DOCUMENTACIÓN DE CÓDIGO

SISTEMA WEB DE GESTIÓN DE RESTAURANTE “LA MURALLA”

Tabla de contenido

Introducción 2

Descripción 2

Descripción detallada de carpetas y archivos 4

Guías de estilo 5

Convenciones aplicadas 5

Formato de Código 5

Comentarios 7

Descripción de código 8

Frontend 8

Backend 12

Descripción de funcionalidad de los módulos 18

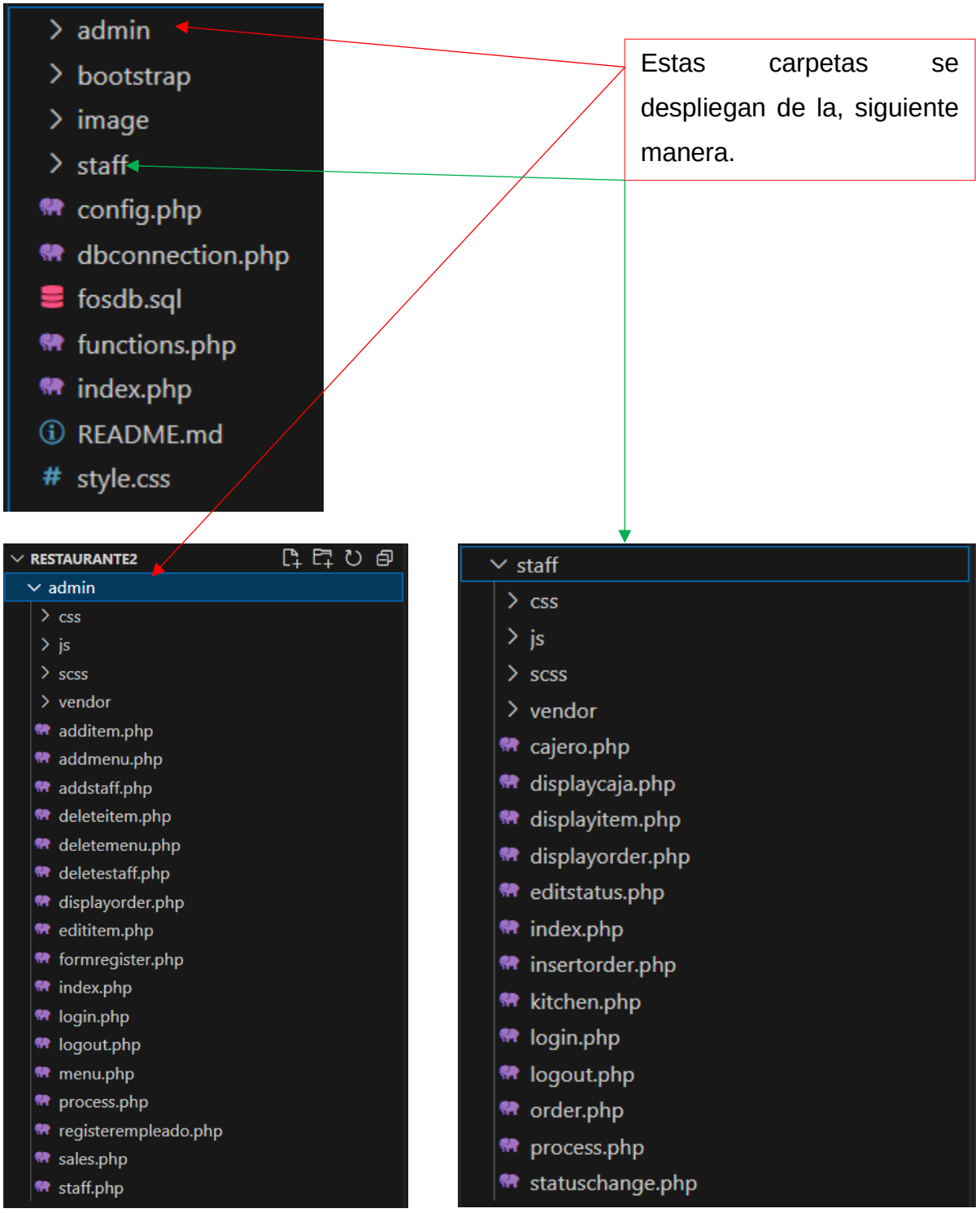
Administrador 18

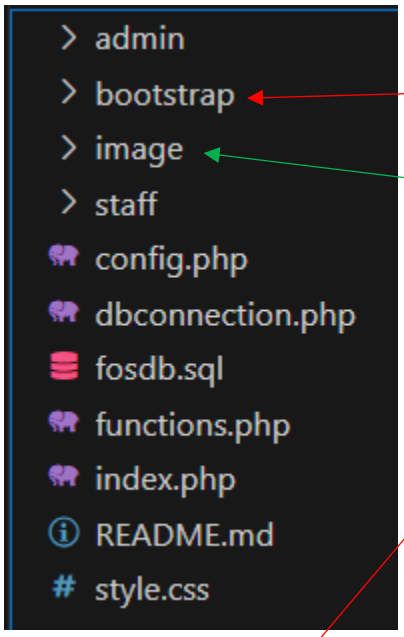
Empleado 21

Introducción

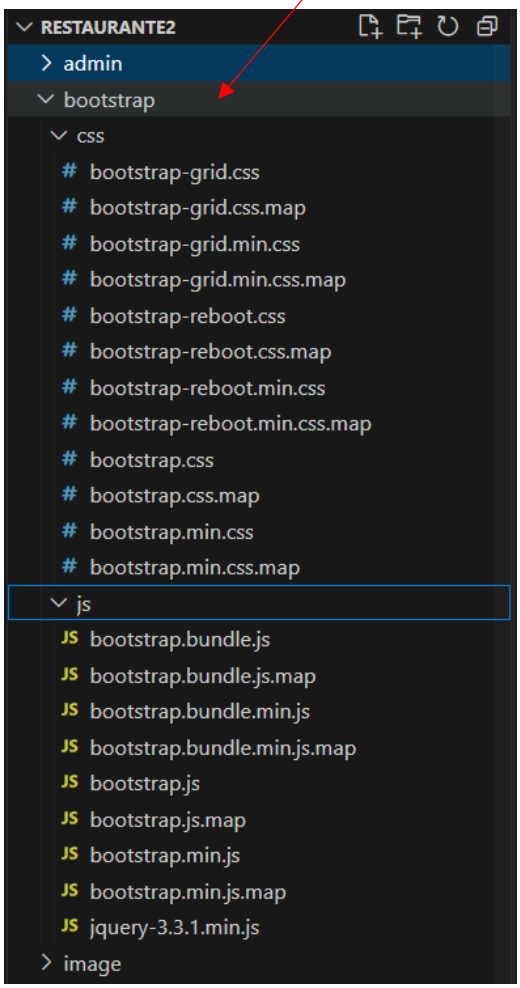
Descripción

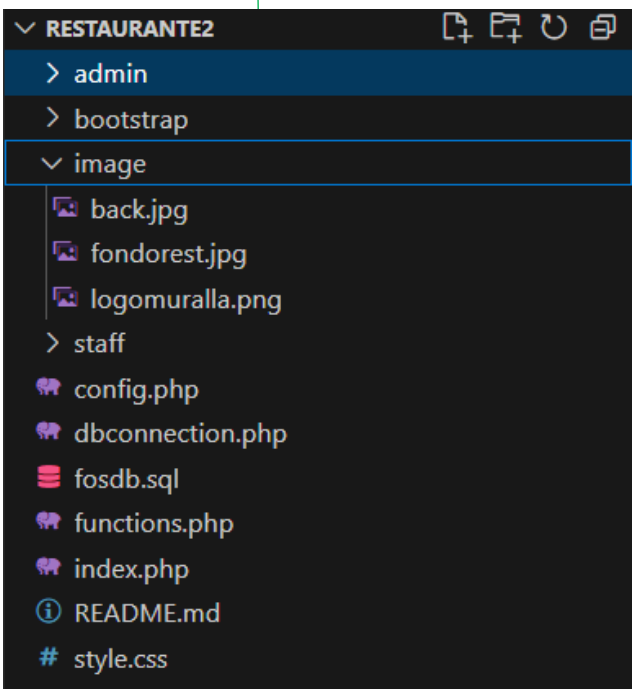
El sistema web está organizado en una estructura de carpetas clara y lógica para separar las responsabilidades y facilitar el mantenimiento y la escalabilidad del código.

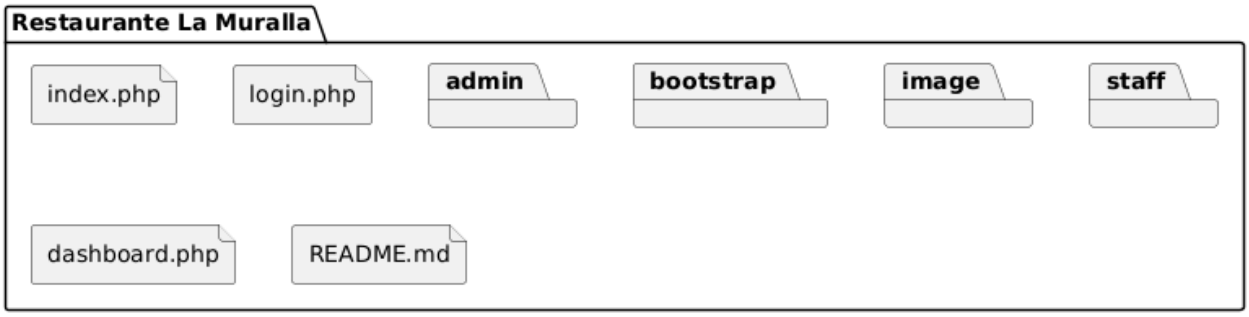




Estas carpetas se despliegan de la, siguiente manera.







Descripción detallada de carpetas y archivos

/admin: Contiene los archivos exclusivos para el desarrollo del módulo de administrador.

/staff: Contiene los archivos dedicados al desarrollo del módulo de los empleados.

Estas carpetas contienen la siguiente distribución internamente.

- /css: Almacena las hojas de estilo en cascada (CSS) que definen la apariencia visual y el diseño general de la aplicación.
- /js: Contiene los scripts JavaScript que implementan la lógica del frontend, gestionando la interacción y la dinámica en el navegador.
- /scss: Almacena archivos Sass (Syntactically Awesome Style Sheets) que mejoran la modularidad y mantenibilidad del código CSS mediante el uso de variables, anidación y mixins.
- /vendor: Incluye dependencias externas y bibliotecas de terceros necesarias para el funcionamiento del backend, facilitando la modularidad y el manejo de paquetes.

/bootstrap: Contiene las librerías de estilos generales proporcionadas por Bootstrap, facilitando el desarrollo de interfaces de usuario responsivas.

/image: Almacena las imágenes e iconos utilizados en toda la página, contribuyendo a la estética y usabilidad visual del sitio.

Guías de estilo

Convenciones aplicadas

Variables:

- Se aplica la notación camelCase para los nombres de las variables.
- Ejemplo: *$nombreVariable*.

Funciones:

- Se aplica la notación camelCase para los nombres de las funciones.
- Ejemplo: *function nombreFuncion($parametro1, $parametro2)*.

Clases:

- Se aplica la notación PascalCase para los nombres de las clases.
- Ejemplo: *class NombreClase*.

Constantes:

- Se aplica la notación en mayúsculas y separadas por guiones bajos.
- Ejemplo: *define('NOMBRE_CONSTANTE', 'valor');*.

Formato de Código

Indentación:

- Se uso Visual Studio Code como editor de código fuente.
- Se aplicó 4 espacios para la indentación. Con tabulaciones.
- Para las condicionales se usó el estilo K&R (Kernighan & Ritchie)

```
if (condition) {
    statement;
} else {
    statement;
}
```

Líneas en blanco:

- Se aplico separaciones en las secciones lógicas del código con una línea en blanco.

```
function calculateSum(a, b) {
    let sum = a + b;

    return sum;
}

function calculateDifference(a, b) {
    let difference = a - b;

    return difference;
}
```

Longitud de línea:

- La longitud fue aplicada por el Visual Studio Code de manera automática.

Comillas:

- Se aplicó comillas simples para cadenas de texto, excepto en las interpolaciones de variables, en cuyo caso se usó comillas dobles.

```
let simpleString = 'Esta es una cadena de texto';
let interpolatedString = `Esta es una cadena con una variable: ${variable}`;
```

Comentarios

Comentarios en línea:

- Use comentarios en línea para explicar el propósito de líneas individuales de código.

```
//verificacion de usuario y contraseña ingresada
if (isset($_POST['username']) && isset($_POST['password']))
```

Comentarios de bloque:

- Use comentarios de bloque para describir secciones completas o bloques de código.

```
/*
*Se aplica las funciones solo para funciones de aplicacion a los modulos
*Solo funciones con consultas query
*/
```

Descripción de código

Frontend

HTML y CSS

- index.php: Página principal que sirve como punto de entrada del sitio web.

index.php

```
4   <head>
9       <style>
52          }
53      </style>
54  </head>
55
56  <body>
57      <div class="container">
58          <h1>Restaurante | La Muralla</h1>
59          <button class="login-btn">Iniciar Sesión</button>
60          <div class="dropdown" id="dropdown">
61              <a href="staff" class="blue">Empleado</a>
62              <a href="admin" class="blue">Administrador</a>
63          </div>
64          <div class="logo-container">
65              <img src="./image/logomuralla.png" alt="Logo del restaurante La Muralla">
66          </div>
67          <!-- Otro contenido va aquí -->
68      </div>
69
70      <script>
71          var dropdown = document.getElementById("dropdown");
72          var loginBtn = document.querySelector(".login-btn");
73
74          loginBtn.addEventListener("click", function () {
75              dropdown.style.display = dropdown.style.display === "block" ? "none" : "block";
76          });
77
78          document.addEventListener("click", function (event) {
79              var isClickInside = loginBtn.contains(event.target) || dropdown.contains(event.target);
80
81              if (!isClickInside) {
82                  dropdown.style.display = "none";
83              }
84          });
85      </script>
86  </body>
```

- config.php: Archivo de configuración que almacena los parámetros necesarios para la conexión y configuración de la base de datos.

```
<?php

    //default database configuration
    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "resta2";

?>
```

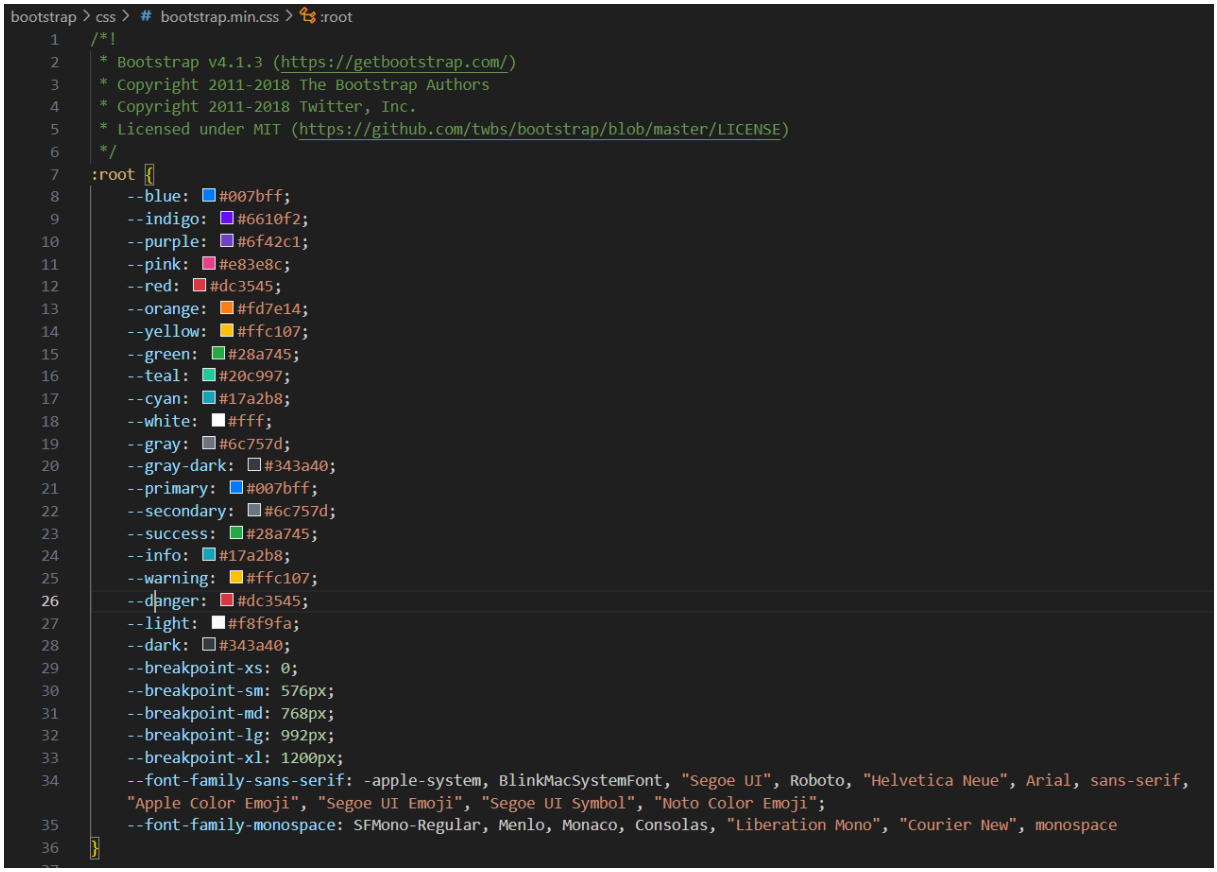
- dbconnection.php: Script que establece y gestiona la conexión con la base de datos, facilitando consultas y operaciones.

```php
<?php

require("config.php");

// Create connection
$sqlconnection = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($sqlconnection->connect_error) {
    die("Connection failed: " . $sqlconnection->connect_error);
}
```

- style.css: Hoja de estilos que define la presentación visual y el diseño consistente de todas las páginas web del sitio, así como el responsive menor a 600px, donde se repite el diseño.

```css
# style.css > ...
72  .button2 {
78      color: white;
79      text-decoration: none;
80  }
81
82
83  .button1:active,
84  .button2:active {
85      top: 108px;
86      box-shadow: 0px 7px 0px 0px #f02046;
87  }
88
89  .button1 {
90      background-color: #F2385A;
91  }
92
93  .button2 {
94      box-shadow: 0px 15px 0px 0px #258cd1;
95  }
96
97  .button2:active {
98      box-shadow: 0px 7px 0px 0px #258cd1;
99  }
100
101 @media screen and (max-width: 600px) {
102     a.blue {
103         height: auto;
104         max-width: auto;
105         padding: 10px;
106         font-size: 16px;
107     }
108 }
109 @media screen and (max-width: 600px) {
110     img {
111         max-width: 80%; /* Ajusta el tamaño del logo para pantallas más pequeñas */
112     }
113 }
```

- bootstrap.min.css: define estilos para componentes reutilizables como botones, formularios, tipografías y elementos de navegación, para las visualizaciones en respuesta del backend.



JavaScript

- bootstrap/js/bootstrap.min.js: son scripts de Bootstrap para activar modales que muestran mensajes emergentes interactivos, tooltips que proporcionan información útil al pasar el ratón sobre elementos específicos, y dropdowns que permiten a los usuarios seleccionar opciones de manera intuitiva, para las visualizaciones en respuesta del backend.

```
/*!
 * Bootstrap v4.1.3 (https://getbootstrap.com/)
 * Copyright 2011-2018 The Bootstrap Authors (https://github.com/twbs/bootstrap/graphs/contributors)
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */
(function (global, factory) {
  typeof exports === 'object' && typeof module !== 'undefined' ? factory(exports, require('jquery')) :
  typeof define === 'function' && define.amd ? define(['exports', 'jquery'], factory) :
  (factory((global.bootstrap = {}),global.jQuery));
}(this, (function (exports,$) { 'use strict';

  $ = $ && $.hasOwnProperty('default') ? $['default'] : $;

  function _defineProperties(target, props) {
    for (var i = 0; i < props.length; i++) {
      var descriptor = props[i];
      descriptor.enumerable = descriptor.enumerable || false;
      descriptor.configurable = true;
      if ("value" in descriptor) descriptor.writable = true;
      Object.defineProperty(target, descriptor.key, descriptor);
    }
  }

  function _createClass(Constructor, protoProps, staticProps) {
    if (protoProps) _defineProperties(Constructor.prototype, protoProps);
    if (staticProps) _defineProperties(Constructor, staticProps);
    return Constructor;
  }

  function _defineProperty(obj, key, value) {
    if (key in obj) {
      Object.defineProperty(obj, key, {
        value: value,
        enumerable: true,
        configurable: true,
        writable: true
      });
```

- En las carpetas de admin y staff, se emplea los mismos códigos para las funcionalidades de respuesta del backend.
    - (admin o staff)/js/demo/sb-admin.min.js: alterna el sidebar, gestiona el desplazamiento en un sidebar fijo, muestra un botón de scroll hacia arriba y animar el desplazamiento hacia la parte superior de la página.

```
admin > js > JS sb-admin.min.js > ...
/*!
 * Start Bootstrap - SB Admin v5.0.2 (https://startbootstrap.com/template-overviews/sb-admin)
 * Copyright 2013-2018 Start Bootstrap
 * Licensed under MIT (https://github.com/BlackrockDigital/startbootstrap-sb-admin/blob/master/LICENSE)
 */

!function(t){"use strict";t("#sidebarToggle").click(function(e){e.preventDefault(),t("body").toggleClass("sidebar-toggled"),
    t(".sidebar").toggleClass("toggled")}),t("body.fixed-nav .sidebar").on("mousewheel DOMMouseScroll wheel",
        function(e){if(768<$window.width()){var o=e.originalEvent,t=o.wheelDelta||-o.detail;this.scrollTop+=30*(t<0?1:-1),
            e.preventDefault()}}),t(document).scroll(function(){100<t(this).scrollTop()?t(".scroll-to-top")
                .fadeIn():t(".scroll-to-top").fadeOut()}),t(document)
                .on("click","a.scroll-to-top",function(e){var o=t(this);t("html, body")
                    .stop().animate({scrollTop:t(o.attr("href")).offset().top},1e3,"easeInOutExpo"),
                    e.preventDefault()})}(jQuery);
```

- (admin o staff)/js/demo/sb-admin.js: activa y desactiva la barra lateral de navegación, previene el desplazamiento no deseado del contenido principal, muestra el botón de scroll para arriba dinámico, y desplazamiento suave al hacer clic a enlaces.

admin > js > JS sb-admin.js > ...

```
(function($) {
  "use strict"; // Start of use strict

  // Toggle the side navigation
  $("#sidebarToggle").on('click',function(e) {
    e.preventDefault();
    $("body").toggleClass("sidebar-toggled");
    $(".sidebar").toggleClass("toggled");
  });

  // Prevent the content wrapper from scrolling when the fixed side navigation hovered over
  $('body.fixed-nav .sidebar').on('mousewheel DOMMouseScroll wheel', function(e) {
    if ($(window).width() > 768) {
      var e0 = e.originalEvent,
        delta = e0.wheelDelta || -e0.detail;
      this.scrollTop += (delta < 0 ? 1 : -1) * 30;
      e.preventDefault();
    }
  });

  // Scroll to top button appear
  $(document).on('scroll',function() {
    var scrollDistance = $(this).scrollTop();
    if (scrollDistance > 100) {
      $('.scroll-to-top').fadeIn();
    } else {
      $('.scroll-to-top').fadeOut();
    }
  });

  // Smooth scrolling using jQuery easing
  $(document).on('click', 'a.scroll-to-top', function(event) {
    var $anchor = $(this);
    $('html, body').stop().animate({
      scrollTop: ($($anchor.attr('href')).offset().top)
    }, 1000, 'easeInOutExpo');
    event.preventDefault();
```

Backend

Controladores

functions.php: contiene funciones para realizar operaciones específicas en la base de datos y manejar los resultados o errores según resulte. También, hace que las consultas SQL, las cuales se instanciasn en los módulos, sean seguras para evitar vulnerabilidades como la inyección SQL.

functions.php

```php
//established the connection between databse
require("dbconnection.php");

session_start();

//insert user defined function here
// TODO - dynamic query
function getNumRowsQuery($query)
{
    global $sqlconnection;
    if ($result = $sqlconnection->query($query))
        return $result->num_rows;
    else
        echo "Algo anda mal la consulta!";
}

function getFetchAssocQuery($query)
{
    global $sqlconnection;
    if ($result = $sqlconnection->query($query)) {

        while ($row = $result->fetch_array(MYSQLI_ASSOC)) {
            echo "\n", $row["itemID"], $row["menuID"], $row["menuItemName"], $row["price"];
        }

        //print_r($result);

        return ($result);
    } else
        echo "Algo anda mal la consulta!";
    echo $sqlconnection->error;
}

function getLastID($id, $table)
{
    global $sqlconnection;

    $query = "SELECT MAX({$id}) AS {$id} from {$table} ";
```

- Función getNumRowsQuery($query): ejecuta la consulta SQL especificada en $query y devuelve el número de filas resultantes (cantidad de filas afectadas por la consulta).
- Función getFetchAssocQuery($query): ejecuta la consulta SQL en $query y devuelve un array asociativo con los resultados de la consulta. Imprime los resultados en formato de texto si se encuentran, de lo contrario, muestra un mensaje de error.
- Función getLastID($id, $table): ejecuta la consulta para obtener el valor máximo de $id desde la tabla $table. Devuelve el último valor de $id encontrado en la tabla.

- Función getCountID($idnum, $id, $table): ejecuta la función cuenta cuántas veces aparece el valor $idnum en la columna $id de la tabla $table. Retorna el número de veces que se encuentra $idnum.
- Función getSalesTotal($orderID): ejcecuta la consulta a la base de datos para obtener el total de ventas correspondiente al $orderID especificado desde la tabla tbl_order.
- Función getSalesGrandTotal($duration): ejecuta el cálculo del total acumulado de ventas basado en un período de tiempo especificado por $duration ("ALLTIME", "DAY", "MONTH", "WEEK"). Realiza una suma de los totales de todas las órdenes de venta en función del período seleccionado.
- Función updateTotal($orderID): realiza la actualización en el campo total en la tabla tbl_order para la orden identificada por $orderID. Calcula el nuevo total sumando los precios de los productos ordenados según la tabla de detalles de orden y luego actualiza el registro en la tabla principal tbl_order.

En las carpetas de admin y staff, se emplea los mismos códigos para las consultas propias de cada módulo.

- (admin o staff)/process.php: valida las credenciales de inicio de sesión a la base de datos (resta2.sql) y establece variables de sesión ($_SESSION) si las credenciales son correctas, todo mientras se protege contra inyecciones SQL utilizando funciones de escape proporcionadas por MySQLi (real_escape_string).

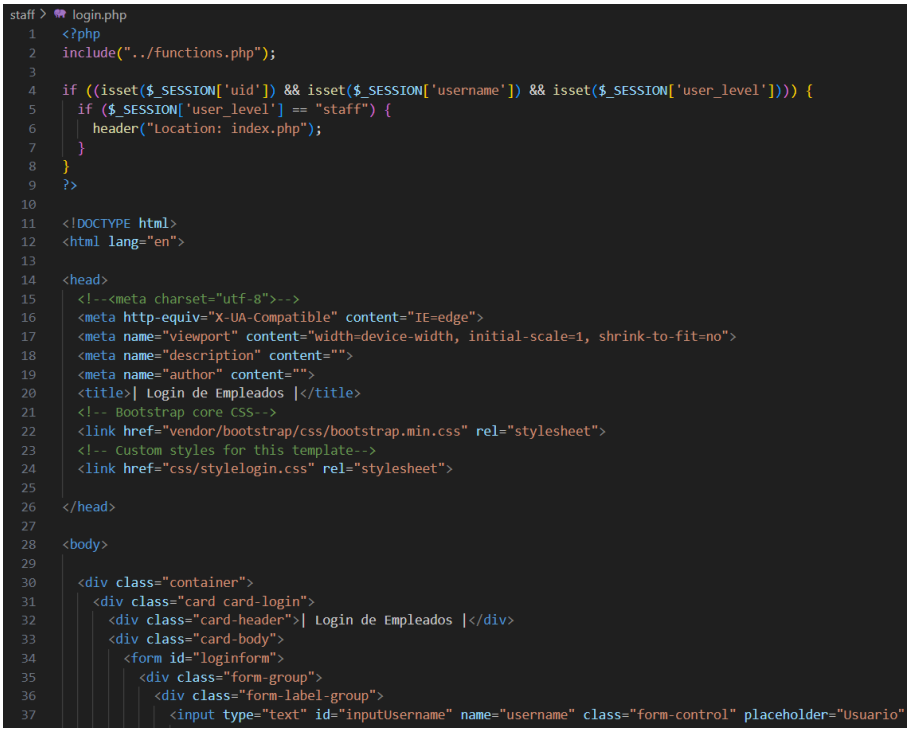
admin > process.php

```php
<?php
    include("../functions.php");

        //verificacion de usuario y contraseña ingresada
        if (isset($_POST['username']) && isset($_POST['password'])) {

            //prevent sql injection by escaping special characters
            $username = $sqlconnection->real_escape_string($_POST['username']);
            $password = $sqlconnection->real_escape_string($_POST['password']);

            //sql statement
            $sql = "SELECT * FROM tbl_admin WHERE username ='$username' AND password = '$password'";

            if ($result = $sqlconnection->query($sql)) {

                if ($row = $result->fetch_array(MYSQLI_ASSOC)) {

                    $uid = $row['ID'];
                    $username = $row['username'];

                    $_SESSION['uid'] = $uid;
                    $_SESSION['username'] = $username;
                    $_SESSION['user_level'] = "admin";

                    echo "correct";
                }

                else {
                    echo "Usuario o contraseña incorrecto.";
                }

            }

        }

?>
```

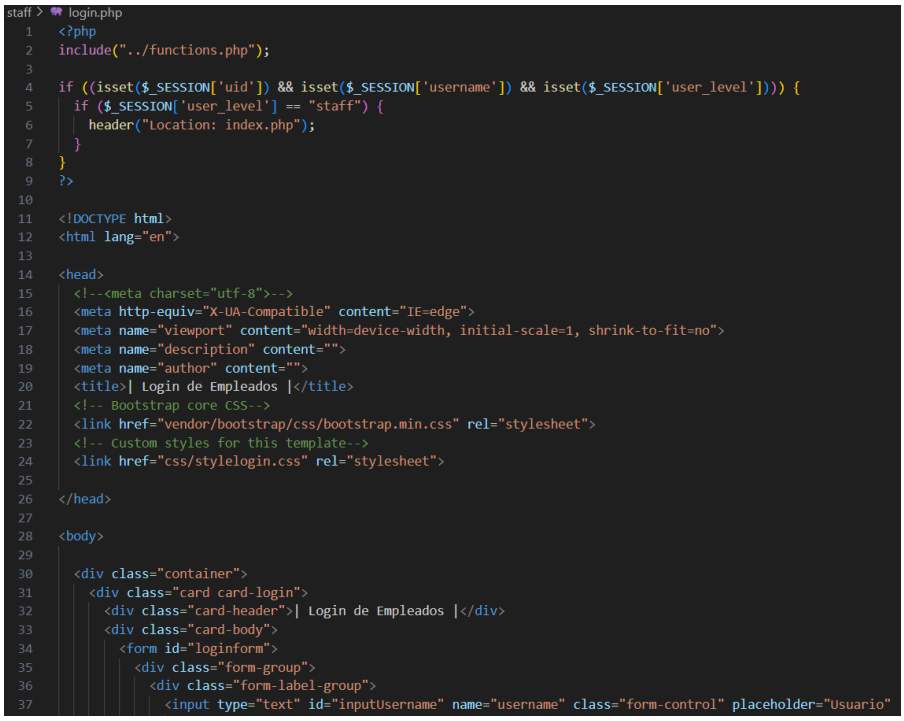
staff > process.php

```php
<?php
include("../functions.php");

//checking username and password input
if (isset($_POST['username']) && isset($_POST['password'])) {

    //prevent sql injection by escaping special characters
    $username = $sqlconnection->real_escape_string($_POST['username']);
    $password = $sqlconnection->real_escape_string($_POST['password']);

    //sql statement
    $sql = "SELECT * FROM tbl_staff WHERE username ='$username' AND password = '$password'";

    if ($result = $sqlconnection->query($sql)) {

        if ($row = $result->fetch_array(MYSQLI_ASSOC)) {

            $uid = $row['staffID'];
            $username = $row['username'];
            $role = $row['role'];

            $_SESSION['uid'] = $uid;
            $_SESSION['username'] = $username;
            $_SESSION['user_level'] = "staff"; // 1 - admin 2 - staff
            $_SESSION['user_role'] = $role;

            echo "correct";
        } else {
            echo "Usuario o contraseña incorrecto.";
        }
    }
}
```

- (admin o staff)/login.php: valida las credenciales de inicio de sesión a la base de datos (resta2.sql) y establece variables de sesión ($_SESSION) si las credenciales son correctas, se redirige automáticamente a index.php. si el nivel de usuario no corresponde a las variables de sesión.



```
staff > login.php
<?php
include("../functions.php");

if ((isset($_SESSION['uid']) && isset($_SESSION['username']) && isset($_SESSION['user_level']))) {
  if ($_SESSION['user_level'] == "staff") {
    header("Location: index.php");
  }
}
?>

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

<head>
  <!--<meta charset="utf-8">-->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">
  <title>| Login de Empleados |</title>
  <!-- Bootstrap core CSS-->
  <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <!-- Custom styles for this template-->
  <link href="css/stylelogin.css" rel="stylesheet">

</head>

<body>

  <div class="container">
    <div class="card card-login">
      <div class="card-header">| Login de Empleados |</div>
      <div class="card-body">
        <form id="loginform">
          <div class="form-group">
            <div class="form-label-group">
              <input type="text" id="inputUsername" name="username" class="form-control" placeholder="Usuario"
```



```
staff > login.php
<?php
include("../functions.php");

if ((isset($_SESSION['uid']) && isset($_SESSION['username']) && isset($_SESSION['user_level']))) {
  if ($_SESSION['user_level'] == "staff") {
    header("Location: index.php");
  }
}
?>

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

<head>
  <!--<meta charset="utf-8">-->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">
  <title>| Login de Empleados |</title>
  <!-- Bootstrap core CSS-->
  <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <!-- Custom styles for this template-->
  <link href="css/stylelogin.css" rel="stylesheet">

</head>

<body>

  <div class="container">
    <div class="card card-login">
      <div class="card-header">| Login de Empleados |</div>
      <div class="card-body">
        <form id="loginform">
          <div class="form-group">
            <div class="form-label-group">
              <input type="text" id="inputUsername" name="username" class="form-control" placeholder="Usuario"
```

- (admin o staff)/logout.php: hace que los usuarios validados con el nivel de usuario correspondiente tengan la opción de cerrar sesión (session_destroy()) y redirige a todos los usuarios, independientemente de su estado de sesión, a la página de inicio de sesión (login.php).

staff > logout.php

```php
<?php
    include("../functions.php");

    if((isset($_SESSION['uid']) && isset($_SESSION['username']) && isset($_SESSION['user_level'])) )  {
        if($_SESSION['user_level'] == "staff") {
            session_destroy();
            header("Location: login.php");
        }
        else
            header("Location: login.php");
    }

    else
        header("Location: login.php");

?>
```

admin > logout.php

```php
<?php
include("../functions.php");

if ((isset($_SESSION['uid']) && isset($_SESSION['username']) && isset($_SESSION['user_level']))) {
    if ($_SESSION['user_level'] == "admin") {
        session_destroy();
        header("Location: login.php");
    } else
        header("Location: login.php");
} else
    header("Location: login.php");

```

Descripción de funcionalidad de los módulos

Administrador

- additem.php: maneja la inserción de un nuevo menú en una base de datos MySQL, valida datos, ejecuta la consulta de inserción segura (real_escape_string) y maneja errores potenciales durante el proceso de inserción de datos ($sqlconnection->error).

admin > additem.php

```php
<?php

//Add new menu item
if (isset($_POST['addItem'])) {

    if (!empty($_POST['itemName']) && !empty($_POST['itemPrice']) && !empty($_POST['menuID'])) {
        $itemName = $sqlconnection->real_escape_string($_POST['itemName']);
        $itemPrice = $sqlconnection->real_escape_string($_POST['itemPrice']);
        $menuID = $sqlconnection->real_escape_string($_POST['menuID']);

        $addItemQuery = "INSERT INTO tbl_menuitem (menuID ,menuItemName ,price) VALUES ({$menuID} ,'{$itemName}' ,{$itemPrice})";

        if ($sqlconnection->query($addItemQuery) === TRUE) {
            header("Location: menu.php");
            exit();
        } else {
            //handle
            echo "someting wong";
            echo $sqlconnection->error;
        }
    }

    //No input handle
    else {
        echo "Jangan bia kosong bang";
    }
}

```

- addmenu.php: permite agregar nuevos elementos, menús o categorías, a la base de datos, empleando el método POST.

admin > addmenu.php

```php
<?php

//Add new menu (category)
if (isset($_POST['addmenu'])) {

    if (!empty($_POST['menuname'])) {
        $menuname = $sqlconnection->real_escape_string($_POST['menuname']);

        $addMenuQuery = "INSERT INTO tbl_menu (menuName) VALUES ('{$menuname}')";

        if ($sqlconnection->query($addMenuQuery) === TRUE) {
            header("Location: menu.php");
        } else {
            //handle
            echo "someting wong";
        }
    }

    //No input handle
    else {
        echo "Jangan bia kosong bang 12";
    }
}

```

- formregister.php: muestra el diseño del panel de control del módulo del administrador.

admin > formregister.php

```php
<?php
include("../functions.php");

// Verifica la sesión del usuario
if ((!isset($_SESSION['uid']) && !isset($_SESSION['username']) && isset($_SESSION['user_level'])))
    header("Location: login.php");

// Verifica el nivel de acceso del usuario
if ($_SESSION['user_level'] != "admin")
    header("Location: login.php");
?>

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

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Panel de Control</title>

    <!-- Bootstrap core CSS-->
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom fonts for this template-->
    <link href="vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">

    <!-- Page level plugin CSS-->
    <link href="vendor/datatables/dataTables.bootstrap4.css" rel="stylesheet">

    <!-- Custom styles for this template-->
    <link href="css/sb-admin.css" rel="stylesheet">

```

- registerempleado.php: maneja la inserción de un nuevo empleado desde un formulario HTML a una base de datos MySQL (método POST), asegurando la seguridad y la integridad de los datos mediante consultas SQL preparadas y el manejo adecuado de errores ($stmt->error).

admin > registerempleado.php

```php
<?php
require("../config.php");

// Establecer la conexión
$conn = new mysqli($servername, $username, $password, $dbname);

// Manejar errores de conexión
if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}

// Obtener datos del formulario
$username = $_POST['username'];
$password = $_POST['password'];
$estatus = 'Offline';
$rolID = null;
$role = $_POST['rol'];
$name = $_POST['name'];
$appPater = $_POST['appPater'];
$appMater = $_POST['appMater'];
$dni = $_POST['dni'];

// Consulta SQL para insertar datos con declaración preparada
$sql = "INSERT INTO tbl_staff (username, password, status, roleID, role, name, appPater, appMater, DNI)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";

// Preparar la declaración
if ($stmt = $conn->prepare($sql)) {
    // Vincular parámetros
    $stmt->bind_param("sssssssss", $username, $password, $estatus, $rolID, $role, $name, $appPater, $appMater, $dni);

    // Ejecutar la consulta
    if ($stmt->execute()) {
        echo "<script>
        alert('Registro exitoso');
        window.location.href = 'formregister.php';
      </script>";
```

- staff.php: valida el incio de sesión correspondiente al nivel de usuario de administrador, actualiza la lista de empleados con tres parámetros ($connection, $role, y $staffID), cambio de rol con consulta (método POST).

admin > staff.php

```php
<?php
include("../functions.php");

// Verificar la sesión y los permisos de administrador
if (!isset($_SESSION['uid']) || !isset($_SESSION['username']) || $_SESSION['user_level'] !== "admin") {
  header("Location: login.php");
  exit();
}

// Función para actualizar el rol del personal
function updateStaffRole($connection, $role, $staffID)
{
  $updateRoleQuery = $connection->prepare("UPDATE tbl_staff SET role = ? WHERE staffID = ?");
  $updateRoleQuery->bind_param("si", $role, $staffID);

  if ($updateRoleQuery->execute()) {
    return true;
  } else {
    // Manejar el error
    return "Algo salió mal: " . $updateRoleQuery->error;
  }

  $updateRoleQuery->close();
}

// Procesar el formulario si se envió
if (!empty($_POST['role'])) {
  $role = $sqlconnection->real_escape_string($_POST['role']);
  $staffID = $sqlconnection->real_escape_string($_POST['staffID']);

  $result = updateStaffRole($sqlconnection, $role, $staffID);

  if ($result !== true) {
    echo $result;
  }
```

Empleado

- displayitem.php: realiza consultas con la base de datos para mostrar elementos de menú y detalles de pedido para la interfaz, siempre validando la autenticación del usuario y el manejo adecuado de datos a través de formularios.

staff > displayitem.php

```php
<?php
include("../functions.php");

if ((!isset($_SESSION['uid']) && !isset($_SESSION['username']) && isset($_SESSION['user_level'])))
    header("Location: login.php");

if ($_SESSION['user_level'] != "staff")
    header("Location: login.php");

if (isset($_POST['btnMenuID'])) {

    $menuID = $sqlconnection->real_escape_string($_POST['btnMenuID']);

    $menuItemQuery = "SELECT itemID,menuItemName FROM tbl_menuitem WHERE menuID = " . $menuID;

    if ($menuItemResult = $sqlconnection->query($menuItemQuery)) {
        if ($menuItemResult->num_rows > 0) {
            $counter = 0;
            while ($menuItemRow = $menuItemResult->fetch_array(MYSQLI_ASSOC)) {

                if ($counter >= 3) {
                    echo "</tr>";
                    $counter = 0;
                }

                if ($counter == 0) {
                    echo "<tr>";
                }

                echo "<td><button style='margin-bottom:4px;white-space: normal;' class='btn btn-warning' onclick = 'setQty({$menuI

                $counter++;
            }
        } else {
            echo "<tr><td>No item in this menu</td></tr>";
        }
    }
```

- displayorder.php: interactúa con la base de datos mediante consultas SQL para gestionar la visualización y el estado de las órdenes, para la interfaz de usuario interactiva y funcional para el personal encargado de la gestión de pedidos.

staff > displayorder.php

```php

<?php
include("../functions.php");

if ((!isset($_SESSION['uid']) && !isset($_SESSION['username']) && isset($_SESSION['user_level'])))
    header("Location: login.php");

if ($_SESSION['user_level'] != "staff")
    header("Location: login.php");

//display none when open /displayorder.php
if (empty($_GET['cmd']))
    die();

//display current order list for kitchen management
if ($_GET['cmd'] == 'currentorder') {

    $displayOrderQuery =  "
                    SELECT O.orderID, M.menuName, OD.itemID,MI.menuItemName,OD.quantity,O.status
                    FROM tbl_order O
                    LEFT JOIN tbl_orderdetail OD
                    ON O.orderID = OD.orderID
                    LEFT JOIN tbl_menuitem MI
                    ON OD.itemID = MI.itemID
                    LEFT JOIN tbl_menu M
                    ON MI.menuID = M.menuID
                    WHERE O.status
                    IN ( 'waiting','preparing','ready')
                ";

    if ($orderResult = $sqlconnection->query($displayOrderQuery)) {

        $currentspan = 0;

        //if no order
        if ($orderResult->num_rows == 0) {
```

- editstatus.php: verifica la autenticación de usuarios, los permisos de acceso, y permite la actualización del estado de órdenes en una base de datos según la entrada del usuario a través de formularios (POST) o parámetros de URL (GET).

staff > editstatus.php

```php
<?php
include("../functions.php");

if ((!isset($_SESSION['uid']) && !isset($_SESSION['username']) && isset($_SESSION['user_level'])))
    header("Location: login.php");

if ($_SESSION['user_level'] != "staff")
    header("Location: login.php");

if (isset($_POST['status']) && isset($_POST['orderID'])) {

    $status = $sqlconnection->real_escape_string($_POST['status']);
    $orderID = $sqlconnection->real_escape_string($_POST['orderID']);

    $addOrderQuery = "UPDATE tbl_order SET status = '{$status}' WHERE orderID = {$orderID};";

    if ($sqlconnection->query($addOrderQuery) === TRUE) {
        echo "inserted.";
    } else {
        //handle
        echo "someting wong";
        echo $sqlconnection->error;
    }
}

if (isset($_GET['orderID'])) {

    $status = "Completed";
    $orderID = $sqlconnection->real_escape_string($_GET['orderID']);

    $addOrderQuery = "UPDATE tbl_order SET status = '{$status}' WHERE orderID = {$orderID};";

    if ($sqlconnection->query($addOrderQuery) === TRUE) {
        echo "inserted.";
        header("Location: index.php");
    } else {
        //handle
```

- insertorder.php: regitra las órdenes de compra correctamente en la base de datos y los detalles asociados se guarden de manera consistente, proporcionando retroalimentación al usuario sobre el estado de las operaciones a través de mensajes de éxito o error.

staff > insertorder.php

```php
<?php
include("../functions.php");

if ((!isset($_SESSION['uid']) && !isset($_SESSION['username']) && isset($_SESSION['user_level'])))
    header("Location: login.php");

if ($_SESSION['user_level'] != "staff")
    header("Location: login.php");

if (isset($_POST['sentorder'])) {

    if (isset($_POST['itemID']) && isset($_POST['itemqty'])) {

        $arrItemID = $_POST['itemID'];
        $arrItemQty = $_POST['itemqty'];

        // Asegúrate de que haya elementos en ambos arrays
        if (count($arrItemID) == count($arrItemQty)) {

            // Insertar una nueva orden
            $insertedOrderID = insertOrderQuery();

            if ($insertedOrderID !== false) {
                // Insertar detalles de la orden
                for ($i = 0; $i < count($arrItemID); $i++) {
                    insertOrderDetailQuery($insertedOrderID, $arrItemID[$i], $arrItemQty[$i]);
                }

                // Actualizar el total de la orden
                updateTotal($insertedOrderID);

                // Redirigir a la página de inicio después de completar la inserción de la orden
                header("Location: index.php");
                exit();
            } else {
                echo "Error al insertar la orden.";
            }
```

- kitchen.php: asegura que solo los usuarios autenticados y con el nivel adecuado de acceso (staff y chef) puedan acceder al panel de administración de la cocina, para que se visualicen en el web.

staff > kitchen.php

```php
<?php
include("../functions.php");

// Redireccionar si no hay una sesión activa
if ((!isset($_SESSION['uid']) || !isset($_SESSION['username']) || !isset($_SESSION['user_level'])))
  header("Location: login.php");

// Redireccionar si el nivel de usuario no es "staff"
if ($_SESSION['user_level'] != "staff")
  header("Location: login.php");

// Verificar el rol del usuario
if ($_SESSION['user_role'] != "chef") {
  echo ("<script>window.alert('¡Disponible solo para chefs!'); window.location.href='index.php';</script>");
  exit();
}
?>

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

<head>

  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">

  <title>Dashboard - FOS Staff</title>

  <!-- Bootstrap core CSS-->
  <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

  <!-- Custom fonts for this template-->
  <link href="vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">
```