

## Sprint 5

PanelApp: plataforma digital disponible en la web para optimizar la comercialización directa de panela entre pequeños productores y cliente final

Presentado por:

Karen Cuéllar  
David Gallo  
Juan Carlos Jerez  
Erika Juliana Rodríguez  
Diego Rojas Forero

## Desarrollo Sprint #5

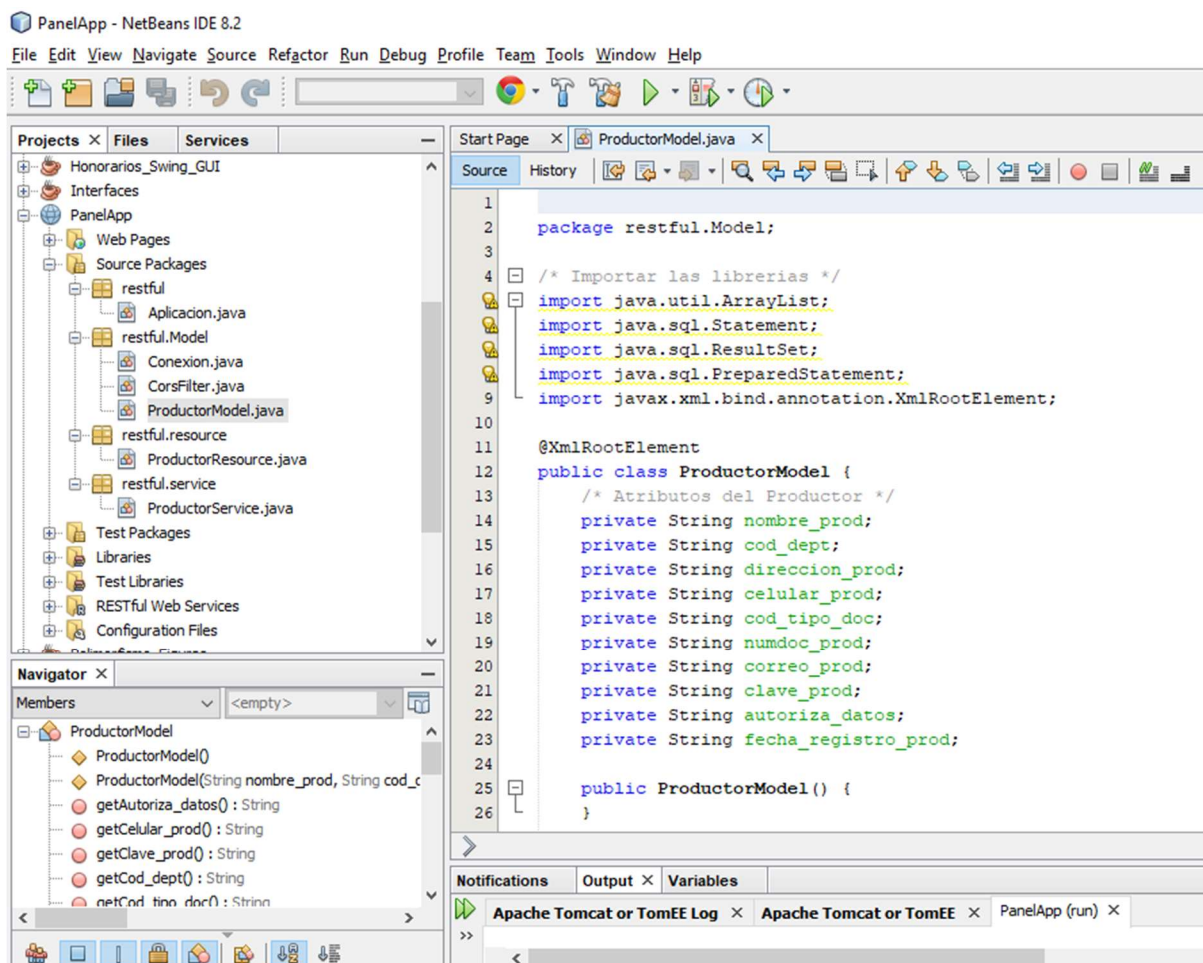
Bienvenido al Sprint 5 del ciclo Desarrollo de Software

### BACKEND - FRONTEND

Para el Sprint 5 se debe entregar como evidencia

1. Aplicación BACKEND donde se desarrolle un CRUD de una de las tablas de la base de datos.

Se muestra la estructura del BackEnd con la construcción de los servicios, recursos y el ProductorModel



Conexión a la base de datos de PanelApp

```

1 package restful.Model;
2
3
4
5 import java.sql.Connection;
6 import java.sql.DriverManager;
7
8 public class Conexion {
9     private Connection con;
10    private static final String DB_DRIVER = "com.mysql.cj.jdbc.Driver";
11    private static final String HOST = "localhost:3306";
12    private static final String DB = "panelapp";
13    private static final String URL = "jdbc:mysql://" + HOST + "/" + DB;
14    private static final String USERNAME = "root";
15    private static final String PASSWORD = "1234";
16
17    public Conexion() {
18        try {
19            Class.forName(DB_DRIVER);
20            con = DriverManager.getConnection(URL, USERNAME, PASSWORD);
21            System.out.println("Conexión exitosa");
22        } catch (Exception e) {
23            System.out.println("Error de conexión");
24        }
25    }
26

```

Servicio web con el CRUD para los servicios de GET, POST, PUT y DELETE

```

1 package restful.service;
2
3
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.Statement;
7 import java.sql.SQLException;
8 import java.util.ArrayList;
9 import restful.Model.ProductorModel;
10 import restful.Model.Conexion;
11
12 public class ProductorService {
13
14    public ArrayList<ProductorModel> getProductores() {
15        ArrayList<ProductorModel> lista = new ArrayList<>();
16        Conexion conn = new Conexion();
17        String sql = "SELECT * FROM productor";
18
19        try {
20            Statement stm = conn.getCon().createStatement();
21            ResultSet rs = stm.executeQuery(sql);
22            while (rs.next()) {
23                ProductorModel productor = new ProductorModel();
24                productor.setNombre_prod(rs.getString("Nombre_prod"));
25                productor.setCod_dept(rs.getString("Cod_dept"));
26                productor.setDireccion_prod(rs.getString("Direccion_prod"));

```

```

Start Page x ProductorModel.java x Conexion.java x ProductorService.java x
Source History
41
42 public ProductorModel getProductor(String numdoc_prod) {
43     ProductorModel productor = new ProductorModel();
44     Conexion conex = new Conexion();
45     String Sql = "SELECT * FROM productor WHERE numdoc_prod = ?";
46
47     try {
48
49         PreparedStatement pstm = conex.getCon().prepareStatement(Sql);
50         pstm.setString(1, numdoc_prod);
51         ResultSet rs = pstm.executeQuery();
52
53         while (rs.next()) {
54
55             productor.setNumdoc_prod(rs.getString("numdoc_prod"));
56             productor.setNombre_prod(rs.getString("nombre_prod"));
57             productor.setCod_dept(rs.getString("cod_dept"));
58             productor.setDireccion_prod(rs.getString("direccion_prod"));
59             productor.setCelular_prod(rs.getString("celular_prod"));

```

```

Start Page x ProductorModel.java x Conexion.java x ProductorService.java x
Source History
69
70     return productor;
71 }
72
73 //Insertar un nuevo Productor
74 public ProductorModel addProductor(ProductorModel productor) {
75     Conexion conex = new Conexion();
76     String Sql = "INSERT INTO productor(numdoc_prod,nombre_prod,direccion_prod,celular_prod,cod_dept)";
77     Sql = Sql + "values (?, ?, ?, ?, ?)";
78
79     try {
80         PreparedStatement pstm = conex.getCon().prepareStatement(Sql);
81         pstm.setString(1, productor.getNumdoc_prod());
82         pstm.setString(2, productor.getNombre_prod());
83         pstm.setString(3, productor.getDireccion_prod());
84         pstm.setString(4, productor.getCelular_prod());
85         pstm.setString(5, productor.getCod_dept());
86         pstm.executeUpdate();
87
88     } catch (SQLException e) {
89         System.out.println(e);
90         return null;
91     }
92     return productor;
93 }

```

```

Start Page x ProductorModel.java x Conexion.java x ProductorService.java x
Source History
93 }
94
95 //Borrar un productor de la base de datos
96 public String delProductor(String numdoc_prod) {
97     Conexion conn = new Conexion();
98
99     String sql = "DELETE FROM productor WHERE numdoc_prod= ?";
100     try {
101         PreparedStatement pstm = conn.getCon().prepareStatement(sql);
102         pstm.setString(1, numdoc_prod);
103         pstm.executeUpdate();
104     } catch (SQLException excepcion) {
105         System.out.println("Ha ocurrido un error al eliminar " + excepcion.getMessage());
106         return "\\\"Accion\\\":\\\"Error\\\"";
107     }
108     return "\\\"Accion\\\":\\\"Registro Borrado\\\"";
109 }

```

```

Start Page x ProductorModel.java x Conexion.java x ProductorService.java x
Source History
109 }
110
111 public ProductorModel updateProductor(ProductorModel productor) {
112     Conexion conn = new Conexion();
113     String sql = "UPDATE productor SET nombre_prod=?, celular_prod=?, direccion_prod=?, cod_dept=? WHERE numdoc_prod="
114     try {
115         PreparedStatement pstm = conn.getCon().prepareStatement(sql);
116         pstm.setString(1, productor.getNombre_prod());
117         pstm.setString(2, productor.getCelular_prod());
118         pstm.setString(3, productor.getDireccion_prod());
119         pstm.setString(4, productor.getCod_dept());
120         pstm.setString(5, productor.getNumdoc_prod());
121         pstm.executeUpdate();
122     } catch (SQLException excepcion) {
123         System.out.println("Ha ocurrido un error al actualizar " + excepcion.getMessage());
124         return null;
125     }
126     return productor;
127 }
128
129 }

```

Creación del Path de PanelApp



```

Start Page X ProductorModel.java X Conexion.java X ProductorService.java X ProductorResource.java X
Source History
21 import javax.ws.rs.PUT;
22
23 /**
24  *
25  * @author SENA
26  */
27 @Path("productor")
28 @Produces(MediaType.APPLICATION_JSON)
29 @Consumes(MediaType.APPLICATION_JSON)
30
31
32 public class ProductorResource {
33
34     ProductorService servicio = new ProductorService();
35
36     // Este método retorna el listado completo de los Productores en formato JSON
37     // El path a usar es http://server/PanelApp/api/productor
38     @GET
39     public ArrayList<ProductorModel> getProductores() {
40
41         return servicio.getProductores();
42     }

```

```

Start Page X ProductorModel.java X Conexion.java X ProductorService.java X ProductorResource.java X
Source History
44 // Este método retorna la información para un Productor particular en formato JSON.
45 // Se debe pasar como parámetro el numero de documento del productor
46 // El path a usar es http://server/PanelApp/api/productor/12345678
47 @Path("/{numdoc_prod}")
48 @GET
49 public ProductorModel getProductor(@PathParam("numdoc_prod") String id) {
50
51     return servicio.getProductor(id);
52 }

```

```

Start Page X ProductorModel.java X Conexion.java X ProductorService.java X ProductorResource.java X
Source History
54 // Este método INSERTA la información para un Productor particular en la base de datos.
55 // Se debe pasar como parámetro en formato JSON la información del Productor, así:
56 /*
57 {
58     "celular_prod": "3201234567",
59     "cod_dept": "25",
60     "direccion_prod": "Cas 45 Bucaramanga",
61     "nombre_prod": "Sergio Espinosa",
62     "numdoc_prod": "357989"
63 }
64 */
65 // El path a usar es http://server/PanelApp/api/productor/
66 // Se puede probar en Postman seleccionando el método POST, BODY y JSON
67 @POST
68 public ProductorModel addProductor(String JSON) {
69     GsonBuilder builder = new GsonBuilder();
70     builder.setPrettyPrinting();
71     Gson gson = builder.create();
72     ProductorModel productor = gson.fromJson(JSON, ProductorModel.class);
73     return servicio.addProductor(productor);
74 }

```

```

Start Page x ProductorModel.java x Conexion.java x ProductorService.java x ProductorResource.java x
Source History
76 // Este método BORRA la información para un Productor particular de la base de datos.
77 // Se debe pasar como parámetro el número del documento del Productor.
78 // El path a usar es http://server/PanelApp/api/productor/12345678
79 // Se puede probar en Postman seleccionando el método DELETE, Send
80 @DELETE
81 @Path("/{numdoc_prod}")
82 public String delProductor(@PathParam("numdoc_prod") String numdoc_prod) {
83
84     return servicio.delProductor(numdoc_prod);
85
86 }
87
88 // Este método ACTUALIZA la información para un Productor particular de la base de datos.
89 // Se debe pasar como parámetro el número del documento del Productor.
90 // El path a usar es http://server/PanelApp/api/productor/12345678
91 // Se puede probar en Postman seleccionando el método DELETE, Send
92 @PUT
93 public ProductorModel updateProductor(String JSON) {
94     GsonBuilder builder = new GsonBuilder();
95     builder.setPrettyPrinting();
96     Gson gson = builder.create();
97     ProductorModel productor = gson.fromJson(JSON, ProductorModel.class);
98     return servicio.updateProductor(productor);
99 }
100
101 }

```

## 2. Aplicación FRONTEND.

El Front-End muestra los principales datos del Productor de Panela, utilizando Ajax para consumir los servicios del Back-End

C:\User\Jcerez\Estudio\Mision2022-MinTIC\C3-G34-DESARROLLO DE SOFTWARE\Sprint5\FrontEnd\_PanelApp\crud-productor.html - Notepad++

```

Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
new 1  new 3  Datos.txt  Polaris_tablas.sql  Crear_tablas.sql  new 7  new 6  crud-productor.html

1  <!DOCTYPE html lang="es">
2  <html>
3  <head>
4      <title>CRUD del Productor - PanelApp</title>
5      <meta charset="utf-8"></meta>
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <meta name="description" content="PanelApp"></meta>
8      <meta name="author" content="David Alejandro Gallo Quintero"></meta>
9      <meta name="author" content="Equipo #4"></meta>
10     <meta name="description" content="Formulario de registro de productor de panela"></meta>
11     <meta name="keywords" content="Panela, PanelApp, producción de panela, productor, registro"></meta>
12     <LINK REL=StyleSheet HREF="css/estilo-panelapp.css" TYPE="text/css" MEDIA=screen>
13     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity=
        "sha384-KyZXEAg3QhqlMpg8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZ0J3BCsw2P0p/We" crossorigin="anonymous">
14     <link rel="stylesheet" href="css/alertify.min.css" />
15     <figure>
16         
17     </figure>
18 </head>
19 <body>
20     <div class="container mt-4 shadow-lg p-3 mb-5 rounded">
21         <button id="btnCrear" type="button" class="btn btn-primary">Crear Productor de Panela</button>
22
23         <table id="pedidos" class="table mt-2 table-bordered table-striped">
24             <thead>
25                 <tr class="text-center" style="background-color: #01422C; color: white;">
26                     <th>Num. Documento</th>
27                     <th>Nombre del Productor</th>
28                     <th>Dirección del Productor</th>
29                     <th>Celular del Productor</th>
30                     <th>Cód. Ciudad</th>
31                     <th>Acciones</th>
32                 </tr>
33             </thead>
34             <tbody>
35
36
37
38
39
40         <div class="modal fade" id="modalProductor" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
41             <div class="modal-dialog">
42                 <div class="modal-content">
43                     <div class="modal-header bg-primary text-white">
44                         <h5 class="modal-title" id="exampleModalLabel">Productor</h5>
45                         <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
46                     </div>
47                     <div class="modal-body">
48                         <form>
49                             <div class="mb-3">
50                                 <label for="id" class="col-form-label">Num. Documento :</label>
51                                 <input id="num_documento" type="number" placeholder="Digite el número de documento" class="form-control" id="recipient-name"
52                                     required autocomplete="off" oninvalid="setCustomValidity('Digite un Número de Documento')">
53                                 <input type="text" value="setCustomValidity('')">
54                             </div>
55                             <div class="mb-3">
56                                 <label for="nombre" class="col-form-label">Nombre Productor :</label>
57                                 <input id="nombre" placeholder="Digite el nombre" type="text" class="form-control" id="recipient-name"
58                                     required autocomplete="off" oninvalid="setCustomValidity('Digite el Nombre')">
59                                 <input type="text" value="setCustomValidity('')">
60                             </div>
61                             <div class="mb-3">
62                                 <label for="direccion" class="col-form-label">Direccion Productor :</label>
63                                 <input id="direccion" placeholder="Digite la Dirección" type="text" class="form-control"
64                                     id="recipient-name" required autocomplete="off" oninvalid="setCustomValidity('Digite la Direccion')">
65                                 <input type="text" value="setCustomValidity('')">
66                             </div>
67                             <div class="mb-3">
68                                 <label for="celular" class="col-form-label">Celular del Productor :</label>
69                                 <input id="celular" placeholder="Digite el Celular" type="number" class="form-control"
70                                     id="recipient-name" required autocomplete="off" oninvalid="setCustomValidity('Digite el Celular')">
71                                 <input type="text" value="setCustomValidity('')">
72                             </div>
73                             <div class="mb-3">
74                                 <label for="ciudad" class="col-form-label">Ciudad Productor :</label>
75                                 <input id="ciudad" placeholder="Digite la Ciudad" type="text" class="form-control" id="recipient-name"

```



```

68 <label for="celular" class="col-form-label">Celular del Productor :</label>
69 <input id="celular" placeholder="Digite el Celular" type="number" class="form-control"
70 id="recipient-name" required autocomplete="off" oninvalid="setCustomValidity('Digite el Celular')\"
71 oninput="setCustomValidity('')\">
72 </div>
73 <div class="mb-3">
74 <label for="ciudad" class="col-form-label">Ciudad Productor :</label>
75 <input id="ciudad" placeholder="Digite la Ciudad" type="text" class="form-control" id="recipient-name"
76 required autocomplete="off" oninvalid="setCustomValidity('Digite la Ciudad')\"
77 oninput="setCustomValidity('')\">
78 </div>
79 <div class="modal-footer">
80 <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>
81 <button type="submit" class="btn btn-primary">Guardar</button>
82 </div>
83 </form>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89
90 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
91 integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXAkRfj\"
92 crossorigin="anonymous"></script>
93 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
94 integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpUGGGon+Y7RswenRtN/tE3MoK7ZeZDyx\"
95 crossorigin="anonymous"></script>
96 <script src="js/alertify.min.js"></script>
97 <script src="js/productor.js"></script>
98
99
100 <div id="footer">Equipo de Trabajo #4 - Programa MisiónTIC 2022 - Octubre de 2021</div>
101 </div>
102 </body>
103 </html>

```

La siguiente imagen muestra el contenido del PRODUCTOR.JS que contiene las funciones para consumir los servicios del backEnd:

```

1 //const url = "http://localhost:8080/PanelApp/api/productor"
2 const url = "http://localhost:8080/PanelApp/api/productor"
3 const contenedor = document.querySelector('tbody')
4 let resultados = ''
5 const modalProductor = new bootstrap.Modal(document.getElementById('modalProductor'))
6 const formProductor = document.querySelector('form')
7 const nombreProductor = document.getElementById('nombre')
8 const direccionProductor = document.getElementById('direccion')
9 const celularProductor = document.getElementById('celular')
10 const ciudadProductor = document.getElementById('ciudad')
11 const NumDocumentoProductor = document.getElementById('num_documento')
12 let opcion = ''
13 btnCrear.addEventListener('click', () => {
14     nombreProductor.value = ''
15     direccionProductor.value = ''
16     celularProductor.value = ''
17     ciudadProductor.value = ''
18     NumDocumentoProductor.value = ''
19     NumDocumentoProductor.disabled = false
20     modalProductor.show()
21     opcion = 'crear'
22 })
23 const ajax = (options) => {
24     let { url, method, success, error, data } = options;
25     const xhr = new XMLHttpRequest();
26
27     console.log(options);
28
29     xhr.addEventListener("readystatechange", (e) => {
30         if (xhr.readyState !== 4) return;
31
32         if (xhr.status >= 200 && xhr.status < 300) {
33             let json = JSON.parse(xhr.responseText);
34             success(json);
35         } else {
36             let message = xhr.statusText || "Ocurrió un error";
37             error(`Error ${xhr.status}: ${message}`);
38         }
39     });

```

```

36 let message = xhr.statusText || "Ocurrió un error";
37 error(Error(xhr.statusText): $message);
38
39 });
40
41 xhr.open(method || "GET", url);
42 xhr.setRequestHeader("Content-type", "application/json; charset=utf-8");
43 xhr.send(JSON.stringify(data));
44
45 const getAll = () => {
46   ajax({
47     url: url,
48     method: "GET",
49     success: (res) => {
50       console.log(res);
51
52       res.forEach((productor) => {
53         resultados += `<tr>
54           <td width="10%">${productor.numdoc_prod}</td>
55           <td width="15%">${productor.nombre_prod}</td>
56           <td width="25%">${productor.direccion_prod}</td>
57           <td width="15%">${productor.celular_prod}</td>
58           <td width="15%">${productor.cod_dept}</td>
59           <td class="text-center" width="20%"><a class="btnEditar btn btn-primary">Editar</a><a class="btnBorrar btn btn-danger">Borrar</a></td>
60         </tr>
61       `);
62     },
63     error: (err) => {
64       console.log(err);
65       $table.insertAdjacentHTML("afterend", `<p><b>{err}</b></p>`);
66     },
67   });
68
69 document.addEventListener("DOMContentLoaded", getAll);
70
71
72 document.addEventListener("click", (e) => {
73
74   if (e.target.matches(".btnBorrar")) {
75     const fila = e.target.parentNode.parentNode
76     const id = fila.firstElementChild.innerHTML
77     console.log(id)
78     alertify.confirm(`¿Estás seguro de eliminar el num_documento ${id}?`,
79       function () {
80         ajax({
81           url: url + "/" + id,
82           method: "DELETE",
83           headers: {
84             'Content-Type': 'application/json'
85           },
86           success: (res) => location.reload(),
87           error: (err) => alert(err),
88         });
89         alertify.success('Registro eliminado')
90       },
91       function () {
92         alertify.error('Cancel')
93       });
94   }
95
96   if (e.target.matches(".btnEditar")) {
97     const fila = e.target.parentNode.parentNode
98     NumDocumentoProductor.value = fila.children[0].innerHTML
99     NombreProductor.value = fila.children[1].innerHTML
100     direccionProductor.value = fila.children[2].innerHTML
101     celularProductor.value = fila.children[3].innerHTML
102     ciudadProductor.value = fila.children[4].innerHTML
103     NumDocumentoProductor.disabled = true
104     opcion = 'editar'
105     modalProductor.show()
106   }
107 })
108
109
110 nombreProductor.value = fila.children[1].innerHTML
111 direccionProductor.value = fila.children[2].innerHTML
112 celularProductor.value = fila.children[3].innerHTML
113 ciudadProductor.value = fila.children[4].innerHTML
114 NumDocumentoProductor.disabled = true
115 opcion = 'editar'
116 modalProductor.show()
117
118
119 formProductor.addEventListener('submit', (e) => {
120   e.preventDefault()
121   let metodo = "POST"
122   if (opcion == 'editar') {
123     metodo = "PUT"
124   }
125   ajax({
126     url: url,
127     method: metodo,
128     headers: {
129       'Content-Type': 'application/json'
130     },
131     success: (res) => location.reload(),
132     error: (err) => {
133       $form.insertAdjacentHTML("afterend", `<p><b>{err}</b></p>`);
134     },
135     data: {
136       "cod_dept": ciudadProductor.value,
137       "direccion_prod": direccionProductor.value,
138       "numdoc_prod": NumDocumentoProductor.value,
139       "nombre_prod": nombreProductor.value,
140       "celular_prod": celularProductor.value
141     },
142   });
143   modalProductor.hide()
144 })
145

```

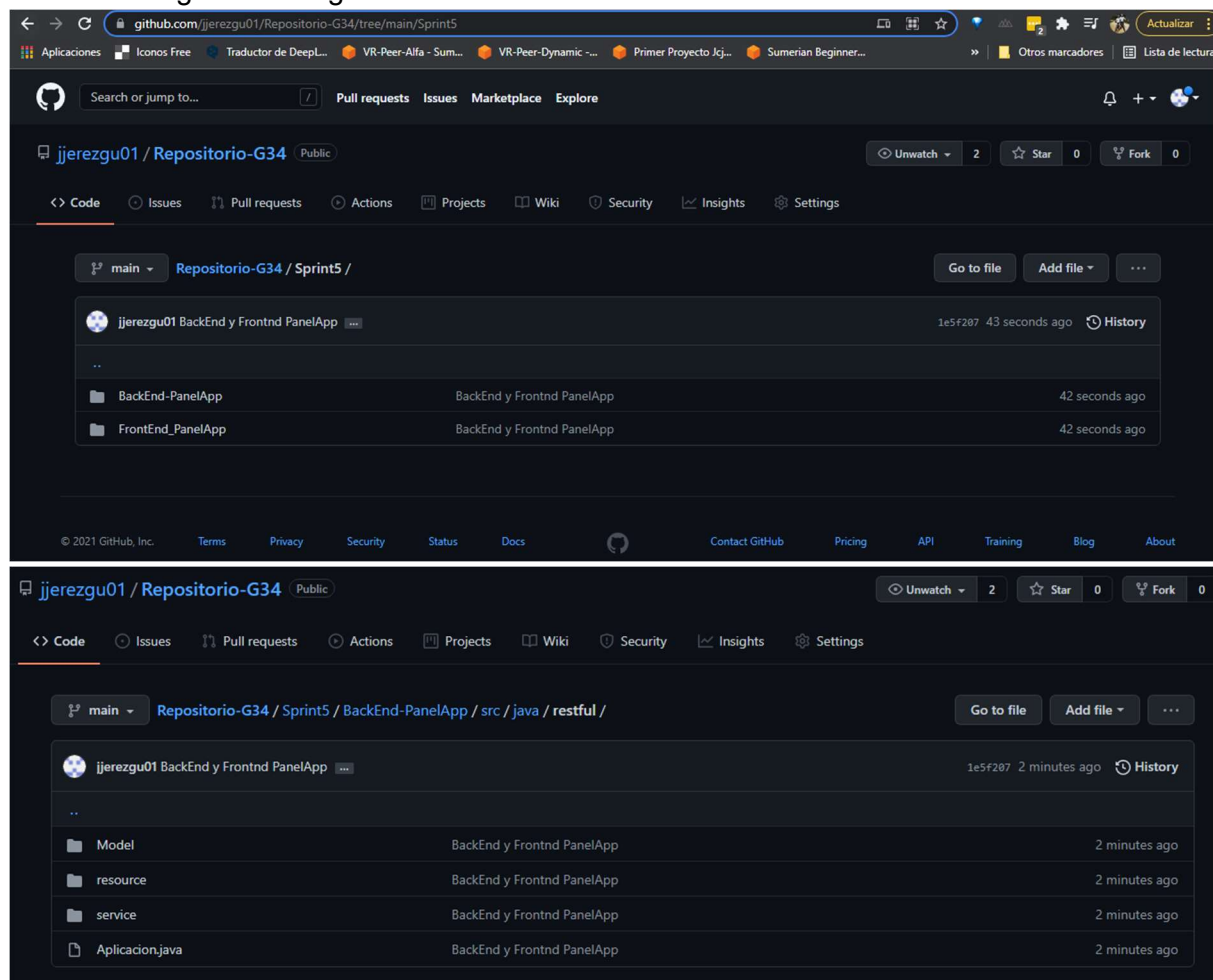
3. Documento en formato PDF donde se presenten pantallazos con las siguientes evidencias:

- a. Pantallazos con la publicación del BACKEND y FRONTEND en el repositorio de código GitHub.

Ruta del GitHub:

<https://github.com/jjerezgu01/Repositorio-G34/tree/main/Sprint5>

Al GitHub se sube el código fuente para el BackEnd y el FronEnd de PanelApp, como lo muestra la siguiente imagen:



- b. Pantallazos con la ejecución del BACKEND, incluyendo el consumo de recursos GET, POST, PUT y DELETE que se pueden verificar con el software POSTMAN.

## Recurso GET

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

My Workspace

New Import

Overview

GET http://localhost:8080/PanelApp/api/productor

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 227 ms Size: 1.9 KB

Save Response

```

1 {
2   "autoriza_datos": "S",
3   "celular_prod": "3235245622",
4   "clave_prod": "kazeng",
5   "cod_dept": "11",
6   "cod_tipo_doc": "1",
7   "correo_prod": "kguzman@gmail.com",
8   "direccion_prod": "Cll 80 No. 118 - 85",
9   "fecha_registro_prod": "2021-10-27",
10  "nombre_prod": "Karen Guzman",
11 }

```

## Recurso POST

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

My Workspace

New Import

Overview

POST http://localhost:8080/PanelApp/api/productor

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Body

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

8 {
9   "direccion_prod": "Cll 80 No. 60 - 63",
10  "fecha_registro_prod": "2021-10-29",
11  "nombre_prod": "Sofia Cuellar",
12  "numdoc_prod": "32478188"
13 }

```

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 227 ms Size: 1.9 KB

Save Response

```

15 {
16   "autoriza_datos": "S",
17   "celular_prod": "3235245688",
18   "clave_prod": "sofiac",
19   "cod_dept": "8",
20   "cod_tipo_doc": "1",
21   "correo_prod": "scuellar@gmail.com",
22   "direccion_prod": "Cll 80 No. 60 - 63",
23   "fecha_registro_prod": "2021-10-29",
24   "nombre_prod": "Sofia Cuellar",
25   "numdoc_prod": "32478188"
26 }

```



MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Crear\_tablas\* Poblar\_producto\* departamento productor

Schemas

Filter objects

panelapp

Tables

departamento

productor

tipodocumento

Views

Stored Procedures

Functions

recurso\_docente

sakila

sys

world

Administration Schemas

Information

No object selected

Object Info Session

Result Grid

Filter Rows:

Edit Export/Import Wrap Cell Content: 15

nombre_prod	cod_dept	direccion_prod	celular_prod	cod_tipo_doc	numdoc_prod	correo_prod	clave_prod	foto_doc_prod	autoriza_datos	fecha_registro_prod
Karen Guzman	11	Cll 80 No. 110 - 05	3235245622	1	32478125	kguzman@gmail.com	kareng	XXXX	S	2021-10-27
Sofia Cuellar	8	Cll 80 No. 50 - 63	3235245688	1	32478188	scuellar@gmail.com	sofiac	XXXX	S	2021-10-29
Manuel Arias	5	Cll 50 No. 45 - 115	3215698740	1	35874569	carmen25@gmail.com	carmenita	XXXX	S	2021-10-27
Pedro Perez	15	Cll 30 No. 82 - 46	3045627945	1	42859147	manuel.arias@hotmail.com	manuel1975	XXXX	S	2021-10-27
Mariana Iglesias	47	Cra 2 No. 25 - 13	3005871230	3	900589652-1	productores@gmail.com	pedro123	XXXX	S	2021-10-27
	13	Cll 45 No. 64 - 15	3112566879	3	900874568-9	marianag@gmail.com	mangles	XXXX	S	2021-10-27

productor 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:54:26	SELECT * FROM panelapp.productor LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

## Recurso PUT

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Upgrade

My Workspace

New Import

Overview PUT http://localhost:80...

No Environment

http://localhost:8080/PanelApp/api/productor

PUT http://localhost:8080/PanelApp/api/productor

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Send

Body

```

8 ..... "direccion_prod": "Cll 80 No. 50 - 63",
9 ..... "fecha_registro_prod": "2021-10-29",
10 ..... "nombre_prod": "Karen Cuellar",
11 ..... "numdoc_prod": "32478188"
12

```

Body Cookies Headers (7) Test Results 32478125

Status: 200 OK Time: 187 ms Size: 561 B Save Response

Pretty Raw Preview Visualize JSON

```

1
2 {
3   "autoriza_datos": "S",
4   "celular_prod": "3235245688",
5   "clave_prod": "kareng",
6   "cod_dept": "8",
7   "cod_tipo_doc": "1",
8   "correo_prod": "kguzman@gmail.com",
9   "direccion_prod": "Cll 80 No. 50 - 63",
10  "fecha_registro_prod": "2021-10-29",
11  "nombre_prod": "Karen Cuellar",
12  "numdoc_prod": "32478188"

```

Find and Replace Console

Bootcamp Runner Trash

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Filter objects

panelapp

Tables

departamento

productor

tipodocumento

Views

Stored Procedures

Functions

recurso\_docente

sakila

sys

world

Administration Schemas

Information

No object selected

Object Info Session

Crear\_tablas\* Poblar productor\* departamento productor productor x

1 • SELECT \* FROM panelapp.productor;

Result Grid

nombre_prod	cod_dept	direccion_prod	celular_prod	cod_tipo_doc	numdoc_prod	correo_prod	clave_prod	foto_doc_prod	autoriza_datos	fecha_registro_prod
Karen Guzman	11	Cil 80 No. 110 - 05	3235245622	1	32478125	kguzman@gmail.com	kareng	0000	S	2021-10-27
Karen Cuellar	8	Cil 80 No. 50 - 63	3235245688	1	32478188	kcuelar@gmail.com	karenc	0000	S	2021-10-29
Carmen Silva	5	Cil 50 No. 45 - 115	3215698740	1	35874569	carmen25@gmail.com	carmenita	0000	S	2021-10-27
Manuel Arias	15	Cil 30 No. 82 - 46	3045627945	1	42899147	manuel.arias@hotmail.com	manuel1975	0000	S	2021-10-27
Pedro Perez	47	Cra 2 No. 25 - 13	3005871230	3	900589652-1	productores@gmail.com	pedro123	0000	S	2021-10-27
Mariana Iglesias	13	Cil 45 No. 64 - 15	3112566879	3	900874568-9	marianai@gmail.com	mangles	0000	S	2021-10-27

productor 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:54:26	SELECT * FROM panelapp.productor LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
2	18:17:05	SELECT * FROM panelapp.productor LIMIT 0, 1000	6 row(s) returned	0.047 sec / 0.000 sec

## Recurso DELETE

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

My Workspace

New Import

Overview

DEL http://localhost:80...

No Environment

http://localhost:8080/PanelApp/api/productor/32478188

DELETE http://localhost:8080/PanelApp/api/productor/32478188

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 341 ms Size: 320 B

Save Response

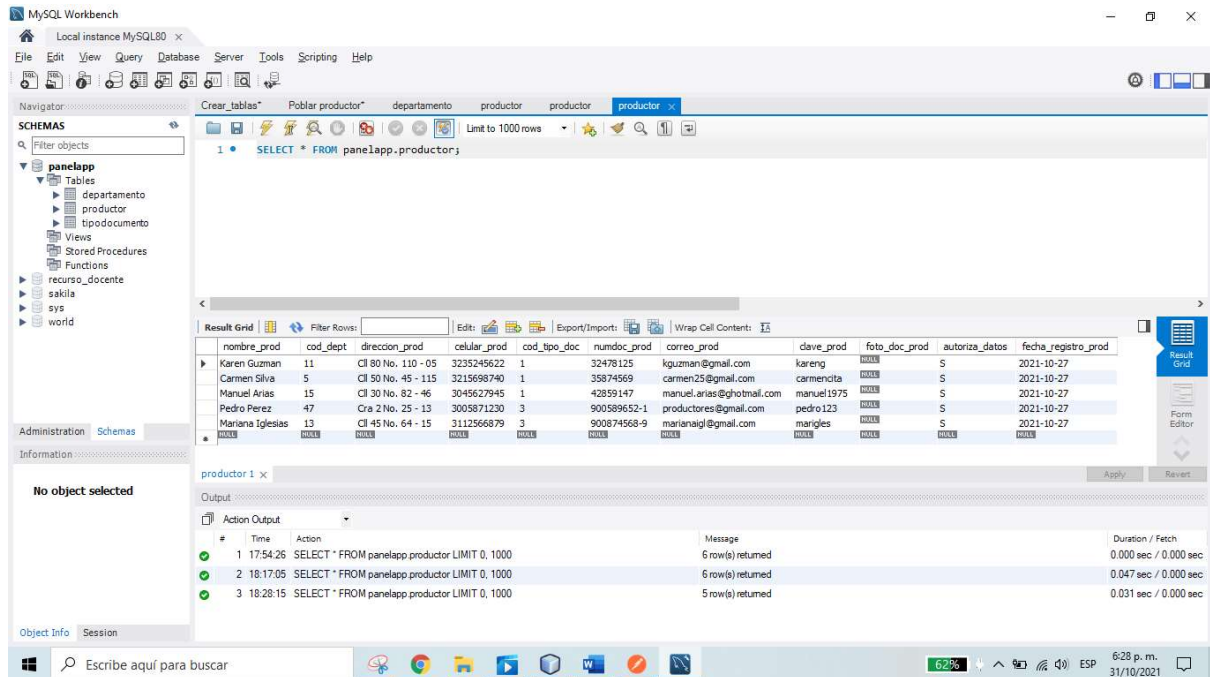
Pretty Raw Preview Visualize JSON

```

1  "Action": "Registro Borrado"
2
3
  
```

Find and Replace Console

Bootcamp Runner Trash



- c. Pantallazos de ejecución del FRONTEND, donde se visualice las operaciones de incluir, modificar y eliminar.

La siguiente imagen muestra la pantalla del CRUD del Productor de Panela dentro de la aplicación de PanelApp



Crear Productor					
Num. Documento	Nombre del Productor	Dirección del Productor	Celular del Productor	Cód. Ciudad	Acciones
12309876	David	Medellín	45678923	05	<a href="#">Editar</a> <a href="#">Borrar</a>
1236789	Karen	Medellín	314897867	05	<a href="#">Editar</a> <a href="#">Borrar</a>
2468	Martin Jerez Updated	Cra 34 este	3108110283	25	<a href="#">Editar</a> <a href="#">Borrar</a>
357989	Sergio Espinosa	Cas 45 Bucaramanga	3201234567	25	<a href="#">Editar</a> <a href="#">Borrar</a>
481216	nombre html	direccion html updated	1000000000	25	<a href="#">Editar</a> <a href="#">Borrar</a>
78091	nOMBRE 2	direccione	3203040	25	<a href="#">Editar</a> <a href="#">Borrar</a>
794567823	Un tal Diego	Bogotá	313456789	11	<a href="#">Editar</a> <a href="#">Borrar</a>
87654321	Juliana Rodriguez	Cra 7 # 152-50	320202020	25	<a href="#">Editar</a> <a href="#">Borrar</a>

La siguiente imagen muestra la ventana Modal para crear Productores.

**Productor**

Num. Documento :  
Digite el número de documento

Nombre Productor :  
Digite el nombre

Direccion Productor :  
Digite la Dirección

Celular del Productor :  
Digite el Celular

Ciudad Productor :  
Digite la Ciudad

Cerrar Guardar

Num. Documento	Nombre del Productor
2468	Martin Jerez Updated
357989	Sergio Espinosa
481216	nombre html
78091	nOMBRE 2
87654321	Juliana Rodríguez

La siguiente imagen muestra la ventana Modal para editar la información de un productor

**Productor**

Num. Documento :  
2468

Nombre Productor :  
Martin Jerez Updated

Direccion Productor :  
Cra 34 este


Celular del Productor :  
3108110283

Ciudad Productor :  
25

Num. Documento	Nombre del Productor
2468	Martin Jerez Updated
357989	Sergio Espinosa
481216	nombre html
78091	nOMBRE 2
87654321	Juliana Rodríguez

La siguiente imagen muestra la confirmación que pide el Front-End antes de borrar un registro de la base de datos.





Crear Productor

AlertifyJS

¿Estás seguro de eliminar el num\_documento 2468?

OK Cancel

Num. Documento	Nombre del Productor	Ciudad	Acciones	
2468	Martin Jerez Updated		Editar	Borrar
357989	Sergio Espinosa	Cas 45 Bucaramanga	3201234567	25
481216	nombre html	dirección html updated	1000000000	25
78091	nOMBRE 2	direccione	3203040	25
87654321	Juliana Rodriguez	Cra 7 # 152-50	320202020	25

Equipo de Trabajo #4 - Programa MisiónTIC 2022 - Octubre de 2021

NOTA: Cada integrante de los equipos de trabajo deben subir la evidencia en su plataforma.