GOBIERNO DE LA CIUDAD DE MÉXICO

AGENCIA DIGITAL DE INNOVACIÓN PÚBLICA

PILARES

Programador Junior

Proyecto Final:

GLOSARIO PARA PROYECTO FINAL

GLOSARIO

El motivo y funcionalidad de este Glosario es para poder entender correctamente la estructura de dicho Proyecto Final.

1. **JDBC**

   Es el estándar de conectividad de bases de datos de Java y proporciona un mecanismo para que los programas Java se conecten a las bases de datos. Para acceder a las bases de datos mediante JDBC, debe utilizar un controlador JDBC.

2. **PreparedStatement**

   También llamadas consultas, comandos o sentencias preparadas, son plantillas para consultas a sistemas de bases de datos en lenguaje SQL y proporciona varios métodos para establecer parámetros.

3. **ResultSet**

   Es una clase de Java que sirve para almacenar datos de una consulta que hagamos con la clase Statement. Un ResultSet lo podemos recorrer para mostrar los datos que nos interese.Proporciona varios métodos para obtener los datos de columna correspondientes a una fila.

4. **JOptionPane**

   Es una clase que nos provee una conjunto de ventanas de diálogo que es ideal, para mostrar mensajes al usuario. Ya sean informativos, advertencias, errores, confirmaciones… O incluso tenemos la posibilidad de solicitar la introducción de un dato.

5. **JPasswordField**

   Este tipo de campo es útil para pedir al usuario que introduzca passwords cuando se conecta o para validar su identidad.

6. **JTextField**

   Permite al operador del programa ingresar una cadena de caracteres por teclado.

7. **executeQuery()**

   Ejecuta una sentencia SQL que devuelve un conjunto de resultados, como una consulta SELECT. Devuelve un objeto ResultSet que contiene los resultados de la consulta.

8. **showMessageDialog**

   Este método crea una ventana que muestra un mensaje entregado en el parámetro message.

9. **.jar**

   Es un tipo de archivo que permite ejecutar aplicaciones y herramientas escritas en el lenguaje Java

10. **JFrame**

    Sirve para generar ventanas sobre las cuales añadir distintos objetos con los que podrá interactuar o no el usuario.

11. **CRUD**

    Es un conjunto de operaciones básicas que se utilizan para gestionar la información almacenada en una base de datos. Estas operaciones se utilizan en muchas aplicaciones y sistemas, y son fundamentales para la creación y gestión de datos.

12. **ActionListener**

    Se usa para detectar y manejar eventos de acción (ActionEvent): los que tienen lugar cuando se produce una acción sobre un elemento del programa.

13. **setText**

    Se usa para establecer el texto que va a mostrar la etiqueta JLabel.

14. setSelectedIndex

Su función es asignar el ítem u opción que se ha de mostrar o seleccionar por defecto en el Combo Box.

15. getConection()

Es una conexión abierta que se puede utilizar para crear sentencias JDBC que pasen nuestras sentencias SQL al controlador de la base de datos.

16. connection

Representa una conexión física con la fuente de datos. Cuando la aplicación establece una conexión con un servidor de datos.

17. ps.executeUpdate()

Nos devuelve un entero que nos dice el número de registros a los que ha afectado la operación, en caso de sentencias INSERT, UPDATE y DELETE.

18. getText

Devuelve una cadena que contiene los datos de texto del formato especificado en este objeto de datos.

19. Close()

Método que cierra el stream y libera todos los recursos del sistema asociados con él.

20. next()

Se utiliza para desplazarse por filas en un ResultSet de una en una.

21. **setVisible(false)**

    Permite visualizar la ventana.

22. **initComponents();**

    Método que es llamado por el constructor, donde se inicializan todos los componentes gráficos. Ambos representan distintas maneras de armar una interfaz gráfica.

23. **import java.sql.Connection**

    Es importante para poder hacer la conexión con la Base de Datos.

24. **import java.sql.Date**

    Dice sólo la fecha en el formato de SQL en el que el JDBC puede entender. La fecha SQL sólo contiene años, meses y días, no hay hora ni zona horaria presentes.

25. **DriverManager**

    Gestiona el conjunto de controladores Java Database Connectivity (JDBC) que están disponibles para que los utilice una aplicación.

26. **import java.sql.PreparedStatement;**

    Es una sentencia SQL precompilada. Las utilizaremos en lugar de una Statement cuando haya que ejecutar varias veces una misma sentencia SQL con distintos parámetros. Ventajas de PreparedStatement sobre Statement: Mayor velocidad de ejecución.

27. **SQLException**

    Es una ampliación de java. lang. Exception y proporciona información adicional relacionada con las anomalías que se producen en un contexto de base de datos.