

Parcial 2 Juan Felipe Velasquez

GitHub: <https://github.com/juanvelasquez0204/parcial-2>

En cuanto al modelo, se realizó una conexión a base de datos de firebase de la siguiente forma:

```
c class Facade {  
  
    private static FirebaseDatabase firebaseDB = null;  
  
    public static FirebaseDatabase getConnection() {  
        try {  
            if(firebaseDB == null) {  
                FirebaseOptions firebaseOptions = new FirebaseOptions.Builder()  
                    .setDatabaseUrl("https://base-de-datos-parcial-d2196-default-rtbd.firebaseio.com/")  
                    .setServiceAccount(new FileInputStream(new File("C:\\Users\\pipevelasquezj\\Documents\\  
                    .build();  
  
                FirebaseApp.initializeApp(firebaseOptions);  
                firebaseDB = FirebaseDatabase.getInstance();  
            }  
        } catch (FileNotFoundException ex) {  
            ex.printStackTrace();  
        }  
  
        return firebaseDB;  
    }  
}
```

A partir de esto se creó el controlador y la clase principal de Alimentos, donde se guardan los datos recolectados del view de en la siguiente clase:

```

public class ItemDAO {
    /**
     *
     * @param alimentos, list to save
     */
    public void saveItemA(String id, Map<String, Object> alimentos ){
        FirebaseDatabase connexion = null;
        try {
            connexion = Facade.getConnection();
            DatabaseReference ref = connexion.getReference("/");
            DatabaseReference childReference = ref.child("alimentos/"+ id);

            childReference.setValue(alimentos);
            Thread.sleep(2000);
        } catch (InterruptedException ex) {
            Logger.getLogger(ItemDAO.class.getName()).log(Level.SEVERE, null,
            ex);
        }
    };
    /**
     *

```

El código inicialmente cuenta con un login para el usuario, este cuenta con el manejo de excepción para cuando los datos ingresados no son correctos.

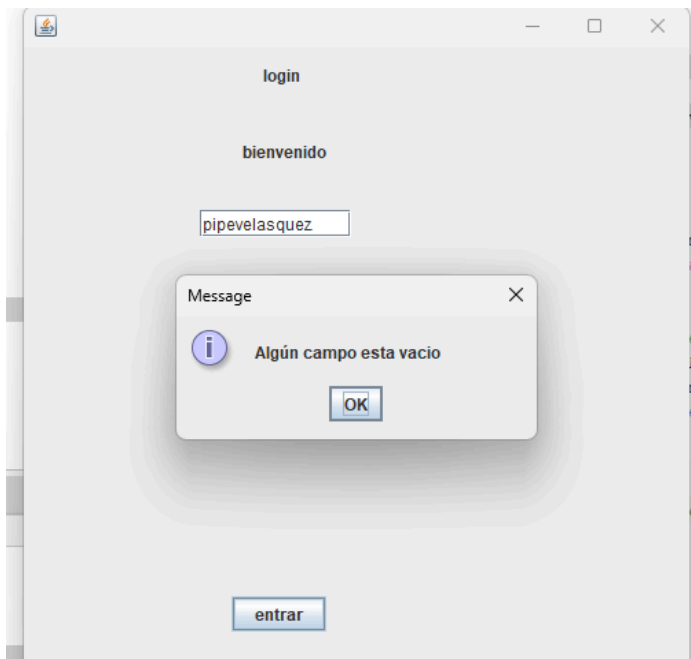
Código:

```

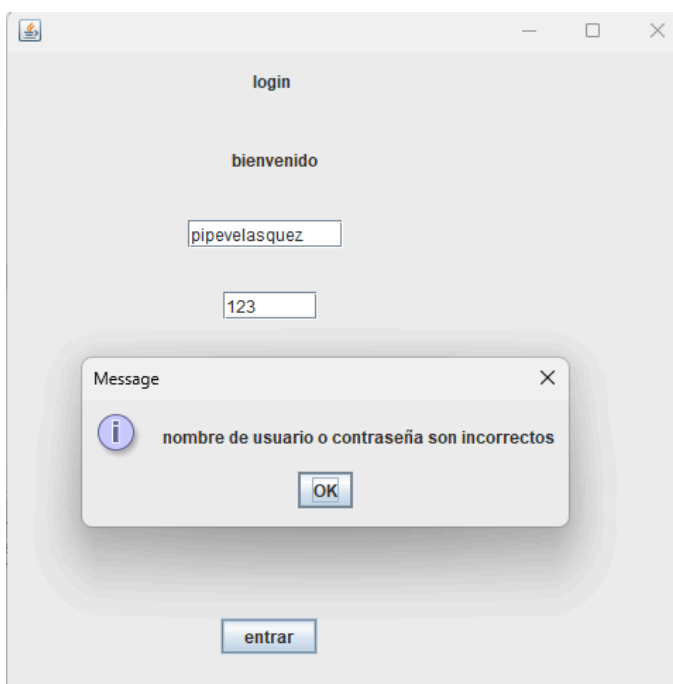
if (usuario.isEmpty() || password.isEmpty()) {
    JOptionPane.showMessageDialog(rootPane, "Algún campo esta vacio");
}
else{
    if (usuario.equals("pipevelasquez") && password.equals("1234")) {
        JOptionPane.showMessageDialog(rootPane, "bienvenido");
        homel newframe = new homel();
        newframe.setVisible(true);
        this.dispose();
    }
    else{
        JOptionPane.showMessageDialog(rootPane, "nombre de usuario o c
    }
}

```

Caso nulo:



Caso Incorrecto:



Caso correcto:

A screenshot of a Java Swing window titled "login". The window has a light gray background and standard window controls (minimize, maximize, close) in the title bar. It contains the following elements:

- The text "login" at the top center.
- The text "bienvenido" below it.
- A text input field containing the username "pipevelasquez".
- A text input field containing the password "1234".
- A blue button labeled "entrar" at the bottom center.
- A modal "Message" dialog box in the center, which is highlighted with a thick gray border. The dialog has a title bar with "Message" and a close button. It contains an information icon (a lowercase 'i' in a circle), the text "bienvenido", and an "OK" button.

En el siguiente caso, el usuario entra a la solicitud de pedidos

A screenshot of a Java Swing window titled "Añadir producto". The window has a light gray background and standard window controls. It contains the following elements:

- The text "Añadir producto" at the top center.
- Two labels: "Número" and "Cantidad".
- A dropdown menu under "Número" with "Zanahoria" selected.
- A text input field under "Cantidad" containing the value "20".
- A blue button labeled "realizar pedido" in the center.
- A blue button labeled "atras" at the bottom center.

Opciones para escoger

Zanahoria		
Lechuga	Fresa	
Tomate	Kiwi	
Zanahoria	Carambola	
Manzana	Uva	
Pera	Mora_Azul	Naranja
Banano	Aceitunas	Sandia
Cebolla	Limon	Melon
Pimenton	Mandarina	Lulo

Envío desde la app a la base de datos:

```
ItemDAO itemDao = new ItemDAO();

Long stock = (long) (Math.random() + 30);

String name = pro.getText();

int price = Integer.parseInt(pre.getText());

String id = UUID.randomUUID().toString().substring(0, 6);

Map<String, Object> item = new HashMap<String, Object>(){
    {
        put("Nombre" , name);
        put("Cantidad" , price);
        put("Stock" , stock);
    }
};

itemDao.saveItemA(id, item);

// TODO add your handling code here:
```

Registro en la base de datos:

<https://base-de-datos-parcial-d2196-default-rtdb.firebaseio.com/>

▼ — alimentos

▼ — 6550cf

Cantidad: 12

Nombre: "Tomate"

Stock: 30

▼ — 799f5e

Cantidad: 20

Nombre: "Zanahoria"

Stock: 30