constante crecimiento

- **GITHUB**

Es un sistema de control de versiones de código y gestión de proyectos, a su vez también funciona como una plataforma de estilo de red social diseñada para desarrolladores para poder compartir código entre más personas y colaborar en el mismo.

- **GITFLOW**

Es un modelo de flujo de trabajo para la gestión de versiones y desarrollo de software utilizando el sistema de control de versiones distribuido Git. Fue propuesto por Vincent Driessen en 2010 y se ha convertido en uno de los enfoques más populares para organizar y gestionar el proceso de desarrollo de software en equipos.

El flujo de trabajo de Gitflow define una estructura clara y bien definida para administrar las ramas de desarrollo, las versiones y las características en un proyecto. Se basa en la idea de tener dos ramas principales:

1. master: Esta rama contiene el código estable y listo para producción. Cualquier commit en esta rama debería ser una versión completamente probada y lista para ser entregada a los usuarios.
2. develop: Esta rama es donde se integran las características y cambios en desarrollo. Se considera la rama de trabajo principal y donde se combinan las nuevas características y correcciones de errores antes de ser consideradas para su implementación en producción.

Requerimientos mínimos del entorno de desarrollo:

JAVA 21 en adelante.

JDK de la versión de JAVA instalada.

IDE de preferencia. (Recomendado NetBeans)

Diccionario de Clases:

1. Clase `ConexionDb` (Clase Pública):

- Representada en la parte superior, la clase `ConexionDb` parece estar diseñada para manejar la conexión a una base de datos.

2. Método `ConexionDb()` (Método Constructor):

- Este es el constructor de la clase. Se llama automáticamente cuando se crea una instancia de `ConexionDb`. En general, este tipo de método se utiliza para inicializar la conexión a la base de datos o configurar los parámetros necesarios para establecer dicha conexión.

3. Atributo ``connection`` (Atributo Protegido):

- Este atributo de tipo ``Connection`` probablemente almacena la conexión activa a la base de datos. El acceso a este atributo está protegido, lo que significa que solo puede ser accedido desde la propia clase ``ConexionDb`` y sus subclases.

4. Método ``disconnect()`` (Método Público):

- Este método parece ser responsable de cerrar o desconectar la conexión de la base de datos. Es un método público, por lo que puede ser llamado desde fuera de la clase ``ConexionDb``. Devuelve un tipo ``void``, lo que significa que no retorna ningún valor.

5. Atributo ``connection`` (Atributo Privado):

- Este es un atributo privado de la clase ``ConexionDb``, lo que significa que solo puede ser accedido desde dentro de la propia clase. Al igual que el atributo protegido, este también parece almacenar una conexión activa a la base de datos.

CLASE LOGS:

En la imagen proporcionada, se muestra una clase llamada ``Logs`` con varios métodos. A continuación te explico cada uno de los elementos:

1. Clase ``Logs`` (Clase Pública):

- Esta es la clase principal llamada ``Logs``. Está diseñada para manejar registros o "logs" de acciones, eventos, o mensajes.

2. Método ``Logs()`` (Método Constructor):

- Este es el constructor de la clase ``Logs``. Se ejecuta automáticamente cuando se crea una instancia de la clase. Generalmente, este método se utiliza para inicializar cualquier propiedad o configuración necesaria para registrar logs.

3. Método ``initComponents()`` (Método Protegido):

- Este método está protegido, lo que significa que solo puede ser llamado desde la clase ``Logs`` y sus subclases. Generalmente, un método llamado ``initComponents()`` se utiliza para inicializar o configurar

componentes internos, como interfaces gráficas o estructuras necesarias para el funcionamiento de la clase.

4. Método ``agregarAccion(String)`` (Método Protegido):

- Este método también está protegido y toma un parámetro de tipo ``String``. Su función parece ser agregar una acción o evento al log. El parámetro ``String`` probablemente represente el mensaje o la acción que se desea registrar. El método es de tipo ``void``, lo que indica que no retorna ningún valor.