

TIENDA DE COMERCIO ELECTRÓNICO

ITERACIÓN 08 – EVALUACIÓN DE PRODUCTOS Y COMENTARIOS

Autor: Mag. Juan Antonio Castro Silva
Versión: 1.0 Enero 29 de 2020 (20200129T1405)

1. EVALUACION DE PRODUCTOS Y COMENTARIOS

En esta iteración vamos a implementar la funcionalidad que permite evaluar y hacer comentarios sobre los productos comprados.

Tabla reviews:

```
001 CREATE TABLE public.reviews
002 (
003     id bigint NOT NULL DEFAULT nextval('reviews_id_seq'::regclass),
004     product_id bigint NOT NULL,
005     user_id bigint NOT NULL,
006     rating double precision NOT NULL,
007     comment text NOT NULL,
008     approved integer NOT NULL DEFAULT 1,
009     spam integer NOT NULL DEFAULT 0,
010     created_at timestamp without time zone DEFAULT now(),
011     updated_at timestamp without time zone DEFAULT now(),
012     CONSTRAINT reviews_pkey PRIMARY KEY (id)
013 )
014 WITH (
015     OIDS=FALSE
016 );
017 ALTER TABLE public.reviews
018     OWNER TO ecommerce;
019 GRANT ALL ON TABLE public.reviews TO ecommerce;
```

2. ASIGNAR PERMISOS

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

```
001 GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE public.reviews TO ecommerce_admin;
002 GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE public.reviews TO ecommerce_client;
003 GRANT SELECT ON TABLE public.reviews TO ecommerce_guest;
```

3. Components

Para evaluar los productos vamos a utilizar un plugin de jQuery llamado RateIt. (<http://gjunge.github.io/rateit.js/> y <http://gjunge.github.io/rateit.js/examples/>)

3.1 Rating Component (RateIt)



Rating plugin for jQuery. Fast, Progressive enhancement, touch support, customizable (just swap out the images, or change some CSS), Unobtrusive JavaScript (using HTML5 data-* attributes), RTL support, supports as many stars as you'd like, and also any step size.

3.2 Componentes BackEnd

Archivo `org.software.reviews/Review.java`

```
001 package org.software.reviews;
002
003 import java.sql.Timestamp;
004
005 public class Review {
006     long id;
007     long product_id;
008     long user_id;
009     String username;
010     double rating;
011     String comment;
012     int approved;
013     int spam;
014     Timestamp created_at;
015     String created_at_text;
016     Timestamp updated_at;
017
018
019     public long getId() {
020         return id;
021     }
022     public void setId(long id) {
023         this.id = id;
024     }
025     public long getProduct_id() {
026         return product_id;
027     }
028     public void setProduct_id(long product_id) {
029         this.product_id = product_id;
030     }
031     public long getUser_id() {
032         return user_id;
033     }
034 }
```

```

035     public void setUser_id(long user_id) {
036         this.user_id = user_id;
037     }
038     public String getUsername() {
039         return username;
040     }
041     public void setUsername(String username) {
042         this.username = username;
043     }
044     public double getRating() {
045         return rating;
046     }
047     public void setRating(double rating) {
048         this.rating = rating;
049     }
050     public String getComment() {
051         return comment;
052     }
053     public void setComment(String comment) {
054         this.comment = comment;
055     }
056     public int getApproved() {
057         return approved;
058     }
059     public void setApproved(int approved) {
060         this.approved = approved;
061     }
062     public int getSpam() {
063         return spam;
064     }
065     public void setSpam(int spam) {
066         this.spam = spam;
067     }
068     public Timestamp getCreated_at() {
069         return created_at;
070     }
071     public void setCreated_at(Timestamp created_at) {
072         this.created_at = created_at;
073     }
074     public String getCreated_at_text() {
075         return created_at_text;
076     }
077     public void setCreated_at_text(String created_at_text) {
078         this.created_at_text = created_at_text;
079     }
080     public Timestamp getUpdated_at() {
081         return updated_at;
082     }
083     public void setUpdated_at(Timestamp updated_at) {
084         this.updated_at = updated_at;
085     }
086 }

```

Archivo org.software.reviews/ReviewList.java

```
001 package org.software.reviews;
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 ReviewList {
009     private List<Review> items;
010
011     public ReviewList() {
012     }
013
014     public ReviewList(List<Review> items) {
015         this.items = items;
016     }
017
018     @XmlElement(name = "data")
019     public List<Review> getItems() {
020         return items;
021     }
022 }
```

Archivo org.software.reviews/ReviewService.java

```
001 package org.software.reviews;
002
003 import java.sql.Connection;
004 import java.sql.PreparedStatement;
005 import java.sql.ResultSet;
006 import java.sql.Statement;
007 import java.sql.Timestamp;
008 import java.text.SimpleDateFormat;
009 import java.util.ArrayList;
010
011 import javax.servlet.http.HttpServletRequest;
012 import javax.ws.rs.GET;
013 import javax.ws.rs.POST;
014 import javax.ws.rs.Path;
015 import javax.ws.rs.PathParam;
016 import javax.ws.rs.Produces;
017 import javax.ws.rs.core.Context;
018 import javax.ws.rs.core.Response;
019
020 import org.software.purchase.PurchaseService;
021 import org.software.util.DataBase;
022
023 @Path("/review")
024 public class ReviewService {
025
026     @POST
027     @Path("/add")
028     @Produces("application/json")
029     public Response addReview(@Context HttpServletRequest request, Review review) {
030
031         PurchaseService purchaseService = new PurchaseService();
032         long user_id = purchaseService.getUserId(request);
```

```

033
034     DataBase database = new DataBase();
035     Connection connection1 = null;
036     PreparedStatement preparedStatement1 = null;
037     String sql = "";
038     String message = "";
039     int inserted = 0;
040
041     try {
042         connection1 = database.getConnection("client");
043
044         sql = "INSERT INTO reviews (product_id, user_id, rating, comment)";
045         sql += " VALUES (?, ?, ?, ?)";
046
047         preparedStatement1 = connection1.prepareStatement(sql);
048         preparedStatement1.setLong(1, review.getProduct_id());
049         preparedStatement1.setLong(2, user_id);
050         preparedStatement1.setDouble(3, review.getRating());
051         preparedStatement1.setString(4, review.getComment());
052
053         inserted = preparedStatement1.executeUpdate();
054     }
055     catch (Exception e) {
056         System.out.println("Error: " + e.toString());
057     }
058     finally {
059         database.closeObject(preparedStatement1);
060         database.closeObject(connection1);
061     }
062
063     if(inserted > 0){
064         message = "{\"success\":\"1\", \"";
065         message += "\"message\":\"Review inserted successfully\"}";
066         return Response.status(200).entity(message).build();
067     }
068     else{
069         message = "{\"success\":\"0\", \"";
070         message += "\"message\":\"Error inserting review\"}";
071         return Response.status(400).entity(message).build();
072     }
073 }
074
075 @GET
076 @Path("/list/{product_id}")
077 @Produces("application/json")
078 public ReviewList getReviews(@PathParam(value = "product_id") int product_id) {
079     ArrayList reviewList = new ArrayList();
080     DataBase database = new DataBase();
081     Connection connection1 = null;
082     Statement statement1 = null;
083     ResultSet rs1 = null;
084     String sql = "";
085     try {
086         connection1 = database.getConnection("admin");
087         statement1 = connection1.createStatement();
088
089         sql = "select * from reviews, users where reviews.user_id = users.id";
090         sql = sql + " and reviews.product_id = " + product_id;
091         sql = sql + " order by reviews.created_at DESC";

```

```

091         rs1 = statement1.executeQuery(sql);
092         while (rs1.next()) {
093             long id = rs1.getLong("id");
094             long user_id = rs1.getLong("user_id");
095             String username = rs1.getString("username");
096             //long product_id = rs1.getLong("product_id");
097             double rating = rs1.getDouble("rating");
098             String comment = rs1.getString("comment");
099             Timestamp created_at = rs1.getTimestamp("created_at");
100
101             String created_at_text = new
102                 SimpleDateFormat("dd/MM/yyyy").format(created_at.getTime());
103
104             Review review = new Review();
105             review.setId(id);
106             review.setUser_id(user_id);
107             review.setUsername(username);
108             review.setComment(comment);
109             review.setRating(rating);
110             review.setCreated_at(created_at);
111             review.setCreated_at_text(created_at_text);
112             reviewList.add(review);
113         }
114     } catch (Exception e) {
115         System.out.println("Error: " + e.toString());
116     }
117     finally {
118         database.closeObject(rs1);
119         database.closeObject(statement1);
120         database.closeObject(connection1);
121     }
122     return new ReviewList(reviewList);
123 }
124 }

```

3.3 Componentes FrontEnd

Archivo js/item.js

```

001 function getCategoryList(){
002     $.getJSON("../ws/portal/categories", function(result){
003         data = result.data;
004         for(var row=0; row<data.length; row=row+1){
005             var id = data[row].id;
006             var name = data[row].name;
007             var item_class = "list-group-item";
008             $("#div_categories").append(
009                 "<a href='../home/?id=" + id
010                 + "' id='category_" + id + "' class='"
011                 + item_class + "'>" + name + "</a>");
012             }
013     });
014 }
015
016
017
018

```

```

019 function getProductById(product_id){
020     $.getJSON("../ws/product/" + product_id, function(result){
021         data = result.data;
022         for(var row=0; row<data.length; row=row+1){
023             var id = data[row].id;
024             var category_id = data[row].category_id;
025             var name = data[row].name;
026             var pricing = data[row].pricing;
027             var short_description = data[row].short_description;
028             var long_description = data[row].long_description;
029             var icon = data[row].icon;
030             var media = data[row].media;
031
032             $("#product_image").attr('src','../fotos/' + icon);
033             $("#product_name").text(name);
034             $("#product_price").text('$ ' + pricing.toFixed(2));
035             $("#product_description").text(long_description);
036
037             $('#product_rating').rateit('value', media.toFixed(1));
038             $('#starts').text(media.toFixed(1) + " starts");
039
040             $('.list-group-item').removeClass('active').addClass('');
041             $("#category_" + category_id).addClass('active');
042         }
043     });
044 }
045
046 function getReviews(product_id){
047     $.getJSON("../ws/review/list/" + product_id, function(result){
048         data = result.data;
049         $('#div_reviews').empty();
050         for(var row=0; row<data.length; row=row+1){
051             var id = data[row].id;
052             var username = data[row].username;
053             var comment = data[row].comment;
054             var created_at = data[row].created_at;
055             var created_at_text = data[row].created_at_text;
056
057             var review = '';
058             review += '<hr>';
059             review += '<p>' + comment + '</p>';
060             review += '<small class="text-muted">Posted by ' +
061             review += username + ' on ' + created_at_text + '</small>';
062
063             $("#div_reviews").append(review);
064         }
065     });
066 }
067
068 function addReview(product_id){
069     var rating = $('#rating').rateit('value');
070     var comment = document.getElementById("comment").value;
071
072     $.ajax({
073         'url': '../ws/review/add',
074         'method': 'POST',
075         'dataType': 'json',
076         'contentType': 'application/json',
077         'data': JSON.stringify({

```

```

077     "product_id":product_id,
078     "rating":rating,
079     "comment":comment
080   }},
081   'success': function(data){
082     $('#rating').rateit('value', 0.0);
083     $('#rating_value').text('0.0');
084     document.getElementById("comment").value = "";
085     $('#collapseComment').collapse('hide');
086
087     getReviews(product_id);
088     getProductById(product_id);
089
090     $('#div_review_response').empty();
091     var at = '<div class="alert alert-success" role="alert">';
092     at += '<button type="button" class="close">';
093     at += ' data-dismiss="alert" aria-label="Close">';
094     at += '<span aria-hidden="true">&times;</span></button>';
095     at += '<strong>Success!</strong> ' + data.message + '!';
096     at += '</div>';
097     $('#div_review_response').append(at);
098
099     window.setTimeout(function() {
100       $(".alert").fadeTo(500, 0).slideUp(500, function(){
101         $(this).remove();
102       });
103     }, 4000);
104   },
105   'failure': function(errMsg) {
106     alert(errMsg);
107   }
108 });
109 }
110
111 function configureRating(){
112   $("#rating").bind('rated', function (event, value) {
113     $('#rating_value').text(value.toFixed(1));
114   });
115
116   $("#rating").bind('reset', function () {
117     $('#rating_value').text('0.0');
118   });
119
120   $("#rating").bind('over', function (event, value) {
121     if (value != null){
122       $('#rating_value').text(value.toFixed(1));
123     }
124     else {
125       $('#rating_value').text($('#rating').rateit('value').toFixed(1));
126     }
127   });
128 }

```


Archivo item/index.jsp

```
001 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
002     pageEncoding="ISO-8859-1"%>
003 <%
004     String product_id = request.getParameter("id");
005 %>
006 <!DOCTYPE html>
007 <html lang="en">
008 <head>
009 <meta charset="utf-8">
010 <meta name="viewport"
011     content="width=device-width, initial-scale=1, shrink-to-fit=no">
012 <meta name="description" content="">
013 <meta name="author" content="">
014
016 <title>Shop Item - Start Bootstrap Template</title>
017
018 <!-- Bootstrap core CSS -->
019 <!--
020 <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
021 -->
022 <link rel="stylesheet"
023     href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
024
025 <!-- Custom styles for this template -->
026 <link href="../css/shop-item.css" rel="stylesheet">
027
028 <script src="../js/webfont.js"></script>
029 <link href="../css/rateit.css" rel="stylesheet" type="text/css">
030 </head>
031
032 <body>
033     <!-- Navigation -->
034     <jsp:include page="../portal/menu.jsp"></jsp:include>
035
036     <!-- Page Content -->
037     <div class="container">
038
039         <div class="row">
040             <div class="col-lg-3">
041                 <h1 class="my-4">Shop Name</h1>
042                 <div class="list-group" id="div_categories"></div>
043             </div>
044             <!-- /.col-lg-3 -->
045
046             <div class="col-lg-9">
047                 <div class="card mt-4" id="div_product">
048                     <img class="card-img-top img-fluid" id="product_image"
049                         src="" alt="">
050                     <div class="card-body">
051                         <h3 class="card-title" id="product_name"></h3>
052                         <h4 id="product_price"></h4>
053                         <p class="card-text" id="product_description"></p>
054                         <a href="javascript:addToCart(<%= product_id %>)"
055                             class="btn btn-info" role="button">Buy</a>
056                     <br/><br/>
057                     <div>
058                         <span class="rateit" id="product_rating"
059                             data-rateit-readonly="true"></span>
060                         <span id="starts"></span>
```

```

061         </div>
062     </div>
063 </div>
064 <!-- /.card -->
065
066 <div class="card card-outline-secondary my-4">
067 <div class="card-header">Product Reviews</div>
068 <div class="card-body" id="div_reviews2">
069     <p>
070         <a class="btn btn-success float-right" data-toggle="collapse"
071             href="#collapseComment" role="button"
072             aria-expanded="false"
073             aria-controls="#collapseComment"> Leave a Review </a>
074     </p>
075     <div class="clearfix"></div>
076     <div id="div_review_response"></div>
077     <div class="collapse" id="collapseComment">
078         <br />
079         <div class="card card-body">
080             <form>
081                 <div class="form-group">
082                     <label for="comment">Comment</label>
083                     <textarea class="form-control"
084                         id="comment" rows="3"></textarea>
085                 </div>
086                 <div class="form-group">
087                     <label for="rating">Rating</label>
088                     <div>
089                         <span class="rateit" id="rating"
090                             data-rateit-step="0.1"
091                             data-rateit-min="0"
092                             data-rateit-max="5"></span>
093                         <span id="rating_value"></span>
094                     </div>
095                 </div>
096                 <a href="javascript:addReview(<%=product_id%>);"
097                     class="btn btn-info float-right" role="button">Save</a>
098             </form>
099         </div>
100     </div>
101     <div id="div_reviews"></div>
102 </div>
103 </div>
104 <!-- /.card -->
105
106 </div>
107 <!-- /.col-lg-9 -->
108 </div>
109 </div>
110 <!-- /.container -->
111
112 <!-- Footer -->
113 <jsp:include page="../portal/footer.jsp"></jsp:include>
114
115
116 <!-- Bootstrap core JavaScript -->
117 <script src="https://code.jquery.com/jquery-3.3.1.js"></script>
118 <script
119 src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js">
120 </script>
121

```

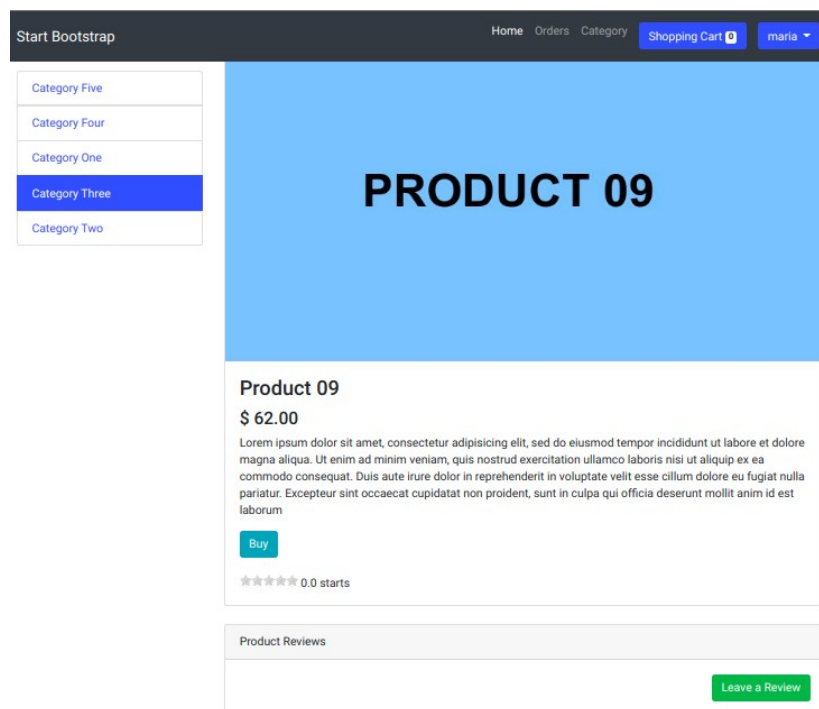
```

122 <script src="../../js/jquery.rateit.js" type="text/javascript"></script>
123 <script src="../../js/portal.js" type="text/javascript"></script>
124 <script src="../../js/item.js" type="text/javascript"></script>
125
126 <script type="text/javascript">
127     $(document).ready(function() {
128         var product_id = <%=product_id%>;
129         getCategoryList();
130         getProductById(product_id);
131         getReviews(product_id);
132         configureRating();
133         updateItemsCount();
134     });
135 </script>
136 </body>
137 </html>

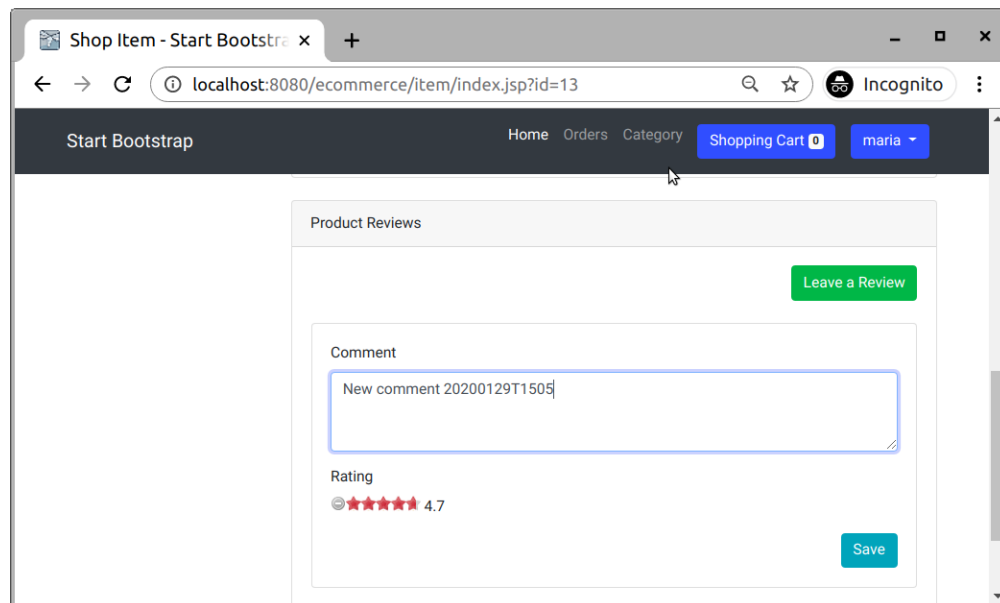
```

4. PRUEBAS

Para evaluar y comentar un producto, seleccione el producto (Item) y haga click abajo en el botón [Leave a Review].



Digite el comentario y seleccione el valor de la evaluación (Rating) y luego haga click en el botón [Save].



Al seleccionar un producto evaluado se puede ver el promedio de la evaluación y el listado de los comentarios realizados.

