

Universidad Interamericana de Puerto Rico, Arecibo

Sistema de Gestión de Inventario para Tienda

Gabriel A. Del Olmo Reyes

Kenix B. Zamot Colón

Johnnathan De Jesús Calderín

COMP 3400

Software Engineering

Prof. Dasta

Tabla de Contenido:

Resumen del Proyecto:

Objetivos del Proyecto:

- Creación una base de datos para el inventario de una tienda.
- Creación una conexión entre la base de datos y el Flask (funciones de CRUD).
- Creación una conexión entre el Flask y el Front End.

Lista de los miembros del equipo:

- Gabriel A. Del Olmo Reyes – Product Owner
- Kenix B. Zamot Colón – Developer
- Johnnathan De Jesús Calderín – Scrum Master

Product Log:

- Crear de la base de datos para el inventario de una tienda.
- Crear la conexión entre el Flask (funciones de CRUD) y la base de datos.
- Crear la conexión entre el Front End (js) y el Flask.

Work Breakdown Structure (WBS):

- Gabriel A. Del Olmo Reyes – Flask
- Kenix B. Zamot Colón – Front End
- Johnnathan De Jesús Calderín – Base de datos

Diseños del Proyecto:

- **Diagrama ER**
- **Diagrama de la Arquitectura del Software**

- **Código**

Back End:

App.py

```
from flask import Flask, jsonify, request
```

```
import mariadb
```

```
import sys
```

```
#import db access config
```

```
from config import DATABASE_CONFIG
```

```
app = Flask(__name__)
```

```
try:
```

```
    conn = mariadb.connect(**DATABASE_CONFIG)
```

```
except mariadb.Error as e:
```

```
    print(f"Error on connection: {e}")
```

```
    sys.exit(1)
```

```
cursor = conn.cursor()
```

```
#En esta direccion saldra mi nombre
```

```
@app.route('/mis_datos', methods=['GET'])
```

```
def mis_datos():
```

```
return jsonify({'datos': 'Gabriel Del Olmo'})
```

#En esta direccion se obtendra la informacion para llenar la tabla de records

```
@app.route('/get_items', methods=['GET'])
```

```
def get_items():
```

```
    cursor.execute("SELECT * FROM Product")
```

```
    items = cursor.fetchall()
```

```
    list = []
```

```
    for item in items:
```

```
        list.append({
```

```
            "UPC":item[0],
```

```
            "name_Product":item[1],
```

```
            "price_Product":item[2],
```

```
            "quantity_Product":item[3]
```

```
        })
```

Convertir los resultados en un formato más amigable o devolverlos directamente

```
response = jsonify({"data":list})
```

```
response.headers.add("Content-type",'application/json')
```

```
response.headers.add('Access-Control-Allow-Origin', '*')
```

```
return response
```

#En esta direccion se obtendra la informacion para llenar la tabla de items

```
@app.route('/get_product_detail', methods=['GET'])
```

```
def get_detail():
```

```
    cursor.execute("SELECT * FROM Detail_Product")
```

```
    details = cursor.fetchall()
```

```
    list = []
```

```
    for detail in details:
```

```
        list.append({
```

```
            "UPC":detail[0],
```

```
            "category_Product":detail[1],
```

```
            "brand":detail[2],
```

```
            "batch":detail[4]
```

```
        })
```

```
# Convertir los resultados en un formato más amigable o devolverlos directamente
```

```
response = jsonify({"data":list})
```

```
response.headers.add("Content-type",'application/json')
```

```
response.headers.add('Access-Control-Allow-Origin', '*')
```

```
return response
```

```
#En esta direccion se obtendra la informacion para llenar la tabla de providers
```

```
@app.route('/get_providers', methods=['GET'])
```

```
def get_providers():
```

```
    cursor.execute("SELECT * FROM Provider")
```

```
providers = cursor.fetchall()
```

```
list = []
```

```
for provider in providers:
```

```
    list.append({
```

```
        "id_Provider":provider[0],
```

```
        "name_Provider":provider[1],
```

```
        "contact":provider[2]
```

```
    })
```

```
# Convertir los resultados en un formato más amigable o devolverlos directamente
```

```
response = jsonify({"data":list})
```

```
response.headers.add("Content-type",'application/json')
```

```
response.headers.add('Access-Control-Allow-Origin', '*')
```

```
return response
```

```
#Insertar un nuevo item en el DB
```

```
@app.route('/insert_item', methods=['POST'])
```

```
def new_item():
```

```
    datos = request.json
```

```
    item_UPC = datos.get('UPC')
```

```
    item_name = datos.get('name_Product')
```

```
    item_price = datos.get('price_Product')
```

```
    item_quantity = datos.get('quantity_Product')
```

```

strQry = 'insert into Product'

strQry += "(UPC, , name_Product, price_Product, quantity_Product)"

strQry += f"values ({'item_UPC'},{'item_name'},{'item_price'},{'item_quantity'})"


cursor.execute(strQry)

conn.commit()

response = {"message": "Record inserted"}

return jsonify(response) , 200

```

#Modificar un item

```

@app.route('/update_item', methods=['POST'])

def update_item():

    datos = request.json

    item_UPC = datos.get('UPC')

    item_name = datos.get('name_Product')

    item_price = datos.get('price_Product')

    item_quantity = datos.get('quantity_Product')


    strQry = 'update Product '

    strQry += f"set name_Product = '{item_name}', "

    strQry += f"price_Product = '{item_price}', "

    strQry += f"quantity_Product = '{item_quantity}', "

    strQry += f"where item_UPC = {UPC} "

```

```
cursor.execute(strQry)
```

```
conn.commit()
```

```
response = {"message": "Record updated"}
```

```
return jsonify(response), 200
```

```
#Eliminar Item
```

```
@app.route('/delete_item/<string:UPC>', methods=['DELETE'])
```

```
def del_item(UPC):
```

```
    strQry = 'DELETE FROM Product WHERE UPC = ?' #eliminar donde item donde  
    el upc sea..
```

```
    cursor.execute(strQry, (UPC,))
```

```
    conn.commit()
```

```
    return jsonify({"message": "Item deleted successfully"}), 200
```

```
if __name__ == '__main__':
```

```
    app.run(host='0.0.0.0', port=5000)
```

Config.py

```
DATABASE_CONFIG = {  
    'host': 'linux-vm2.arecibo.inter.edu',  
    'port': 3306,  
    'user': 'tiendamanager',  
    'password': 'tiendasecreta',  
    'database': 'InventoryDB'  
}
```

Front End:

Style.css

```
*{
margin: 0;
padding: 0;
}

body{
background-color: antiquewhite;
}

nav{
width: 100%;
```

```
height: 100px;  
background-color:whitesmoke ;  
}
```

```
ul {  
nav-index: auto;  
margin-left: 100px;  
}
```

```
ul li {  
list-style: style none; ;  
display: inline-block;  
line-height:100px ;  
}
```

```
ul li a{  
display: block;  
text-decoration: none;  
font-family: "Roboto", sans-serif;  
font-weight: 700;  
font-style: normal;  
padding: 0 150px;  
}
```

```
.fondo{  
    background-image: url("gestion.jpg");  
    background-size: cover;  
    min-height: 500px;  
    height: 100vh;  
}
```

```
ul li a:hover{  
    color: rgb(16, 89, 190);  
}
```

```
ul li{  
    text-decoration: none;  
    float:left;  
}
```

Records.html

```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
  <meta charset="UTF-8" />  
  <title>User List</title>  
  <link href="https://cdn.datatables.net/1.11.3/css/jquery.dataTables.min.css" rel="stylesheet" />  
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet" />  
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdn.datatables.net/1.11.3/js/jquery.dataTables.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<style>
  body {
    background-color: black;
  }

  .container {
    background-color: #f5f5dc;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  }

  table {
    border-collapse: collapse;
  }

  table, th, td {
    border: 1px solid #ccc;
  }

  th {
    background-color: #fff8dc;
    color: #333;
  }

  th, td {
    padding: 8px;
    text-align: left;
  }

  tbody tr:hover {
    background-color: #f0ead6;
  }

  h1 {
    color: black;
  }
</style>
</head>

<body>
  <div class="container mt-5">
    <header class="background">
      <h2>Records</h2>
      <table id="tablaDetail" class="display" style="width: 100%">
```

```

<thead>
  <tr>
    <th>UPC</th>
    <th>Name Product</th>
    <th>Product Price</th>
    <th>Product Quantity</th>
    <th class="text-center">
      <button type="button" id="btn-add-new" class="btn btn-primary btn-sm">
        Create
      </button>
    </th>
  </tr>
</thead>
</table>
</header>
</div>

<!-- Modal for Add, Edit, and Delete -->
<div class="modal" id="actionModal">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="modalTitle">Action</h5>
        <button type="button" class="close" data-dismiss="modal">
          &times;
        </button>
      </div>
      <div class="modal-body">
        <form id="myForm">
          <input type="hidden" name="user_id" id="user_id">
          <div class="mb-2">
            <label for="upc" class="form-label">UPC</label>
            <input type="text" class="form-control" name="upc" id="name" placeholder="UPC" />
          </div>
          <div class="mb-2">
            <label for="productname" class="form-label">Product name</label>
            <input type="text" class="form-control" name="product name" id="product name" placeholder="Product
name" />
          </div>
          <div class="mb-2">
            <label for="product price" class="form-label">Product Price</label>
            <input type="product price" class="form-control" name="product price" id="product price"
placeholder="Product Price" />
          </div>
          <div class="mb-2">
            <label for="product quantity" class="form-label">Product Quantity</label>
            <input type="product quantity" class="form-control" name="product quantity" id="product quantity"
placeholder="Product Quantity" />

```

```

        </div>
        <div class="mb-2">
            <label for="role" class="form-label">Role</label>
            <select name="role" id="role" class="form-control">
                <option value="teacher">Usuario</option>
                <option value="student">Proveedor</option>
            </select>
        </div>
    </form>
    <div id="modalMessages"></div>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-dismiss="modal">
        Close
    </button>
    <button type="button" id="btn-modal-crud-action" class="btn" data-dismiss="modal">
        Action
    </button>
</div>
</div>
</div>
</div>

<script src=".js/users.js"></script>
<script src=".js/functions-tools.js"></script>
</body>
</html>

```

Providers.html

```

<!DOCTYPE html>
<html lang="es">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Lista de Datos Providers</title>
        <!-- Bootstrap CSS -->
        <link
            href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
            rel="stylesheet"
        />
        <!-- DataTables CSS -->
        <link
            href="https://cdn.datatables.net/1.10.24/css/jquery.dataTables.min.css"
            rel="stylesheet"/>
    </head>
    <body>
        <div class="container">
            <div class="row">
                <div class="col-12">
                    <div class="card">
                        <div class="card-header">
                            <h3>Lista de Datos Providers</h3>
                        </div>
                        <div class="card-body">
                            <table>
                                <thead>
                                    <tr>
                                        <th>ID</th>
                                        <th>Nombre</th>
                                        <th>Apellido</th>
                                        <th>Email</th>
                                        <th>Telefono</th>
                                        <th>Rol</th>
                                    </tr>
                                </thead>
                                <tbody>
                                    <tr>
                                        <td>1</td>
                                        <td>Juan</td>
                                        <td>Perez</td>
                                        <td>juan.perez@example.com</td>
                                        <td>1234567890</td>
                                        <td>Usuario</td>
                                    </tr>
                                    <tr>
                                        <td>2</td>
                                        <td>Maria</td>
                                        <td>Garcia</td>
                                        <td>maria.garcia@example.com</td>
                                        <td>9876543210</td>
                                        <td>Proveedor</td>
                                    </tr>
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </body>
</html>

```

```

<style>
  body {
    background-color: black;
  }

  .container {
    background-color: #f5f5dc;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  }

  table {
    border-collapse: collapse;
  }

  table, th, td {
    border: 1px solid #ccc;
  }

  th {
    background-color: #fff8dc;
    color: #333;
  }

  th, td {
    padding: 8px;
    text-align: left;
  }

  tbody tr:hover {
    background-color: #f0ead6;
  }

  h1 {
    color: black;
  }
</style>

</head>
<body>

<div class="container mt-5">
  <h1>Providers</h1>
  <table id="tablaProviders" class="table">
    <thead>

```

```

        <tr>
            <th>id_Provider</th>
            <th>name_Provider</th>
            <th>contact</th>
        </tr>
    </thead>
    <tbody>
        <!-- Aquí se llenará la tabla con datos -->
    </tbody>
</table>
</div>

<!-- jQuery y Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.js"></script>
<!-- DataTables JS -->
<script src="https://cdn.datatables.net/1.10.24/js/jquery.dataTables.min.js"></script>
<!-- Script personalizado -->
<script src="./js/script.js"></script>
</body>
</html>

```

Item.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Items</title>
    <!-- Bootstrap CSS -->
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet" />
    <!-- DataTables CSS -->
    <link href="https://cdn.datatables.net/1.10.24/css/jquery.dataTables.min.css" rel="stylesheet" />
    <style>
        .background {
            background-color: black;
        }

        .container {
            background-color: #f5f5dc;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }
    </style>

```



```

    }

    table {
        border-collapse: collapse;
    }

    table, th, td {
        border: 1px solid #ccc;
    }

    th {
        background-color: #fff8dc;
        color: #333;
    }

    th, td {
        padding: 8px;
        text-align: left;
    }

    tbody tr:hover {
        background-color: #f0ead6;
    }

    h1 {
        color: black;
    }
</style>
</head>

<body class="background">
    <div class="container mt-5">
        <h1>Product Data List</h1>
        <table id="tblalItems" class="table">
            <thead>
                <tr>
                    <th>UPC</th>
                    <th>Category Product</th>
                    <th>Brand</th>
                    <th>Batch</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>123456789012</td>
                    <td>Beverages</td>
                    <td>Coca-Cola</td>
                    <td>Batch A1</td>
                </tr>
            </tbody>
        </table>
    </div>
</body>
</html>

```

```

        </tr>
        <tr>
            <td>987654321098</td>
            <td>Snacks</td>
            <td>Lays</td>
            <td>Batch B2</td>
        </tr>
        <tr>
            <td>456789123456</td>
            <td>Dairy</td>
            <td>Yoplait</td>
            <td>Batch C3</td>
        </tr>
        <!-- Table data will be populated here -->
    </tbody>
</table>
</div>

<!-- jQuery and Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.js"></script>
<!-- DataTables JS -->
<script src="https://cdn.datatables.net/1.10.24/js/jquery.dataTables.min.js"></script>
<!-- Custom Script -->
<script src="./js/script.js"></script>
</body>

</html>

```

Index.html

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Página de Inicio</title>
    <link rel="stylesheet" href="style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">
</head>
<body>
    <div class="fondo">
        <nav class="navbar">

```

```
<ul class="nav-list">
  <li><a href='items.html'>Items</a></li>
  <li><a href='providers.html'>Providers</a></li>
  <li><a href='records.html'>Ingreso de Registros</a></li>

</ul>
</nav>
</div>
</body>
</html>
```

Users.js

```
$(document).ready(function () {

  let crudAction = "edit";

  var table = $("#tablaDetail").DataTable({

    ajax: {

      url: "http://0.tcp.ngrok.io:16590/get_items", // Change this to your API URL

    },

    columns: [

      { data: "UPC" },

      { data: "name_Product" },

      { data: "price_Product" },

      { data: "quantity_Product" },

      {

        data: null,

        defaultContent:
```

```
"<div class='text-center'>" +  
  
    "<button type='button' class='btn btn-info btn-sm btnEdit'><i class='bi bi-  
pencil'></i></button>&nbsp;" +  
  
    "<button type='button' class='btn btn-danger btn-sm btnDelete'><i class='bi bi-  
trash3'></i></button>" +  
  
    "</div>",  
  
    },  
  
    ],  
  
});
```

```
$("#btn-add-new").click(function () {
```

```
    crudAction = "new";
```

```
    $("#form").get(0).reset();
```

```
    $("#UPC").prop("readonly", false);
```

```
    $("#name_Product").prop("readonly", false);
```

```
    $("#price_Product").prop("readonly", false);
```

```
    $("#quantity_Product").prop("readonly", false);
```

```
    updateModal("Add Item");
```

```
});
```

```
$("#tablaDetail tbody").on("click", ".btnEdit", function () {
```

```
var data = table.row($(this).parents("tr")).data(); // Get ROW DATA in TABLE
```

```
setDataElementsReadOnly(data, false);
```

```
fillForm(data);
```

```
crudAction = "edit";
```

```
updateModal("Edit Item");
```

```
});
```

```
$("#tablaDetail tbody").on("click", ".btnDelete", function () {
```

```
var data = table.row($(this).parents("tr")).data(); // Get ROW DATA in TABLE
```

```
setDataElementsReadOnly(data, true);
```

```
fillForm(data);
```

```
crudAction = "delete";
```

```
$("#modalTitle").text("Delete Item");
```

```
$("#modalMessages").html(
```

```
'<div class="alert alert-danger" role="alert">' +
```

```
'Are you sure you want to delete the item <strong>' + data.UPC + '</strong>?' +
```

```
"</div>"
```

```
);
```

```
$("#btn-modal-crud-action").addClass("btn-danger");

$("#btn-modal-crud-action").removeClass("btn-primary");

$("#btn-modal-crud-action").text("Delete");


$("#actionModal").modal("show");

});
```

```
$("#btn-modal-crud-action").click(function () {

    let url, method, data;

    if (crudAction === "new") {

        url = "http://0.tcp.ngrok.io:16590/insert_item";

        method = "POST";

        data = {

            UPC: $("#UPC").val(),

            name_Product: $("#name_Product").val(),

            price_Product: $("#price_Product").val(),

            quantity_Product: $("#quantity_Product").val(),

        };

        // Execute REST API to INSERT

    } else if (crudAction === "edit") {

        url = "http://0.tcp.ngrok.io:16590/update_item";

        method = "POST";

        data = {
```

```

    UPC: $("#UPC").val(),
    name_Product: $("#name_Product").val(),
    price_Product: $("#price_Product").val(),
    quantity_Product: $("#quantity_Product").val(),
};

// Execute REST API to UPDATE

$.ajax({
    url: url,
    method: method,
    data: JSON.stringify(data),
    contentType: "application/json",
    success: function (response) {
        table.ajax.reload();
        $("#actionModal").modal("hide");
    },
    error: function (error) {
        console.error("Error updating item:", error);
    },
});

} else {

    var UPC = $("#UPC").val(); // Get the UPC from the form

    $.ajax({

```

```
url: `http://0.tcp.ngrok.io:16590/delete_item/${UPC}`, // Adjust the URL as needed

method: "DELETE",

success: function (response) {

    table.ajax.reload();

    $("#actionModal").modal("hide");

},

error: function (error) {

    console.error("Error deleting item:", error);

},

});

}
```

```
function updateModal(title) {

    $("#modalTitle").text(title);

    $("#btn-modal-crud-action").addClass("btn-primary");

    $("#btn-modal-crud-action").removeClass("btn-danger");

    $("#btn-modal-crud-action").text("Save");

    $("#actionModal").modal("show");

}
```



```
function setDataElementsReadOnly(data, isReadOnly) {

    $("#UPC").prop("readonly", isReadOnly);

    $("#name_Product").prop("readonly", isReadOnly);

    $("#price_Product").prop("readonly", isReadOnly);

    $("#quantity_Product").prop("readonly", isReadOnly);

}
```

```
function fillForm(data) {

    $("#UPC").val(data.UPC);

    $("#name_Product").val(data.name_Product);

    $("#price_Product").val(data.price_Product);

    $("#quantity_Product").val(data.quantity_Product);

}

});
```

Scripts.js

```
$(document).ready(function () {
    // llenar la tabla de contenidos con el api de la maquina virtual para obtener items en la tabla
    $("#tblaItems").DataTable({
        ajax: {
            url: "http://0.tcp.ngrok.io:16590/get_product_detail",
        },
        error: function (xhr, status, error) {
            console.log("Error al obtener los datos del API:", error);
        },
        columns: [
            { data: "UPC" },
            { data: "category_Product" },
            { data: "brand" },
```

```

    { data: "batch" },
  ],
});
// llenar la tabla de contenidos con el api de la maquina virtual para obtener proveedores en la tabla
$("#tablaProviders").DataTable({
  ajax: {
    url: "http://0.tcp.ngrok.io:16590/get_providers",
  },
  error: function (xhr, status, error) {
    console.log("Error al obtener los datos del API.", error);
  },
  columns: [
    { data: "id_Provider" },
    { data: "name_Provider" },
    { data: "contact" },
  ],
});
})

```

Function-tools.js

// formData is a JSON Object with all values from the REST API

```

function fillForm(formData) {

  Object.keys(formData).forEach(function(key) {

    var element = document.querySelector('[name="' + key + "']");

    if (element) {

      if (element.type === 'checkbox' || element.type === 'radio') {

        element.checked = element.value === formData[key].toString();

      } else if (element.tagName.toLowerCase() === 'select') {

        element.value = formData[key];

      } else {

        element.value = formData[key];

      }

    }

  });
}

```

```
}  
});  
}
```

```
function setDataElementsReadOnly(formData, isReadOnly) {  
  Object.keys(formData).forEach(function(key) {  
    var element = document.querySelector('[name="' + key + '"]');  
    if (element) {  
      if (isReadOnly) {  
        element.setAttribute('readonly', 'readonly');  
      } else {  
        element.removeAttribute('readonly');  
      }  
    }  
  });  
}
```

- **Screen Shoot**

Product Data List

Show 10 entries Search:

UPC	Category Product	Brand	Batch
0005031315135	Grocery	Carmela	1
00044770007505	Dairy	Oscay Mayer	1

Showing 1 to 2 of 2 entries Previous 1 Next

Providers







Show 10 entries Search:

id_Provider	name_Provider	contact
1	Kraft	1-855-598-5493
2	B.Suarez	787-792-1212

Showing 1 to 2 of 2 entries Previous 1 Next

Records

Show 10 entries Search:

UPC	Name Product	Product Price	Product Quantity	Create
0005031315135	Salchicha Carmela	0.59	50	 
00044770007505	Hotdogs Oscar Mayer Beef	6.59	18	 
0088592700104	24pk water 16oz Salutaris	3.68	113	 

Showing 1 to 3 of 3 entries Previous 1 Next

Retrospectiva del trabajo:

o Preguntas:

- ¿Qué salió bien? Por qué

Las 3 interfaces se lograron ver con el contenido de la base de datos 'InventoryDB'. Adicional, se creó la base de datos y se logró conectar hacia el Front End por medio del Flask.

- ¿Qué salió mal? Por qué

No se pudo implementar las funciones de CRUD en las interfaces, por que no se supo implementar en el JavaScript.

- ¿Qué aprendió?

Se pudo aprender a conectar una base de datos y sus funciones de CRUD, por medio de Flask, a un Front End.

- ¿Qué requiere fortalecer?

Requiere fortalecer el conocimiento de JavaScript y de Nginx. Sobre js, por enlazar las funciones de CRUD con la interfaz y por otra parte, con el Nginx para hacer la conexión de Front End con el cliente.

- ¿Qué beneficios pudiera aportar la metodología AGILE o SCRUM en el proyecto?

Los beneficios que pudiera aportar estas metodologías en el proyecto, es en el cómo el equipo pudiera gestionar las prioridades del proyecto

- ¿Cree que debería haber alguien que distribuya el trabajo en su grupo? Por qué

Si, entendemos que debería haber alguien distribuyendo los trabajos para asi el quipo no perder el enfoque y tener una dirección.

Dirección de GITHUB: