

TIENDA DE COMERCIO ELECTRÓNICO

ITERACIÓN 07 – ORDEN DE COMPRA

Autor: Mag. Juan Antonio Castro Silva
Versión: 1.0 Enero 29 de 2020 (20200129T1325)

1. ORDEN DE COMPRAS

En esta iteración se va implementar la funcionalidad de registrar las ordenes de compra (purchases), para ello se deben crear unas nuevas tablas en la base de datos, componentes y servicios web

2. CREAR LAS TABLAS

Tabla purchases:

```
001 CREATE TABLE public.purchases
002 (
003     id bigserial NOT NULL,
004     user_id bigint NOT NULL,
005     created_at timestamp without time zone DEFAULT now(),
006     updated_at timestamp without time zone DEFAULT now(),
007     CONSTRAINT purchases_pkey PRIMARY KEY (id),
008     CONSTRAINT purchases_user_id_fkey FOREIGN KEY (user_id)
009         REFERENCES public.users (id) MATCH SIMPLE
010         ON UPDATE NO ACTION ON DELETE NO ACTION
011 )
012 WITH (
013     OIDS=FALSE
014 );
015 ALTER TABLE public.purchases
016     OWNER TO ecommerce;
```

Tabla purchase_items:

```
001 CREATE TABLE public.purchase_items
002 (
003     id bigserial NOT NULL,
004     purchase_id bigint NOT NULL,
005     product_id bigint NOT NULL,
006     price double precision NOT NULL,
007     quantity double precision NOT NULL,
008     created_at timestamp without time zone DEFAULT now(),
009     updated_at timestamp without time zone DEFAULT now(),
010     CONSTRAINT purchase_items_pkey PRIMARY KEY (id),
```

```

011     CONSTRAINT purchase_items_product_id_fkey FOREIGN KEY (product_id)
012     REFERENCES public.products (id) MATCH SIMPLE
013     ON UPDATE NO ACTION ON DELETE NO ACTION,
014     CONSTRAINT purchase_items_purchase_id_fkey FOREIGN KEY (purchase_id)
015     REFERENCES public.purchases (id) MATCH SIMPLE
016     ON UPDATE NO ACTION ON DELETE NO ACTION
017 )
018 WITH (
019     OIDS=FALSE
020 );
021 ALTER TABLE public.purchase_items
022     OWNER TO ecommerce;

```

3. ASIGNAR PERMISOS

Para asignar los permisos de acceso a las tablas, ejecute el comando GRANT.

```

001 GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE public.purchases TO ecommerce_admin;
002 GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE public.purchases TO ecommerce_client;
003 GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE public.purchase_items TO
    ecommerce_admin;
005 GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE public.purchase_items TO
    ecommerce_client;
007 GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO ecommerce_admin;
008 GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO ecommerce_client;

```

Archivo org.software.purchase/Purchase.java

```

001 package org.software.purchase;
002
003 import java.sql.Timestamp;
004
005 public class Purchase {
006     long id;
007     long user_id;
008     Timestamp created_at;
009     String created_at_text;
010     Timestamp updated_at;
011
012     public long getId() {
013         return id;
014     }
015     public void setId(long id) {
016         this.id = id;
017     }
018     public long getUser_id() {
019         return user_id;
020     }
021     public void setUser_id(long user_id) {
022         this.user_id = user_id;
023     }
024     public Timestamp getCreated_at() {
025         return created_at;
026     }
027     public void setCreated_at(Timestamp created_at) {
028         this.created_at = created_at;
029     }
030     public Timestamp getUpdated_at() {

```

```

031         return updated_at;
032     }
033
034     public void setUpdated_at(Timestamp updated_at) {
035         this.updated_at = updated_at;
036     }
037     public String getCreated_at_text() {
038         return created_at_text;
039     }
040     public void setCreated_at_text(String created_at_text) {
041         this.created_at_text = created_at_text;
042     }
043 }

```

Archivo org.software.purchase/PurchaseList.java

```

001 package org.software.purchase;
002
003 import java.util.List;
004 import javax.xml.bind.annotation.XmlElement;
005 import javax.xml.bind.annotation.XmlRootElement;
006
007 @XmlRootElement(name = "listing")
008 public class PurchaseList {
009     private List<Purchase> items;
010
011     public PurchaseList() {
012     }
013
014     public PurchaseList(List<Purchase> items) {
015         this.items = items;
016     }
017
018     @XmlElement(name = "data")
019     public List<Purchase> getItems() {
020         return items;
021     }
022 }

```

Archivo org.software.purchase/PurchaseService.java

```

001 package org.software.purchase;
002
003 import java.io.PrintWriter;
004 import java.io.StringReader;
005 import java.sql.Connection;
006 import java.sql.PreparedStatement;
007 import java.sql.ResultSet;
008 import java.sql.Statement;
009 import java.sql.Timestamp;
010 import java.text.SimpleDateFormat;
011 import java.util.ArrayList;
012 import java.util.Map;
013 import java.util.Vector;
014
015 import javax.json.Json;
016 import javax.json.JsonArray;

```

```

017 import javax.json.JsonArrayBuilder;
018 import javax.json.JsonBuilderFactory;
019 import javax.json.JsonObject;
020 import javax.json.JsonObjectBuilder;
021 import javax.json.JsonReader;
022 import javax.servlet.http.HttpServletRequest;
023 import javax.servlet.http.HttpSession;
024 import javax.ws.rs.Consumes;
025 import javax.ws.rs.GET;
026 import javax.ws.rs.POST;
027 import javax.ws.rs.Path;
028 import javax.ws.rs.PathParam;
029 import javax.ws.rs.Produces;
030 import javax.ws.rs.core.Context;
031 import javax.ws.rs.core.Response;
032
033 import org.software.cart.Item;
034 import org.software.cart.ItemList;
035 import org.software.category.Category;
036 import org.software.category.CategoryList;
037 import org.software.product.Product;
038 import org.software.product.ProductList;
039 import org.software.util.DataBase;
040
041 @Path("/purchase")
042 public class PurchaseService {
043
044     @GET
045     @Path("/items/{purchase_id}")
046     @Produces("application/json")
047     public ItemList.getItems(@Context HttpServletRequest request,
048         @PathParam(value = "purchase_id") int purchase_id) {
049         ArrayList lista = new ArrayList();
050
051         DataBase database = new DataBase();
052         Connection connection1 = null;
053         Statement statement1 = null;
054         ResultSet rs1 = null;
055         String sql = "";
056
057         try {
058             connection1 = database.getConnection("client");
059             statement1 = connection1.createStatement();
060
061             HttpSession session = request.getSession();
062             long user_id = getUserId(request);
063
064             if (purchase_id == 0) {
065                 sql = "select id from purchases";
066                 sql += " where user_id = " + user_id;
067                 sql += " order by created_at DESC limit 1";
068                 rs1 = statement1.executeQuery(sql);
069                 if (rs1.next()) {
070                     purchase_id = rs1.getInt("id");
071                 }
072             }
073
074             sql = "select purchase_items.id as item_id, *"
075             sql += " from purchases, purchase_items, products";
076             sql += " where purchases.id = purchase_items.purchase_id";
077             sql += " and purchase_items.product_id = products.id";

```

```

078         sql += " and purchase_id = " + purchase_id;
079
080
081         rs1 = statement1.executeQuery(sql);
082         while (rs1.next()) {
083             long id = rs1.getInt("item_id");
084             double quantity = rs1.getDouble("quantity");
085             double price = rs1.getDouble("price");
086             long product_id = rs1.getLong("product_id");
087             String product_name = rs1.getString("name");
088             String short_description = rs1.getString("short_description");
089             String product_icon = rs1.getString("icon");
090
091             Item item = new Item();
092             item.setId(id);
093             item.setQuantity(quantity);
094             item.setPrice(price);
095             item.setProduct_id(product_id);
096             item.setProduct_name(product_name);
097             item.setProduct_description(short_description);
098             item.setProduct_icon(product_icon);
099
100             lista.add(item);
101         }
102     } catch (Exception e) {
103         System.out.println("Error: " + e.toString());
104     } finally {
105         database.closeObject(rs1);
106         database.closeObject(statement1);
107         database.closeObject(connection1);
108     }
109
110     return new ItemList(lista);
111 }
112
113 @GET
114 @Path("/list")
115 @Produces("application/json")
116 public PurchaseList getPurchases(@Context HttpServletRequest request) {
117     ArrayList lista = new ArrayList();
118
119     HttpSession session = request.getSession();
120     long user_id = getUserId(request);
121
122     DataBase database = new DataBase();
123     Connection connection1 = null;
124     Statement statement1 = null;
125     ResultSet rs1 = null;
126     String sql = "";
127
128     SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");
129
130     try {
131         connection1 = database.getConnection("client");
132         statement1 = connection1.createStatement();
133
134         sql = "select * from purchases";
135         sql += " where user_id = " + user_id;
136         sql += " order by purchases.created_at DESC";
137
138

```

```

139
140
141
142         rs1 = statement1.executeQuery(sql);
143         while (rs1.next()) {
144             long id = rs1.getInt("id");
145             Timestamp created_at = rs1.getTimestamp("created_at");
146             Timestamp updated_at = rs1.getTimestamp("updated_at");
147
148             // String created_at_text = new
149             // SimpleDateFormat("dd/MM/yyyy").format(created_at.getTime());
150             String created_at_text = dateFormat.format(created_at.getTime());
151
152             Purchase purchase = new Purchase();
153             purchase.setId(id);
154             purchase.setUser_id(user_id);
155             purchase.setCreated_at(created_at);
156             purchase.setCreated_at_text(created_at_text);
157
158             lista.add(purchase);
159         }
160     } catch (Exception e) {
161         System.out.println("Error: " + e.toString());
162     } finally {
163         database.closeObject(rs1);
164         database.closeObject(statement1);
165         database.closeObject(connection1);
166     }
167
168     return new PurchaseList(lista);
169 }
170
171 public long getUserId(@Context HttpServletRequest request) {
172     long user_id = -1;
173
174     DataBase database = new DataBase();
175     Connection connection1 = null;
176     Statement statement1 = null;
177     ResultSet rs1 = null;
178     String sql = "";
179
180     String username = "";
181     try {
182         username = request.getUserPrincipal().getName();
183     } catch (Exception e) {
184         username = "";
185     }
186
187     if (username.length() > 0) {
188         try {
189             connection1 = database.getConnection("admin");
190             statement1 = connection1.createStatement();
191
192             sql = "select id from users where username = '" + username + "'";
193
194             rs1 = statement1.executeQuery(sql);
195
196             if (rs1.next()) {
197                 user_id = rs1.getLong("id");
198             }
199

```

```

200         } catch (Exception e) {
201             System.out.println("Error: " + e.toString());
202         } finally {
203             database.closeObject(rs1);
204             database.closeObject(statement1);
205             database.closeObject(connection1);
206         }
207     }
208     return user_id;
209 }
210
211 public int insertPurchase(long user_id) {
212     int purchase_id = 0;
213
214     DataBase database = new DataBase();
215     Connection connection1 = null;
216     ResultSet rs1 = null;
217     PreparedStatement preparedStatement1 = null;
218     String sql = "";
219
220     try {
221         connection1 = database.getConnection("client");
222
223         sql = "INSERT INTO purchases (user_id)";
224         sql = sql + " VALUES (?) RETURNING id";
225
226         preparedStatement1 = connection1.prepareStatement(sql);
227         preparedStatement1.setLong(1, user_id);
228         rs1 = preparedStatement1.executeQuery();
229
230         if (rs1.next()) {
231             purchase_id = rs1.getInt("id");
232         }
233
234     } catch (Exception e) {
235         System.out.println("Error: " + e.toString());
236     } finally {
237         database.closeObject(rs1);
238         database.closeObject(preparedStatement1);
239         database.closeObject(connection1);
240     }
241     return purchase_id;
242 }
243
244 @GET
245 @Path("/checkout")
246 @Produces("application/json")
247 public Response updateItems(@Context HttpServletRequest request) {
248     String message = "";
249     int inserted = 0;
250
251     HttpSession session = request.getSession();
252     long user_id = getUserId(request);
253
254     ArrayList cart = (ArrayList) session.getAttribute("cart");
255
256     if (cart == null) {
257         cart = new ArrayList();
258     }
259 }
260

```

```

261     if (cart.size() > 0) {
262         long purchase_id = insertPurchase(user_id);
263
264         DataBase database = new DataBase();
265         Connection connection1 = null;
266         Statement statement1 = null;
267         String sql = "";
268
269         try {
270             connection1 = database.getConnection("client");
271             statement1 = connection1.createStatement();
272
273             for (int i = 0; i < cart.size(); i = i + 1) {
274                 Item item = (Item) cart.get(i);
275                 long product_id = item.getProduct_id();
276                 double quantity = item.getQuantity();
277                 double price = item.getPrice();
278
279                 sql = "INSERT INTO purchase_items";
280                 sql += " (purchase_id, product_id, price, quantity)";
281                 sql += " VALUES (" + purchase_id + "," + product_id;
282                 sql += "," + price + "," + quantity + ")";
283                 insertedds += statement1.executeUpdate(sql);
284             }
285         } catch (Exception e) {
286             System.out.println("Error: " + e.toString());
287         } finally {
288             database.closeObject(statement1);
289             database.closeObject(connection1);
290         }
291     }
292
293     cart = new ArrayList();
294     session.setAttribute("cart", cart);
295
296     if (inserteds > 0) {
297         message = "{\"success\": \"1\", \"message\": \"Adicionar OK\"}";
298         return Response.status(200).entity(message).build();
299     } else {
300         message = "{\"success\": \"0\", \"message\": \"Error al adicionar\"}";
301         return Response.status(400).entity(message).build();
302     }
303 }
304 }

```

Archivo js/purchase.js

```

001 var data;
002 function getPurchases() {
003     $.getJSON("../ws/purchase/list", function(result) {
004         data = result.data;
005         for (var row = 0; row < data.length; row = row + 1) {
006             var id = data[row].id;
007             var name = data[row].created_at_text;
008             var item_class = "list-group-item";
009             $("#div_purchases").append(
010                 "<a href='javascript:getItems(" + id
011                 + ")';' id='purchase_" + id + "' class='"
012                 + item_class + "'>" + name + " [" + id + "]</a>");

```



```

013     }
014     });
015 }
016
017 function getItem(purchase_id) {
018     $('.list-group-item').removeClass('active').addClass('');
019     $("#purchase_" + purchase_id).addClass('active');
020
021     $.getJSON("../ws/purchase/items/" + purchase_id, function(result) {
022         data = result.data;
023         $('#div_items').empty();
024         var total = 0;
025         for (var row = 0; row < data.length; row = row + 1) {
026             var product_id = data[row].product_id;
027             var name = data[row].product_name;
028             var pricing = data[row].price;
029             var short_description = data[row].product_description;
030             var icon = data[row].product_icon;
031             var quantity = data[row].quantity;
032
033             total = total + pricing * quantity;
034
035             var dp = '';
036             dp += '<div id="row_' + product_id + '" class="row">';
037             dp += '<div class="col-12 col-sm-12 col-md-2 text-center">';
038             dp += '';
040             dp += '</div>';
041             dp += '<div class="col-12 text-sm-center col-sm-12';
042             dp += ' text-md-left col-md-6">';
043             dp += '<h4 class="product-name"><strong>';
044             dp += '<h4>';
045             dp += '<small>' + short_description + '</small>';
046             dp += '</h4>';
047             dp += '</div>';
048
049             dp += '<div class="col-12 col-sm-12 text-sm-center';
050             dp += ' col-md-4 text-md-right row">';
051             dp += '<div class="col-12 col-sm-12 col-md-12';
052             dp += ' text-md-right" style="padding-top: 5px">';
053             dp += '<h6><strong>' + pricing.toFixed(2);
054             dp += ' <span class="text-muted">x</span> ' + quantity + ' = ';
055             dp += ' (pricing * quantity).toFixed(2) + '</strong></h6>';
056             dp += '</div>';
057
058             dp += '</div>';
059             dp += '</div>';
060             dp += '<hr>';
061
062             $("#div_items").append(dp);
063         }
064
065         $("#total").empty();
066         $("#total").append("$" + total.toFixed(2));
067     });
068 }
069

```

Archivo js/cart.js

```
001 var data;
002
003 $("#success-alert").hide();
004 $("#error-alert").hide();
005
006 function getItems(){
007     $.getJSON("../ws/cart/list", function(result){
008         data = result.data;
009         $('#div_items').empty();
010         var total = 0;
011         for(var row=0; row<data.length; row=row+1){
012             var product_id = data[row].product_id;
013             var name = data[row].product_name;
014             var pricing = data[row].price;
015             var short_description = data[row].product_description;
016             var icon = data[row].product_icon;
017             var quantity = data[row].quantity;
018
019             total = total + pricing * quantity;
0120
021             var dp = '';
022             dp += '<div id="row_' + product_id + '" class="row">';
023             dp += '<div class="col-12 col-sm-12 col-md-2 text-center">';
024             dp += '';
026             dp += '</div>';
027             dp += '<div class="col-12 text-sm-center col-sm-12';
028             dp += '" text-md-left col-md-6">';
029             dp += '<h4 class="product-name"><strong>' + name + '</strong></h4>';
030             dp += '<h4>';
031             dp += '<small>' + short_description + '</small>';
032             dp += '</h4>';
033             dp += '</div>';
034             dp += '<div class="col-12 col-sm-12 text-sm-center';
035             dp += '" col-md-4 text-md-right row">';
036             dp += '<div class="col-3 col-sm-3 col-md-6';
037             dp += '" text-md-right" style="padding-top: 5px">';
038             dp += '<h6><strong>' + pricing.toFixed(2);
039             dp += '" <span class="text-muted">x</span></strong></h6>';
040             dp += '</div>';
041             dp += '<div class="col-4 col-sm-4 col-md-4">';
042             dp += '<div class="quantity">';
043             dp += '<input type="button" value="+" class="plus">';
044             dp += '<input type="number" step="1" max="99";
045             dp += '" min="1" value="' + quantity + '" title="Qty";
046             dp += '" class="qty" size="4">';
047             dp += '<input type="button" value="-" class="minus">';
048             dp += '</div>';
049             dp += '</div>';
050             dp += '<div class="col-2 col-sm-2 col-md-2 text-right">';
051             dp += '<a class="nav-link" href="javascript:removeItem(' +
052             dp += 'product_id + ');">';
053             dp += '<button type="button" class="btn btn-outline-danger btn-xs">';
054             dp += '<i class="fa fa-trash" aria-hidden="true"></i>';
055             dp += '</button>';
056             dp += '</a>';
057             dp += '</div>';
058             dp += '</div>';
059             dp += '</div>';
```

```

060         dp += '<hr>';
061         $("#div_items").append(dp);
062     }
063
064     $("#total").empty();
065     $("#total").append("$" + total.toFixed(2));
066
067     var ub = '';
068     ub += '<div class="pull-right">';
069     ub += '<a href="javascript:updateCart();"';
070     ub += ' class="btn btn-outline-secondary pull-right">';
071     ub += 'Update shopping cart';
072     ub += '</a>';
073     ub += '</div>';
074     $("#div_items").append(ub);
075 });
076 }
077
078 function removeItem(product_id){
079     for(var row=0; row<data.length; row=row+1){
080         if(data[row].product_id == product_id){
081             data.splice(row, 1);
082         }
083     }
084     $.getJSON("../ws/cart/del/" + product_id, function(result){
085         getItems();
086         updateItemsCount();
087     });
088 }
089
090 function updateCart(){
091     var allListElements = $(".qty").get();
092     var text = '';
093     var separator = '';
094     for(var i=0; i<allListElements.length; i=i+1){
095         var product_id = data[i].product_id;
096         var quantity = allListElements[i].value;
097         text += separator + '{"product_id":"' + product_id;
098         text += ',"quantity":"' + quantity + '}'';
099         separator = ',';
100     }
101
102     text = '{"data":[' + text + ']}';
103     var posting = $.post("../ws/cart/update/", text);
104
105     // Put the results in a div
106     posting.done(function(result) {
107         var total = 0;
108         for(var i=0; i<allListElements.length; i=i+1){
109             var id = data[i].product_id;
110             var pricing = data[i].price;
111             var quantity = allListElements[i].value;
112             total = total + pricing * quantity;
113         }
114         $('#total').empty();
115         $("#total").append("$" + total.toFixed(2));
116     });
117 }
118
119
120

```

```

121 function checkout(){
122     $.getJSON("../ws/purchase/checkout", function(result){
123         var jsonResult = JSON.stringify(result);
124         var parsed = JSON.parse(jsonResult);
125         var success = parsed.success;
126         var message = parsed.message;
127         var title = "Success !";
128         if(success == false){
129             title = "Error !";
130         }
131
132         if (success == 1){
133             $("#success_title").text(title);
134             $("#success_message").text(message);
135             $("#success-alert").fadeOut(2000, 500).slideUp(500, function() {
136                 $("#success-alert").slideUp(500);
137             });
138         }
139         else{
140             $("#error_title").text(title);
141             $("#error_message").text(message);
142             $("#categoryModal .close").click();
143             $("#error-alert").fadeOut(2000, 500).slideUp(500, function() {
144                 $("#error-alert").slideUp(500);
145             });
146         }
147         $("#checkout").addClass("disabled");
148         updateItemsCount();
149     });
150 }

```

Archivo purchase/index.jsp

```

001 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
002     pageEncoding="ISO-8859-1"%>
003 <!DOCTYPE html>
004 <html lang="en">
005 <head>
006 <meta charset="utf-8">
007 <meta name="viewport"
008     content="width=device-width, initial-scale=1, shrink-to-fit=no">
009 <meta name="description" content="">
010 <meta name="author" content="">
011
012 <title>Shop Homepage - Start Bootstrap Template</title>
013
014 <!-- Bootstrap core CSS -->
015 <link rel="stylesheet"
016     href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
017
018
019 <!-- Custom styles for this template -->
020 <link href="../css/shop-homepage.css" rel="stylesheet">
021
022 </head>
023
024 <body>
025     <!-- Navigation -->
026     <jsp:include page="../portal/menu.jsp"></jsp:include>

```

```

027
028 <!-- Page Content -->
029 <link href="../../css/cart.css" rel="stylesheet">
030 <link
031     href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
032     rel="stylesheet" id="bootstrap-css">
033 <script
034     src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js">
035 </script>
036 <script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
037 <!------- Include the above in your HEAD tag ----->
038
039 <script src="https://use.fontawesome.com/c560c025cf.js"></script>
040
041 <br /> <br /> <br />
042 <div class="container">
043     <div class="row">
044
045         <div class="col-lg-3">
046             <h1 class="my-4">Purchases</h1>
047             <div class="list-group" id="div_purchases"></div>
048         </div>
049         <!-- /.col-lg-3 -->
050
051         <div class="col-lg-9">
052             <div class="card shopping-cart">
053                 <div class="card-header bg-dark text-light">
054                     <i class="fa fa-shopping-cart" aria-hidden="true"></i>
055                     Shopping cart
056                 <div class="clearfix"></div>
057             </div>
058             <div id="div_items" class="card-body">
059             </div>
060             <div class="card-footer">
061                 <div class="coupon col-md-5 col-sm-5 no-padding-left pull-left">
062                     <div class="pull-right" style="margin: 10px">
063                         Coupon code: <b id="coupon"></b>
064                     </div>
065                 </div>
066                 <div class="pull-right" style="margin: 10px">
067                     <div class="pull-right" style="margin: 5px">
068                         Total price: <b id="total">$0.00</b>
069                     </div>
070                 </div>
071             </div>
072         </div>
073
074     </div>
075     <!-- /.col-lg-9 -->
076
077 </div>
078 <!-- /.row -->
079
080 </div>
081 <!-- /.container -->
082
083 <br/> <br/>
084
085 <!-- Footer -->
086 <jsp:include page="../../portal/footer.jsp"></jsp:include>
087

```

```

088
089 <!-- Bootstrap core JavaScript -->
090 <script src="https://code.jquery.com/jquery-3.3.1.js"></script>
091 <script
092     src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js">
093 </script>
094 <script src="../js/portal.js"></script>
095 <script src="../js/purchase.js"></script>
096
097 <script>
098     $(document).ready(function() {
099         getPurchases();
100         getItem(0);
101     });
102 </script>
103
104 </body>
105 </html>

```

Archivo cart/index.jsp

```

001 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
002     pageEncoding="ISO-8859-1"%>
003 <!DOCTYPE html>
004 <html lang="en">
005 <head>
006 <meta charset="utf-8">
007 <meta name="viewport"
008     content="width=device-width, initial-scale=1, shrink-to-fit=no">
009 <meta name="description" content="">
010 <meta name="author" content="">
011
012 <title>Shop Homepage - Start Bootstrap Template</title>
013
014 <!-- Bootstrap core CSS -->
015 <link rel="stylesheet"
016     href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
017
018 <!-- Custom styles for this template -->
019 <link href="../css/shop-homepage.css" rel="stylesheet">
020
021 <link href="../css/cart.css" rel="stylesheet">
022 </head>
023
024 <body>
025 <!-- Navigation -->
026 <jsp:include page="../portal/menu.jsp"></jsp:include>
027
028 <!-- Page Content -->
029 <script src="https://use.fontawesome.com/c560c025cf.js"></script>
030
031 <br />
032 <div class="container">
033     <div class="card shopping-cart">
034         <div class="card-header bg-dark text-light">
035             <i class="fa fa-shopping-cart" aria-hidden="true"></i>
036             Shopping cart
037             <a href=".." class="btn btn-outline-info btn-sm pull-right">
038                 Continue shopping</a>

```

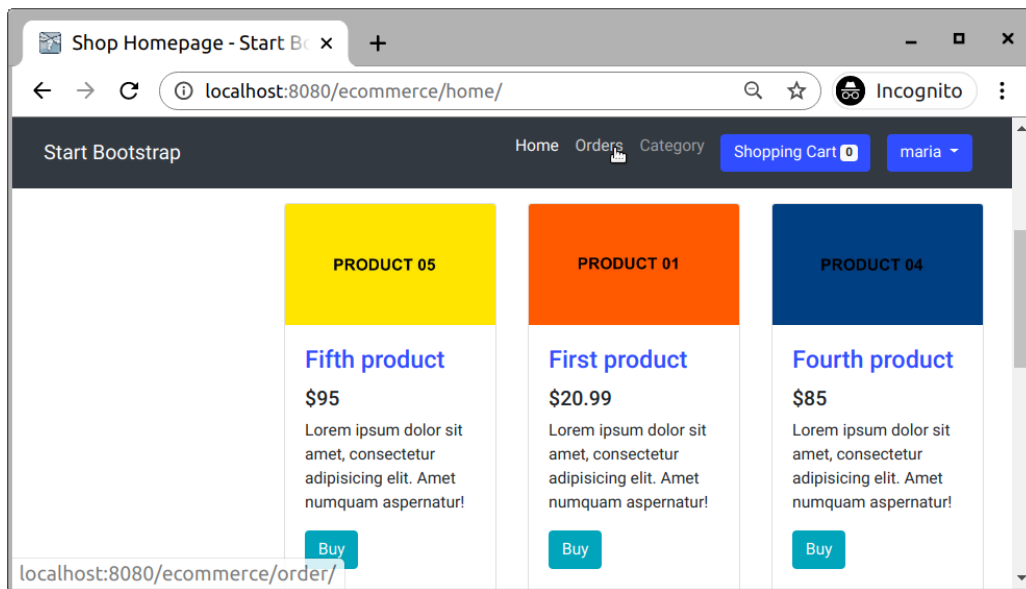
```

039         <div class="clearfix"></div>
040     </div>
041     <div class="alert alert-success" id="success-alert">
042         <strong id="success_title"></strong>
043         <span id="success_message"></span>
044     </div>
045     <div class="alert alert-danger" id="error-alert">
046         <strong id="error_title"></strong>
047         <span id="error_message"></span>
048     </div>
049     <div id="div_items" class="card-body">
050         <div class="pull-right">
051             <a href="" class="btn btn-outline-secondary pull-right">
052                 Update shopping cart </a>
053         </div>
054     </div>
055     <div class="card-footer">
056         <div class="coupon col-md-5 col-sm-5 no-padding-left pull-left">
057             <div class="row">
058                 <div class="col-6">
059                     <input type="text" class="form-control"
060                         placeholder="cupone code">
061                 </div>
062                 <div class="col-6">
063                     <input type="submit" class="btn btn-default"
064                         value="Use cupone">
065                 </div>
066             </div>
067         </div>
068         <div class="pull-right" style="margin: 10px">
069             <a href="javascript:checkout();" id="checkout"
070                 class="btn btn-success pull-right">Checkout</a>
071             <div class="pull-right" style="margin: 5px">
072                 Total price: <b id="total">50.00</b>
073             </div>
074         </div>
075     </div>
076 </div>
077 </div>
078
079 <br /> <br />
080
081 <!-- Footer -->
082 <jsp:include page="../portal/footer.jsp"></jsp:include>
083
084 <!-- Bootstrap core JavaScript -->
085 <script src="https://code.jquery.com/jquery-3.3.1.js"></script>
086 <script
087     src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js">
088 </script>
089 <script src="../js/portal.js"></script>
090 <script src="../js/cart.js"></script>
091 <script>
092     $(document).ready(function() {
093         getItem();
094     });
095 </script>
096
097 </body>
098 </html>

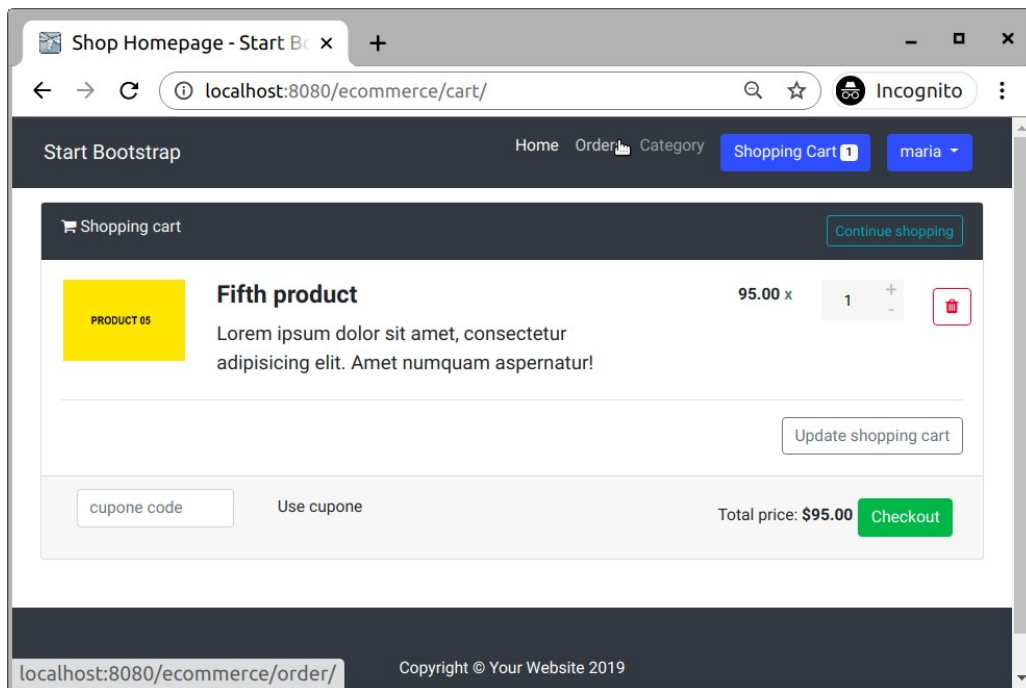
```

4. PRUEBAS

Para realizar una compra el usuario debe estar logueado, seleccionar el producto a comprar y hacer click en el boton [Buy] para adicionar el producto al carro de compras.



Para registrar la compra se debe hacer click en el boton del carro de compras [Shopping Car] y luego hacer click abajo a la derecha en el boton [Checkout].



Para visualizar el historial de compras se debe hacer click arriba en el menú [Order] o en el boton de opciones del perfil del usuario en la opción [Purchases].

