

Manejo de excepciones java

En el programa, se manejan las excepciones para asegurar que los errores se manejen de manera adecuada y se informe al usuario cuando algo sale mal. A continuación, se describen los diferentes tipos de excepciones manejadas y cómo se implementaron en el código.

1. Excepciones Personalizadas

ParseoJsonException

Esta excepción se utiliza para manejar errores específicos relacionados con el parseo de JSON. Se define en el archivo ParseoJsonException.java:

```
package entregable.excepciones;

public class ParseoJsonException extends Exception {
    public ParseoJsonException(String mensaje) {
        super(mensaje);
    }
}
```

ConexionApiException

Esta excepción se utiliza para manejar errores específicos relacionados con la conexión a la API. Se define en el archivo ConexionApiException.java:

```
package entregable.excepciones;

public class ConexionApiException extends Exception {
    public ConexionApiException(String mensaje) {
        super(mensaje);
    }
}
```

RegistroException

Esta excepción se utiliza para manejar errores específicos relacionados con el registro de usuarios. Se define en el archivo RegistroException.java:

```
package entregable.controladores;

public class RegistroException extends Exception {
    public RegistroException(String mensaje) {
        super(mensaje);
    }
}
```

2. Manejo de Excepciones en el Código

Ejemplo de Manejo de Excepciones en el Registro de Usuarios

En el controlador de registro de usuarios, se manejan las excepciones para validar los datos de entrada y verificar si el usuario ya existe. Se utiliza la excepción RegistroException para manejar estos errores:

```
private void registrarUsuario() {
    // Obtiene los datos de los campos
    String email = pantallaRegistro.getEmail();
    String username = pantallaRegistro.getUsuario();
    String password = pantallaRegistro.getPassword();
    boolean terminosAceptados = pantallaRegistro.isTerminosAceptados();

    try {
        if (!validarDatos(email, username, password, terminosAceptados)) {
            return;
        }

        if (usuarioExiste(username, email)) {
            JOptionPane.showMessageDialog(pantallaRegistro, "El usuario o correo ya existe. Intente con otros datos.", "Error", JOptionPane.ERROR_MESSAGE);
            return;
        }

        // Crea un nuevo usuario e inserta en la base de datos
        Usuario.insertarUsuario(con, username, email, password, terminosAceptados);
        JOptionPane.showMessageDialog(pantallaRegistro, "Usuario registrado con éxito. Por favor, inicie sesión.", "Registro Exitoso", JOptionPane.INFORMATION_MESSAGE);

        // Redirige al login
        volverALogin();
    } catch (RegistroException e) {
        JOptionPane.showMessageDialog(pantallaRegistro, e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}
```

Ejemplo de Manejo de Excepciones en la Conexión a la Base de Datos

En la clase principal, se maneja la excepción `SQLException` para manejar errores relacionados con la conexión a la base de datos:

```
public class App {
    public static void main(String[] args) {
        String url = "jdbc:sqlite:projectDatabase.db";

        try {
            // Conexión a la base de datos
            Connection con = DriverManager.getConnection(url);
            System.out.println("Conexión exitosa a la base de datos.");

            // Crear tablas (si no existen)
            creacionDeTablas.crearTablaMoneda(con);
            creacionDeTablas.crearTablaActivos(con);
            creacionDeTablas.crearTablaTransaccion(con);
            creacionDeTablas.crearTablaUsuario(con);
            creacionDeTablas.insertarMonedas(con);

            // Inicializar la interfaz gráfica
            PantallaLogin pantallaLogin = new PantallaLogin();
            LoginController loginController = new LoginController(pantallaLogin, con);

            pantallaLogin.setVisible(true); // Mostrar la interfaz gráfica
        } catch (SQLException e) {
            System.out.println("Error al conectar con la base de datos: " + e.getMessage());
        }
    }
}
```

Ejemplo de Manejo de Excepciones en la Consulta de Precios de Criptomonedas

En el controlador de cotizaciones, se maneja la excepción `IOException` para manejar errores relacionados con la consulta de precios de criptomonedas desde una API externa:

```
private void cargarCotizacionesIniciales() {
    try {
        Map<String, Double> datos = ConsultarPrecioCripto.obtenerPrecios();
        actualizarBaseDeDatos(datos);
        vista.cargarCriptos(datos);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(vista, "Error al cargar cotizaciones iniciales: " + e.getMessage(),
            "Error", JOptionPane.ERROR_MESSAGE);
    }
}
```