Documentación Proyecto Just Eat

Programación a Internet

William Adzet Mauchline
Carlos García Rodríguez

Requisitos del proyecto y explicación de la resolución.

- **Registro y acceso a usuarios:** Para este punto se han implementado los servlet “RegisterServlet” y “LoginServlet”, en el primer servlet en el método doGet se llama al Register.jsp y en el método doPost se realiza una conexión con la base de datos, se accede al DAO del usuario y se crea un nuevo usuario en la base de datos con los datos que el usuario ha rellenado en el formulario perteneciente al Register.jsp. Una vez el usuario le da botón de “Create account”, se le envía al LoginServlet.
  En el método doGet del LoginServlet se llama al Login.jsp y en el método doPost se realiza una conexión con la base de datos, se accede al DAO de los usuarios. Una vez el usuario ha rellenado los campos del formulario de Login.jsp, se realiza una comprobación de que el usuario exista en la base de datos y en caso de que exista, que la contraseña sea la correcta. En caso de que ambos campos sean correctos se entra dentro de la aplicación (SearchServlet).
- **Gestión de restaurantes:** Para este punto se ha implementado el servlet “EditRestaurantServlet” y el archivo .jsp “EditRestaurant.jsp”, en el método doGet se realiza una conexión con la base de datos y se accede al DAO de restaurantes. Este método doGet recibe el id del restaurante mediante la cabecera para poder editar el restaurante que se ha elegido. Posteriormente se llama al EditRestaurant.jsp. En el método doPost se realiza una conexión con la base de datos y se accede al DAO de restaurantes. Los campos del formulario del EditRestaurant.jsp vendrá precargado con los datos del restaurante y el usuario propietario del restaurante podrá cambiar los datos del restaurante. Estos cambios también se verán reflejados en la base de datos.
  Para este punto y para el siguiente punto se ha implementado un servlet “OwnedRestaurantServlet” y un archivo .jsp “OwnedRestaurant.jsp” que permite al usuario ver cuales son sus restaurantes y la opción de gestionar sus restaurantes y la opción de gestionar los platos de un restaurante.

William Adzet Mauchline
Carlos García Rodríguez

- **Gestión de platos de un restaurante:** Para este punto se han implementado los servlets “EditMenuRestaurantServlet”, “DishDetailsServlet” y los archivos .jsp “EditMenuRestaurant.jsp” “DishDetails.jsp”, en el método doGet realiza una conexión con la base de datos y se acceden a los DAOs de platos y de restaurantes. Se recibe el id del restaurante para saber de que restaurante se están gestionando los platos. Se almacena en una lista los platos del restaurante todos los platos y comprobando que tengan el mismo idr que el id del restaurante. De esta manera el propietario del restaurante podrá ver los platos que tiene. En el método doPost se almacena y se muestra el nuevo plato que el propietario del restaurante ha creado. También podrá ver los detalles de los platos del restaurante.
- **Gestión de pedidos:** Para este punto se han implementado los servlet “OrderRestaurantServlet”, “ConfirmedOrderServlet”, “HistoricalServlet” y “CartServlet” y sus respectivos archivos .jsp. En el primer servlet, en el método doGet se realiza una conexión a la base de datos y recibe el id del restaurante del que quieres realizar un pedido. De igual manera que en los puntos anteriores se utiliza una lista para poder mostrar los platos del restaurante con el archivo “OrderRestaurantServlet.jsp”, el usuario podrá añadir cualquiera de los pedidos del restaurante y una vez pulse el botón el añadir le mande a “ConfirmedOrderServlet” donde se muestre un mensaje para confirmar el pedido que ha realizado. En el método doPost del primer servlet mencionado, se crea un nuevo “Order” y un nuevo “OrderDishes” y se almacenan en la base de datos. Por último, el usuario podrá comprobar los artículos añadidos al carrito, este carrito esta implementado a nivel de sesión.
- **Valoración:** Para este punto se ha implementado el servlet “ReviewServlet” y su respectivo archivo .jsp. En el método doGet se realiza una conexión con la base de datos y recibe el id del restaurante del que se quiere realizar una valoración. En el método doPost se recibe los campos que el usuario ha rellenado en el formulario, en este caso el comentario y la valoración, y se almacena en la base de datos como una nueva valoración. También mencionar el usuario cuando entre dentro de un restaurante podrá ver las reviews de otros usuarios que hayan pedido en el

restaurante. Esto se realiza en el servlet “OrderRestaurantServlet” y se muestra en el OrderRestaurant.jsp

- **Búsqueda de restaurante:** Para este punto se han implementado los servlet “SearchServlet” y “SearchContentServlet” junto a sus respectivos archivos .jsp. En la pantalla del primer servlet/jsp podemos encontrar un buscador de restaurantes por localidad, nombre o descripción de restaurante y en la pantalla del segundo servlet/jsp se muestra el resultado de la búsqueda que el usuario ha realizado.
- **Estado del restaurante:** Para este punto no se ha implementado ningun servlet ni archivo .jsp, si no que se ha modificado el servlet “SearchServlet” y el archivo .jsp. En el archivo .jsp se han añadido unos checkbox para poder seleccionar: todos, solo que aceptan pedidos y los que no aceptan pedidos. De esta manera cuando el usuario selecciona todos y busca, saldrán todos los restaurantes que cumplan la query de búsqueda, cuando selecciona solo los que aceptan pedidos saldrán los que cumplan la query y que acepten pedidos y, por último si se selecciona los restaurantes que no aceptan pedidos se mostrarán aquellos restaurantes que cumplan la query y que no acepten pedidos.

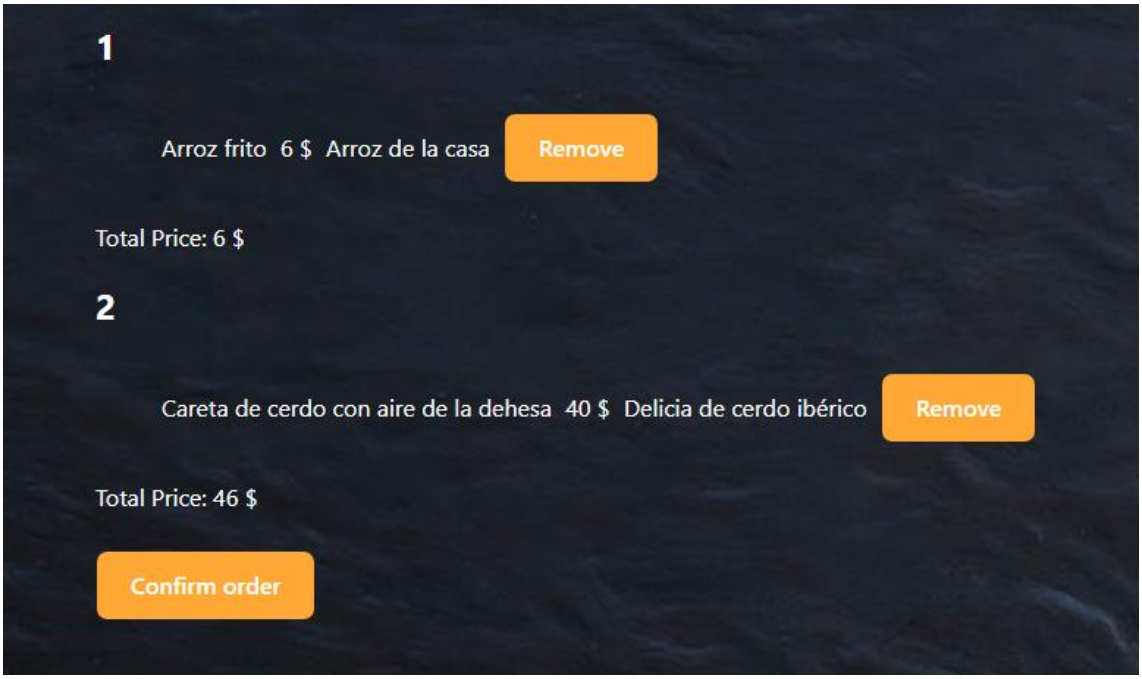
Ampliaciones

- **Cálculo de cuenta instantánea:** Para este punto se ha modificado el archivo “Cart.jsp”.

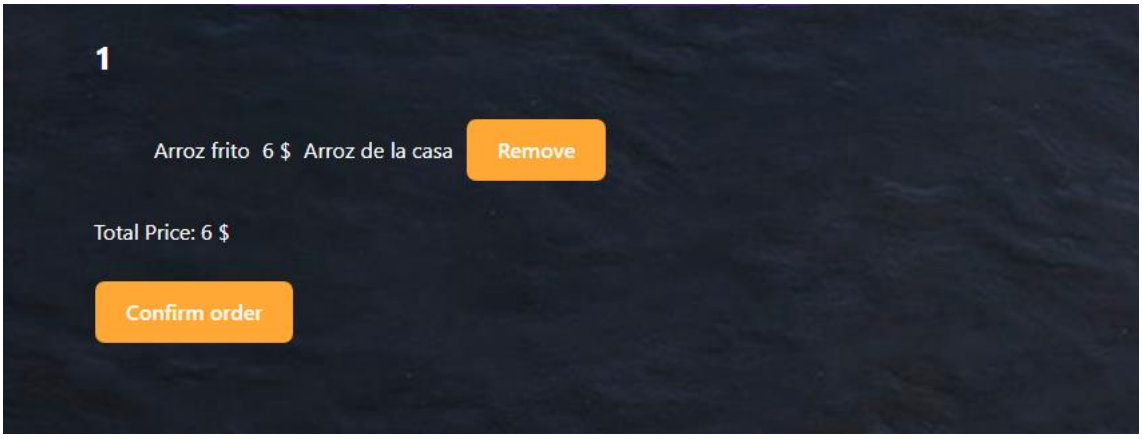
```
<c:forEach var="dish" items="${dishList}">
    <h2>${dish.id}</h2>
    <ul>
        <li>${dish.name}</li>
        <li>${dish.price} $</li>
        <li>${dish.description}</li>
        <li><button name="${dish.name}">Remove</button></li>
        <c:set var="amount" value="${amount + dish.price}"/>
    </ul>
    <p> Total Price: ${amount} $</p>
</c:forEach>
<input type="submit" name="confirmedorder" value="Confirm order">
```

William Adzet Mauchline
Carlos García Rodríguez

De esta manera cada vez que se añada un plato se le sumará el precio al precio total del pedido y cuando se elimine, se le restará el precio total del pedido.



Cuando pulsemos el botón de Remove se borrará el plato y se actualizará el precio total.



William Adzet Mauchline
Carlos García Rodríguez