

# GREATFOOD

MEMORIA DEL PROYECTO

---



Componentes del grupo:

- Álvaro López
- Gonzalo Ruiz
- José Torres

## ÍNDICE

1. Descripción y objetivos del proyecto.	3
2. Cronograma. Evolución en los diagramas de GANTT.	4
3. Diseño lógico: Análisis de los datos.	5
4. Diseño físico de los datos.	7
5. Mapa de navegación.	9
6. Código de los módulos y captura de pantallas.	10
7. Juego de ensayo y pruebas.	41
8. Recursos y bibliografía utilizada.	51

## 1. Descripción y objetivos del proyecto.

Greatfood es una empresa ficticia cuyo objetivo principal es el de repartir comida a domicilio a sus clientes. La idea surgió a través de un cliente real (un grupo de alumnos de Administración y Finanzas). Realizamos varias reuniones para que nos transmitieran su idea y nos dieran pautas para la realización de la web que nos pedían.

El nombre de Greatfood es la combinación de dos palabras en inglés, 'Great' y 'Food', cuyo significado en español es "gran/grande" y "comida rara".

Tras todo esto hemos llegado a la conclusión de tener como objetivos crear y diseñar una base de datos la cual tiene que incluir información relacionada con los clientes que van a usar la página, los platos disponibles y las compras que realizan dichos clientes. También tenemos que maquetar una página web que albergue aplicaciones (HTML/CSS/JS), la cual debe de ser intuitiva, ingeniosa y novedosa, sin repetir código, ya que esto permitirá una navegación más fluida.

Debemos tener gestión total del diseño de la web, utilizando Bootstrap y JQuery UI, mediante la realización de una plantilla inicial, la cual nos sirve para el resto de vistas.

La web debe tener una funcionalidad de login/registro usando sesiones de PHP y conexiones a MySQL, el registro deberá usar validaciones mediante expresiones regulares y JQuery. También debe tener un carrito de la compra avanzado que utiliza PHP, MySQL, cookies y Ajax, dicho carrito deberá mostrar los diferentes platos disponibles en la empresa, así como las funcionalidades de borrar carrito y de mantener el carrito si se cambia de vista.

Al procesar el carrito, deberemos implementar una TPV Virtual que simula un pago real. También debemos tener un lugar donde poder consultar los pedidos realizados y otro que permita cambiar la contraseña al usuario.

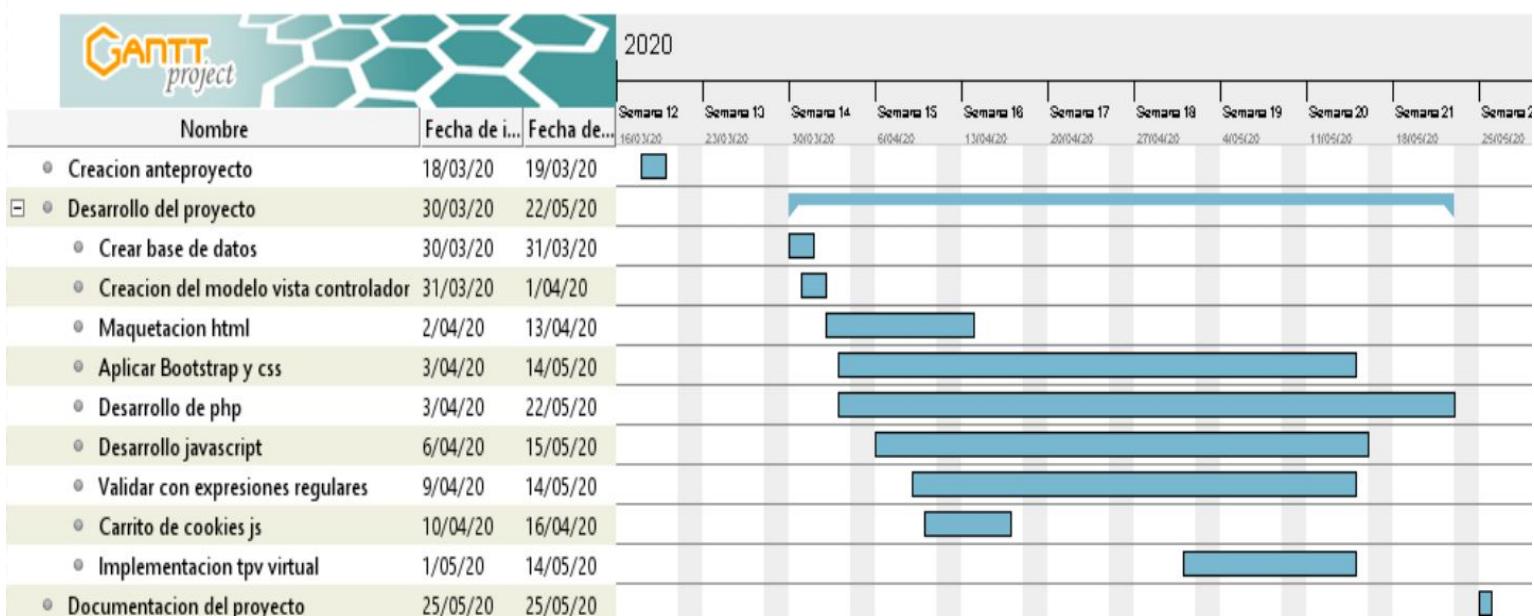
Otra funcionalidad importante es implementar un servicio de correo que envíe información a los usuarios cuando se registren o cuando realicen un pedido.

Debemos crear un formulario de contacto que envíe correos con la información correspondiente. Por último debemos desplegar la aplicación completa en la nube utilizando servicios como Red Hat OpenShift o clouding.io y GitHub.

## 2. Cronograma. Evolución en los diagramas de GANTT.

Un diagrama de GANTT es una forma de representar el tiempo previsto a la realización de diferentes tareas de un proyecto.

La siguiente imagen representa los tiempos dedicados a cada tarea necesaria para finalizar la web, desde la creación del anteproyecto hasta la realización de la documentación final.

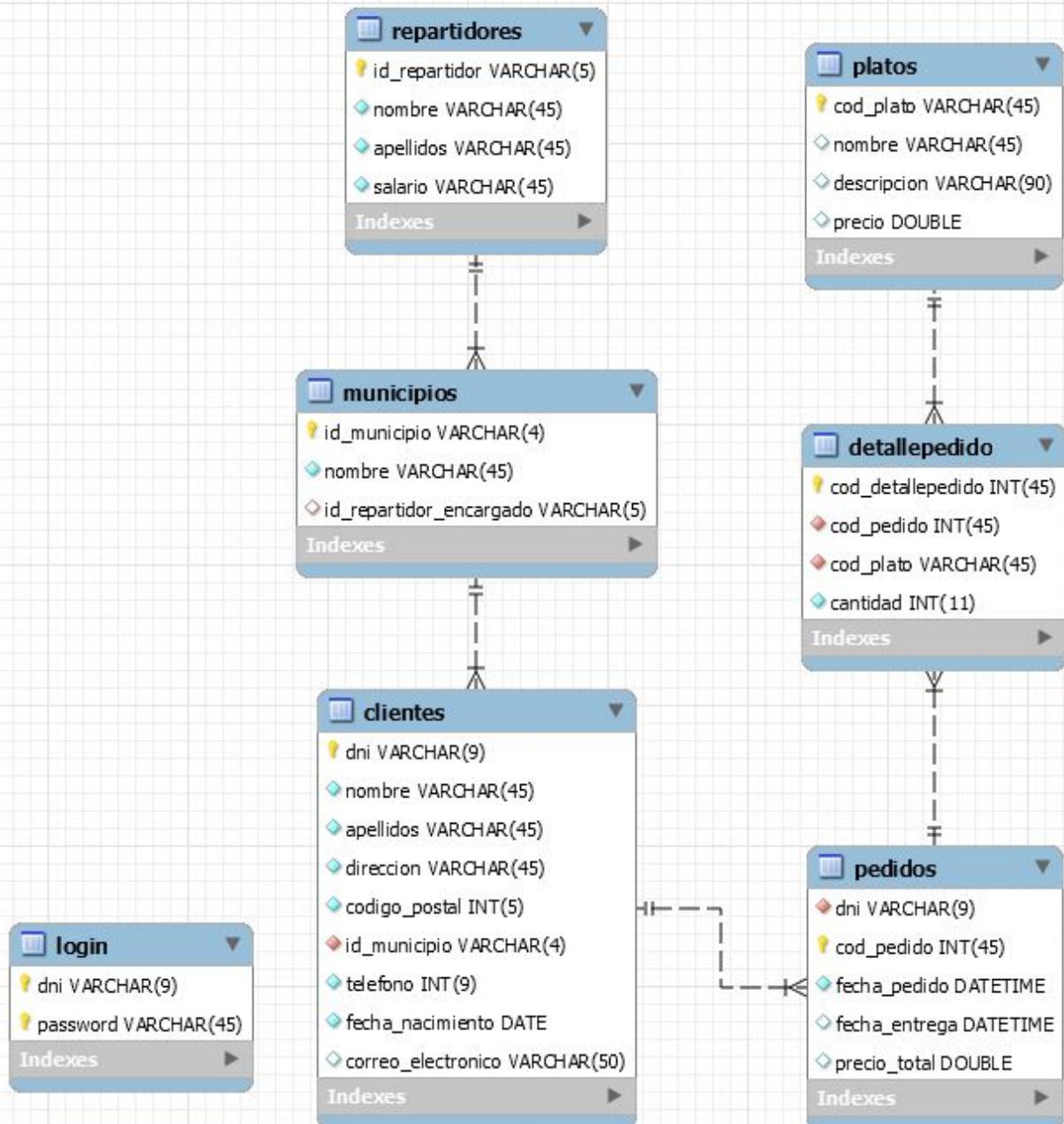


Al final, cumplimos nuestras previsiones de tiempo, empezamos trabajando con la base de datos, ya que era lo más imprescindible al principio.

Los bloques que más tiempo hemos tenido que dedicar han sido los relacionados con el desarrollo, tanto en PHP, como en JavaScript y CSS.

### 3. Diseño lógico: Análisis de los datos.

Los datos de nuestra aplicación están estructurados de la siguiente forma:



Amarillo: Claves primarias

Rojo: Claves ajenas

La base de datos está organizada de manera que:

- Los clientes son los usuarios registrados con sus todos sus datos generales, los cuales ha dado a la hora de registrarse.
- La tabla login es una tabla de clientes pero la cual solo contiene el DNI de la persona registrada y la contraseña que él mismo puso.
- Los clientes pueden hacer varios pedidos mientras la fecha de pedido sea distinta (la fecha\_pedido contabiliza hasta los segundos).
- La tabla detallepedidos es una tabla que contiene todos los platos que se ha realizado en un pedido.
- Los platos constituyen todas las comidas que hay en la base de datos con su correspondiente información general.
- El reparto de la comida se hace de forma que dependiendo del municipio que resida el clientes, se le asigna un repartidor que siempre reparte en esa zona, por eso están relacionados la tabla clientes y repartidores con una tabla intermedia llamada municipios.

```
+-----+
| Tables_in_greatfood |
+-----+
| clientes           |
| detallepedido     |
| login              |
| municipios         |
| pedidos            |
| platos             |
| repartidores       |
+-----+
7 rows in set (0.00 sec)
```

#### 4. Diseño físico de los datos.

Para la base de datos del punto anterior hemos creado un fichero .sql el cual ejecutamos la primera vez que desplegamos la aplicación (en este caso en nuestra máquina virtual de clouding.io).

Dicho fichero tiene las tablas y los inserts necesarios para el correcto funcionamiento de la aplicación.

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

CREATE DATABASE IF NOT EXISTS `greatfood` DEFAULT CHARACTER SET
    latin1 COLLATE latin1_swedish_ci;
USE `greatfood`;

DROP TABLE IF EXISTS `clientes`;
CREATE TABLE IF NOT EXISTS `clientes` (
    `dni` varchar(9) NOT NULL,
    `nombre` varchar(45) NOT NULL,
    `apellidos` varchar(45) NOT NULL,
    `direccion` varchar(45) NOT NULL,
    `codigo_postal` int(5) NOT NULL,
    `id_municipio` varchar(4) NOT NULL,
    `telefono` int(9) NOT NULL,
    `fecha_nacimiento` date NOT NULL,
    `correo_electronico` varchar(50) DEFAULT NULL,
    PRIMARY KEY (`dni`),
    KEY `fk_clientes_municipios` (`id_municipio`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `detallepedido`;
CREATE TABLE IF NOT EXISTS `detallepedido` (
    `cod_detallepedido` int(45) NOT NULL,
    `cod_pedido` int(45) NOT NULL,
    `cod_plato` varchar(45) NOT NULL,
    `cantidad` int(11) NOT NULL,
    PRIMARY KEY (`cod_detallepedido`),
    KEY `fk_detallepedido_platos` (`cod_plato`),
    KEY `fk_detallepedido_pedidos` (`cod_pedido`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `login`;
CREATE TABLE IF NOT EXISTS `login` (
    `dni` varchar(9) NOT NULL,
    `password` varchar(45) NOT NULL,
    PRIMARY KEY (`dni`,`password`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `municipios`;
CREATE TABLE IF NOT EXISTS `municipios` (
    `id_municipio` varchar(4) NOT NULL,
    `nombre` varchar(45) NOT NULL,
    `id_repartidor_encargado` varchar(5) DEFAULT NULL,
    PRIMARY KEY (`id_municipio`),
    KEY `fk_municipios_repartidores` (`id_repartidor_encargado`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```

INSERT INTO `municipios` (`id_municipio`, `nombre`, `id_repartidor_encargado`) VALUES
('M001', 'Ajaelvir', 'R0002'),
('M002', 'Chapinería', 'R0001'),
('M003', 'Leganés', 'R0003'),
('M004', 'Perales de Tajuña', 'R0004'),
('M005', 'Madrid', 'R0003'),
('M006', 'Talamanca de Jarama', 'R0001'),
('M007', 'Tres Cantos', 'R0004'),
('M008', 'Getafe', 'R0005'),
('M009', 'Villarejo de Salvanés', 'R0005');

DROP TABLE IF EXISTS `pedidos`;
CREATE TABLE IF NOT EXISTS `pedidos` (
`dni` varchar(9) NOT NULL,
`cod_pedido` int(45) NOT NULL,
`fecha_pedido` datetime NOT NULL,
`fecha_entrega` datetime DEFAULT NULL,
`precio_total` double DEFAULT NULL,
PRIMARY KEY (`cod_pedido`),
KEY `fk_pedidos_clientes` (`dni`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `platos`;
CREATE TABLE IF NOT EXISTS `platos` (
`cod_plato` varchar(45) NOT NULL,
`nombre` varchar(45) DEFAULT NULL,
`descripcion` varchar(90) DEFAULT NULL,
`precio` double DEFAULT NULL,
PRIMARY KEY (`cod_plato`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

INSERT INTO `platos` (`cod_plato`, `nombre`, `descripcion`, `precio`) VALUES
('1', 'Pizza Peperoni', 'Pizza de peperoni', 10),
('2', 'Pizza 4 quesos', 'Pizza de cuatro quesos', 11),
('3', 'Pizza Barbacoa', 'Pizza de barbacoa', 15),
('4', 'Pizza Queso De Cabra', 'Pizza de queso de cabra', 13),
('5', 'Pizza Queso De Cabramelizada', 'Pizza de queso de cabra con cebolla caramelizada', 12),
('6', 'Pizza Jamon Y Queso', 'Pizza de jamon y queso', 9),
('7', 'Pizza Pepperoni', 'Pizza de pepperoni', 16),
('8', 'Arroz 3 delicias', 'Arroz con huevo guisantes y jamon', 5),
('9', 'Arroz con calamares en su tinta', '', 10);

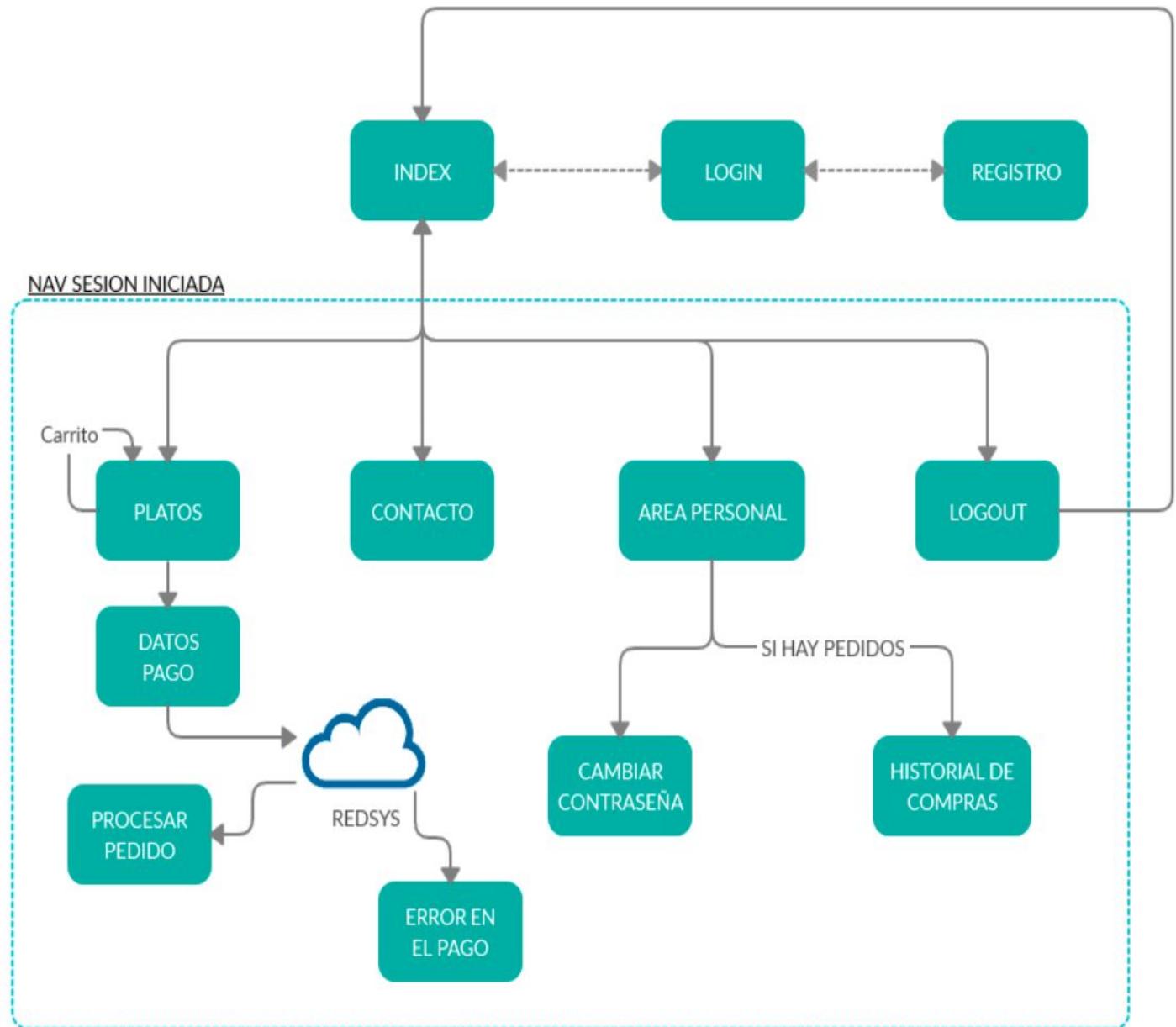
DROP TABLE IF EXISTS `repartidores`;
CREATE TABLE IF NOT EXISTS `repartidores` (
`id_repartidor` varchar(5) NOT NULL,
`nombre` varchar(45) NOT NULL,
`apellidos` varchar(45) NOT NULL,
`salario` varchar(45) NOT NULL,
PRIMARY KEY (`id_repartidor`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

INSERT INTO `repartidores` (`id_repartidor`, `nombre`, `apellidos`, `salario`) VALUES
('R0001', 'David', 'Perez Serrano', '1400'),
('R0002', 'Ana', 'Lopez Banana', '1600'),
('R0003', 'Elisa', 'Sanchez Romero', '1600'),
('R0004', 'Jose', 'Ruñ Sanchez', '1800'),
('R0005', 'Maria', 'Castejon Ruiz', '1400');
COMMIT;

```

## 5. Mapa de navegación.

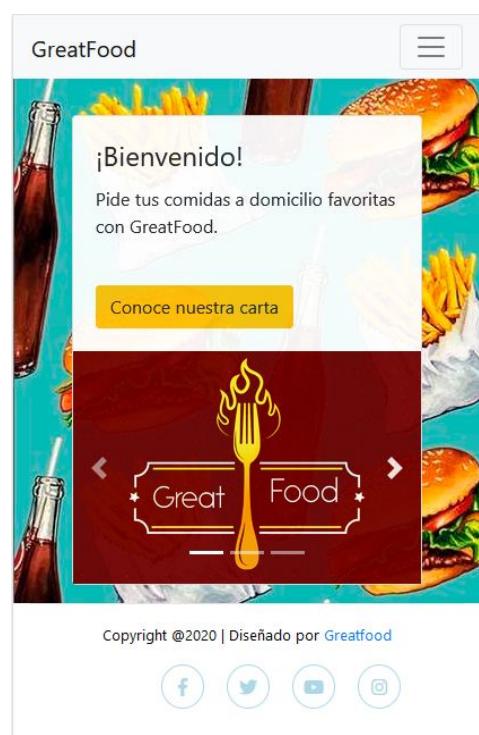
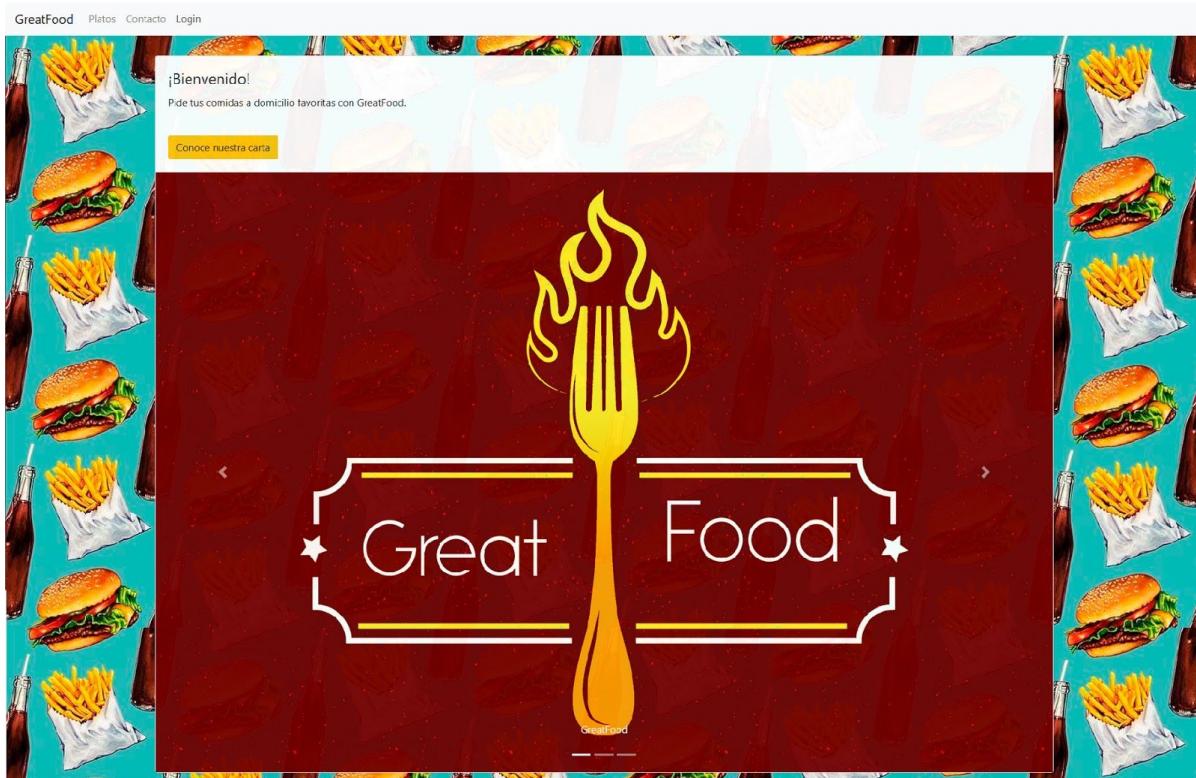
Un mapa de navegación es una representación gráfica de la estructura de una web. La siguiente imagen muestra cómo está organizada la web de Greatfood.



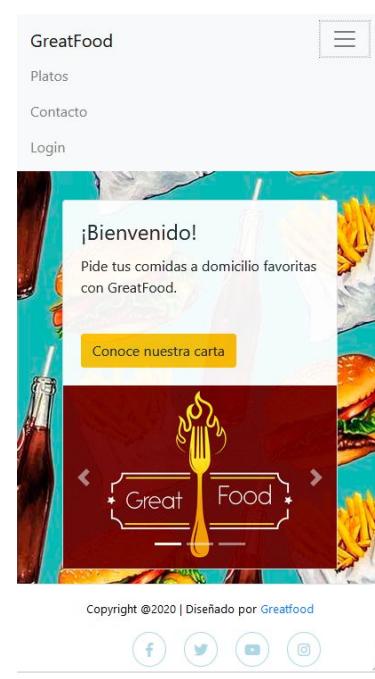
## 6. Código de los módulos y captura de pantallas.

En este apartado vamos a explicar las diferentes secciones (módulos) de nuestra web, incluyendo parte del código, además de su respectiva explicación.

### Index



Vista móvil



Copyright @2020 | Diseñado por Greatfood



```

<?php
    session_start();
    require_once("./db/db.php");
    $_SESSION['fallo'] = "no";

    if (isset($_SESSION['dni'])) {
?>
        <a class="nav-link" href="./controllers/area_personal.php" tabindex="-1">
            Area personal
        </a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="./controllers/logout.php" tabindex="-1">
            Logout
        </a>
    </li>
</?php

} else {
    $past = time() - 3600;
    foreach ($_COOKIE as $key => $value) {
        if ($key != 'PHPSESSID') {
            setcookie($key, $value, $past, '/');
        }
    }
?>
    <a class="nav-link" href="./controllers/login.php" tabindex="-1">
        Login
    </a>
</li>

<?php
}
?>

```

```

<footer class="footer">
    <div class="container">
        <hr>
        <!--Footer_bottom_ul_amrc ends here-->
        <p class="text-center">Copyright @2020 | Diseñado por <a href="#">Greatfood</a></p>

        <ul class="social_footer_ul">
            <li><a href="http://facebook.com"><i class="fab fa-facebook-f"></i></a></li>
            <li><a href="http://twitter.com"><i class="fab fa-twitter"></i></a></li>
            <li><a href="http://youtube.com"><i class="fab fa-youtube"></i></a></li>
            <li><a href="http://instagram.com"><i class="fab fa-instagram"></i></a></li>
        </ul>
        <!--social_footer_ul ends here-->
    </div>
</footer>

```

```

<div id="car" class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#car" data-slide-to="0" class="active"></li>
    <li data-target="#car" data-slide-to="1"></li>
    <li data-target="#car" data-slide-to="2"></li>
  </ol>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption d-none d-md-block">
        <p>GreatFood</p>
      </div>
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <a class="carousel-control-prev" href="#" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Anterior</span>
  </a>
  <a class="carousel-control-next" href="#" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Siguiente</span>
  </a>
</div>

```

### Explicacion del codigo:

La página index nos sirve de plantilla para el resto, tiene un diseño responsive, gracias a bootstrap.

Determinamos que un usuario ha iniciado sesión cuando hay algo en la variable `$_SESSION['DNI']`, si no hay, no mostramos en el nav el logout y el área persona (captura 1).

Tenemos un footer que muestra las redes sociales (captura 2). Tanto el nav como el footer los usamos en todas las páginas de la web.

Por último, tenemos un div central donde ponemos el cuerpo y toda la información del módulo. En este caso, al ser el index, hemos puesto un carrusel de bootstrap con 3 imágenes (captura 3).

## Registro

Vista móvil

The image displays two versions of a registration form for "GreatFood".

**Left Screenshot (Desktop View):**

- Zona de Registro:** ¡Rellena tus datos y pide cuando quieras!
- DNI: [Input Field]
- Contraseña: [Input Field]
- Nombre: [Input Field]
- Apellidos: [Input Field]
- Dirección: [Input Field]
- Código Postal: [Input Field]
- Municipio: [Input Field]  
Ajalvir
- Teléfono: [Input Field]
- Fecha de Nacimiento: [Input Field]  
dd / mm / aaaa
- Correo Electrónico: [Input Field]
- Botón:** Regístrate

**Right Screenshot (Mobile View):**

- GreatFood** [Logo]
- Zona de Registro:** ¡Rellena tus datos y pide cuando quieras!
- DNI: [Input Field]
- Contraseña: [Input Field]
- Nombre: [Input Field]
- Apellidos: [Input Field]
- Dirección: [Input Field]
- Código Postal: [Input Field]
- Municipio: [Input Field]  
Ajalvir
- Teléfono: [Input Field]
- Fecha de Nacimiento: [Input Field]  
dd / mm / aaaa
- Correo Electrónico: [Input Field]
- Botón:** Regístrate

A la hora de cargar el select de los municipios hacemos una llamada Ajax desde javascript al servidor, responde con una cadena XML, la cual se carga en el formulario mediante DOM.

```
<script type="text/javascript">
var peticion1;

if (document.addEventListener) {
    window.addEventListener("load", iniciar);
} else if (document.attachEvent) {
    window.attachEvent("onload", iniciar)
}

function iniciar() {

    if (window.XMLHttpRequest) { // Mozilla, Safari, ...
        peticion1 = new XMLHttpRequest();
    } else if (window.ActiveXObject) { // IE
        peticion1 = new ActiveXObject("Microsoft.XMLHTTP");
    }

    if (document.addEventListener) {
        peticion1.addEventListener("readystatechange", muestra);
    } else if (document.attachEvent) {
        peticion1.attachEvent("onreadystatechange", muestra)
    }

    peticion1.open("GET", "cargarmunicipios.php", true);
    peticion1.setRequestHeader("Content-Type", "application/x-www-form-urlencoded"); //Esta para XML
    peticion1.send(null);
}
```

```
function muestra() {
    if (peticion1.readyState == 4) {
        if (peticion1.status == 200) {
            var respuestaXML = peticion1.responseXML;
            //alert(respuestaXML);
            var datos = respuestaXML.getElementsByTagName("datos").item(0);
            var misMunicipios = datos.getElementsByTagName("municipios").item(0);
            var misCódigos = datos.getElementsByTagName("códigos").item(0);
            var todosLosMunicipios = misMunicipios.getElementsByTagName("municipio");
            var todosLosCódigos = misCódigos.getElementsByTagName("codigo");
            var miselect = document.getElementById("miselect");
            for (var VnbIndice = 0; VnbIndice < todosLosMunicipios.length; VnbIndice++) {
                var mm = todosLosMunicipios.item(VnbIndice).textContent;
                var elemento = document.createElement("option");
                var valor = document.createTextNode(mm);
                elemento.appendChild(valor);
                elemento.setAttribute('value', todosLosCódigos.item(VnbIndice).textContent);
                miselect.appendChild(elemento);
            }
        }
    }
}
```

```
cargarmunicipios.php  x
1 <?php
2     header('Content-Type:text/xml');
3     $cadenaXML='<municipios><municipio>Ajalvir</municipio><municipio>Tetuán</municipio><municipio>Madrid</municipio><municipio>Getafe</municipio><municipio>Villarejo de Salvanés</municipio>';
4     $cadena2XML='<códigos><codigo>M001</codigo><codigo>M002</codigo><codigo>M007</codigo><codigo>M008</codigo><codigo>M009</codigo>';
5     $final='<datos>' . $cadenaXML . $cadena2XML . '</datos>';
6     echo $final;
7     ?>
8
```

```

1 <?php
2 session_start();
3 error_reporting(0);
4
5
6 require("../db/db.php");
7 require("../models/comprobarDNI.php");
8 require("../models/registro.php");
9 require("../recursos/correo/enviarCorreo.php");
10
11 $dnireal=comprobarDNI($conn);
12
13
14 if ($dnireal==null) {
15
16     $nombre=$_POST["nombre"];
17     $dni=$_POST["dni"];
18     $correo_electronico=$_POST["mail"];
19     $mensaje=$nombre.', su registro se ha comple
20

```

```

1 <?php
2 function comprobarDNI($conn) {
3
4     //$resultado=true;
5
6     $dni=$_POST['dni'];
7     $sql="SELECT dni as d FROM login WHERE dni='$dni'
8         ";
9     $result = mysqli_query($conn, $sql);
10    $fila = mysqli_fetch_assoc($result);
11    $dnireal = $fila['d'];
12
13    $conn=null;
14    return $dnireal;
15
16    //return $resultado;
17 }
18 ?>

```

Cuando enviamos el formulario comprobamos que no exista ya un usuario registrado con el mismo DNI que se ha introducido. Tras esto, procesamos el registro realizando los 2 inserts de la siguiente captura:

```

<?php
function procesar($conn){
    $dni=$_POST["dni"];
    $nombre=$_POST["nombre"];
    $apellidos=$_POST["apellidos"];
    $direccion=$_POST["direccion"];
    $cp=$_POST["cp"];
    $municipio=$_POST["municipio"];
    $telefono=$_POST["telefono"];
    $fecha_nac=$_POST["fecha_nac"];
    $password=$_POST["password"];
    $correo_electronico=$_POST["mail"];

    $sql="INSERT INTO
        clientes(dni,nombre,apellidos,direccion,codigo_postal,id_municipio,telefono,fecha_nacimiento,correo_electrónico)
    VALUES ('$dni','$nombre','$apellidos','$direccion','$cp','$municipio','$telefono','$fecha_nac',
'$correo_electronico')";

    if (mysqli_query($conn, $sql)) {
        $sql="INSERT INTO login(dni,password) VALUES ('$dni','$password')";

        mysqli_query($conn, $sql);

        echo "Registro Completado";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }

    mysqli_close($conn);
}

?>

```

Envío del correo con información del pedido realizado:

Registro completado en GreatFood Recibidos x



GreatFood <contacto.greatfood@gmail.com>

para mí ▾

jose, su registro se ha completado correctamente, gracias por usar GreatFood. Puedes iniciar sesión utilizando su DNI 12345678z y su contraseña.

Código:

```
enviarCorreo($correo_electronico,'Registro completado en GreatFood',$mensaje,$nombre);
```

```
<?php

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'src/Exception.php';
require 'src/PHPMailer.php';
require 'src/SMTP.php';

function enviarCorreo($hacia,$titulo,$contenido,$nombre){

    $mail = new PHPMailer;
    $mail->isSMTP();
    $mail->SMTPDebug = 0;
    $mail->Host = "smtp.gmail.com"; // use $mail->Host = gethostby
    $mail->Port = 587; // TLS only
    $mail->SMTPSecure = 'tls'; // ssl is deprecated
    $mail->SMTPAuth = true;
    $mail->Username = 'contacto.greatfood@gmail.com';
    $mail->Password = [REDACTED];
    $mail->setFrom('contacto.greatfood@gmail.com', 'GreatFood');
    $mail->addAddress($hacia, $nombre);
    $mail->Subject = $titulo;
    $mail->msgHTML($contenido); // $mail->msgHTML(file_get_content
referenced images to embedded,
    $mail->AltBody = 'HTML no soportado';
    // $mail->addAttachment('images/phpmailer_mini.png'); //Attach

    $mail->send();

}

?>
```

Función que nos permite enviar un correo a un determinado usuario cuando se crea la cuenta, nos permite poner un título, un nombre y un contenido.

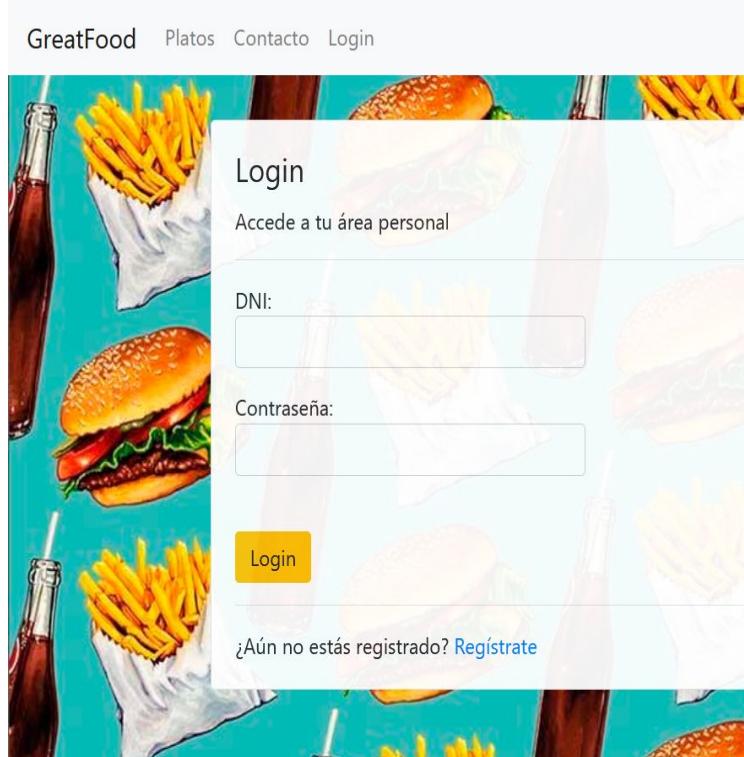
## Inserción de los datos de registro en las tablas de clientes y login de la base de datos

```
MariaDB [greatfood]> select * from login;
+-----+-----+
| dni   | password |
+-----+-----+
| 12345678Z | josetorres |
+-----+
1 row in set (0.000 sec)

MariaDB [greatfood]> select * from clientes;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| dni   | nombre | apellidos | direccion | codigo_postal | id_municipio | telefono | fecha_nacimiento | correo_electronico |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 12345678Z | jose   | torres    | Lugo      | 21577        | M007       | 614789455  | 1988-11-07    | josetorressmr@gmail.com |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

### Login:

Vista móvil



Código:

```
<div id="bienvenida">

    <div class="card">
        <div class="card-body ">
            <h4 class="card-title">Login</h4>
            <p class="card-text ">Accede a tu área personal</p>

            <form action="main.php" method="POST">
                <hr>
                <div class="form-group" id="login">
                    <p>DNI:<input type="text" name="dni" class="form-control" pattern="^[0-9]{8}[A-Za-z]{1}" required /></p>
                    <p>Contraseña:<input type="password" name="password" class="form-control" required /></p><br />
                    <input type="submit" value="Login" class="btn btn-warning " />
                </div>
            </form>
            <hr>
            <p class="card-text" style="display: inline;">¿Aún no estás registrado?</p>
            <a class="text-primary" href="registro.php" style="display: inline;">Regístrate</a>
        </div>
    </div>
</div>
```

```
<?php
session_start();

require("../db/db.php");
require("../models/comprobarUsuario.php");

if (comprobarUsuario($conn)) {
    header("location:../index.php");
    $_SESSION['fallo']="no";
} else {
    $_SESSION['fallo']="login";
    header("location:./login.php");
}
```

#### Explicación:

El módulo de login contiene un formulario sencillo donde se introducen el DNI y la contraseña.

Al enviar el formulario se comprueban los datos en la base de datos y si son correctos se redirecciona a la página index donde tendría acceso a todas las opciones debido a que con una variable de sesión verificamos si está un usuario logueado o no, y si es erróneo recarga la página

## Logout

ea personal Logout

```
<?php
session_start();
if(!isset($_SESSION['dni'])){
    echo "<script>console.log('no hay sesion en el logout')</script>";
    header("location:../index.php");
} else {
    echo "<script>console.log('hay sesion en logout')</script>";

    $past = time() - 3600;
    foreach ( $_COOKIE as $key => $value ) {
        if ($key != 'PHPSESSID') {
            setcookie( $key, $value, $past, '/' );
        }
    }

    session_unset();
    session_destroy();
    header("location:../index.php");
}
```

Al clicar en el módulo de logout, se ejecuta el siguiente código que realiza una limpieza tanto de las variables creadas como de las cookies y lo restaura todo desde cero.

Tras esto, vuelve a la página inicial, index, mediante un header.

## Platos

```
<?php
//Funcion que devuelve todos los platos
function getPlatos($conn) {
    $platos = [];
    $sql = "select * from platos";

    $resultado = mysqli_query($conn, $sql);
    while ($row = mysqli_fetch_assoc($resultado)) {
        $platos[] = $row;
    }
    return $platos;
}
```

Función que retorna un array con todos los platos que hay registrados en la base de datos en la tabla de platos

La próxima captura muestra el código sirve para mostrar el nombre de los platos y su precio en el carrito, ya que lo que realmente almacenamos es el ID del plato mediante cookies. Esto lo logramos mediante una conexión Ajax que devuelve un JSON con la información de los platos, la cual se procesa mediante Javascript para mostrar el carrito correctamente.

```
<script type="text/javascript">
var peticion2;

iniciar();

function iniciar() {
    if (window.XMLHttpRequest) { // Mozilla, Safari, ...
        peticion2 = new XMLHttpRequest();
    } else if (window.ActiveXObject) { // IE
        peticion2 = new ActiveXObject("Microsoft.XMLHTTP");
    }

    if (document.addEventListener) {
        peticion2.addEventListener("readystatechange", carga);
    } else if (document.attachEvent) {
        peticion2.attachEvent("onreadystatechange", carga);
    }

    peticion2.open("GET", "../controllers/getDatosCarrito.php", true);
    peticion2.setRequestHeader("Content-Type", "application/json");
    peticion2.send(null);

}
```

```

function carga() {
    if (peticion2.readyState == 4) {
        if (peticion2.status == 200) {
            //console.log(peticion2.response);
            var JSONPlatos = JSON.parse(peticion2.responseText);
            console.log(JSONPlatos);
            var cookies = document.cookie.split(";");
            if (cookies.length > 1) {
                $("#divcarrito").show("slow");

                //-----

                var cookies = document.cookie.split(";");
                //alert(cookies.length);
                var preciototal = 0;
                for (var i = 0; i < cookies.length; i++) {
                    var cookie = cookies[i];
                    var eqPos = cookie.indexOf("=");
                    var name = eqPos > -1 ? cookie.substr(0, eqPos) : cookie;
                    if (String(name) != "PHPSESSID") {
                        var idAux = String(name);

                        for (var clave in JSONPlatos.platos) {
                            if (JSONPlatos.platos.hasOwnProperty(clave)) {
                                var a = parseInt(JSONPlatos.platos[clave].codplato);
                                var b = parseInt(idAux);
                                if (a == b) {
                                    $("#listcarrito").append(
                                        '<li class="list-group-item d-flex justify-content-between align-items-center">' +
                                        JSONPlatos.platos[clave].nombre +
                                        " " + JSONPlatos.platos[clave].precio + '€ <span class="badge badge-info badge-pill">' + getCookie(idAux) + '</span></li>');
                                }
                            }
                        }
                    }
                }
                preciototal = parseInt(preciototal) + (parseInt(JSONPlatos.platos[clave].precio) * parseInt(getCookie(idAux)));
            }
        }
        setCookie("pTotal", preciototal, 2);
        $("#buttonaux").before('<p>Total: ' + preciototal + '€</p>');
        //-----
    }
}

}

```

## Formulario Carrito

GreatFood Platos Contacto Área personal Logout



Código:

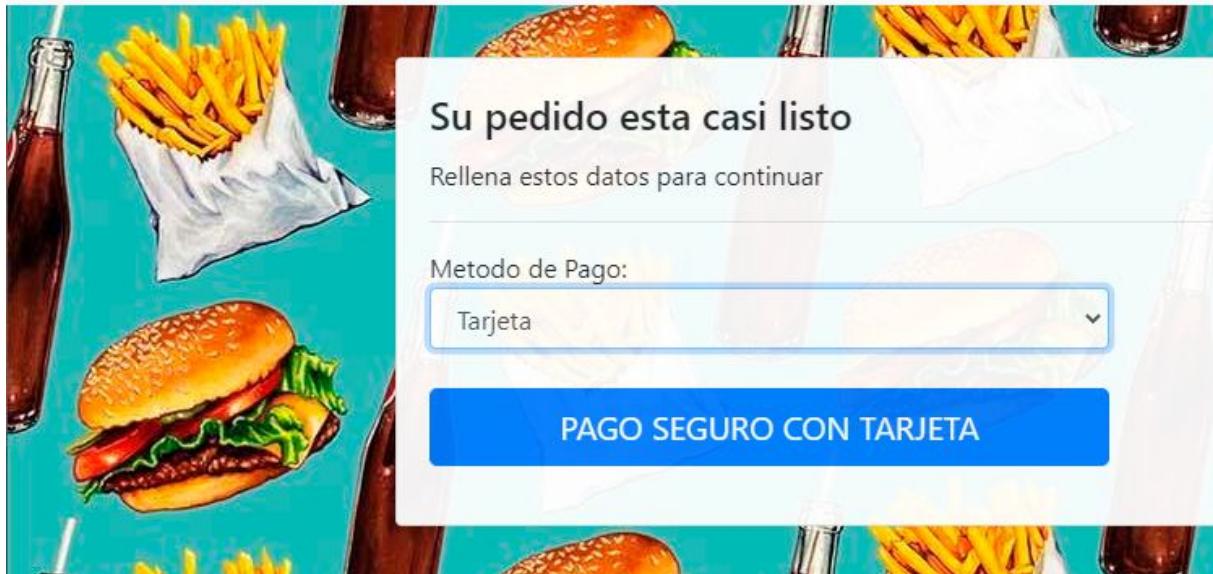
```
<form action="tarjeta_intermedio.php" method="POST" id="formfecha">
<hr>
    <div class="form-group" id="login">
        Fecha Entrega: <input type="date" <?php
            $date_format = 'Y-m-d';
            $tomorrow = time() + (1 * 24 * 60 * 60);
            echo "min=\"".gmdate($date_format, $tomorrow)."\"";
        ?> name="fecha_entrega" class="form-control" id="formfechaentrega" required><br>

        <input type="submit" value="Continuar" name="confirmar" class="btn btn-warning ">
    </div>
</form>
```

Variable que se actualizan con la hora actual y no permite seleccionar una fecha menor a igual a la de mañana.

## Tarjeta Intermedio

GreatFood Platos Contacto Area personal Logout



Código:

```
<div class="card">
    <div class="card-body ">
        <h4 class="card-title">Su pedido esta casi listo</h4>
        <p class="card-text " >Rellena estos datos para continuar</p>

        <!--<form action="" method="POST">-->
        <form action="https://sis-i.redsys.es:25443/sis/realizarPago" method="POST" id="formfecha">
            <hr>
            <div class="form-group" id="login">

                Metodo de Pago:
                <select id="pago" name="pago" class="form-control" required>
                    <option value="Tarjeta">Tarjeta</option>
                    <option value="PayPal" disabled>PayPal</option>
                    <option value="Bizum" disabled>Bizum</option>
                </select><br>
                <input type="hidden" name='Ds_SignatureVersion' value='<?php echo $version; ?>'>
                <input type="hidden" name='Ds_MerchantParameters' value='<?php echo $params; ??>'>
                <input type="hidden" name='Ds_Signature' value='<?php echo $signature; ?>'>
                <input class="btn btn-lg btn-primary btn-block" type="submit" name="submitPayment" value="PAGO SEGURO CON TARJETA" />
            </div>
        </form>
    </div>
```

## Formulario con variables predefinidas para el correcto uso del TPV Virtual

```
<?php
    session_start();
    require_once("../db/db.php");

    include "apiRedsys.php";
    $miObj = new RedsysAPI;

    // Valores de entrada que no hemos cambiado para ningun ejemplo
    $name = 'GreatFood';

    $fuc="999008881";
    $terminal="1";
    $moneda="978";
    $trans="0";
    $url="http://daw.com/";
    $urlOK="http://daw.com/controllers/procesar_carrito.php";
    $urlKO="http://daw.com/views/resultado_carritoKO.php";
    $id=time();
    $amount=((int)$_SESSION['preciototal'])*100;

    // Se Rellenan los campos
    $miObj->setParameter("DS_MERCHANT_AMOUNT",$amount);
    $miObj->setParameter("DS_MERCHANT_ORDER",$id);
    $miObj->setParameter("DS_MERCHANT_MERCHANTCODE",$fuc);
    $miObj->setParameter("DS_MERCHANT_CURRENCY",$moneda);
    $miObj->setParameter("DS_MERCHANT_TRANSACTIONTYPE",$trans);
    $miObj->setParameter("DS_MERCHANT_TERMINAL",$terminal);
    $miObj->setParameter("DS_MERCHANT_MERCHANTURL",$url);
    $miObj->setParameter("DS_MERCHANT_MERCHANTNAME",$name);
    $miObj->setParameter("DS_MERCHANT_URLOK",$urlOK);
    $miObj->setParameter("DS_MERCHANT_URLKO",$urlKO);

    //Datos de configuración
    $version="HMAC_SHA256_V1";
    $kc = 'sq7HjrU0BfKmC576ILgskD5srU870gJ7';//Clave recuperada de CANALES
    // Se generan los parámetros de la petición
    $request = "";
    $params = $miObj->createMerchantParameters();
    $signature = $miObj->createMerchantSignature($kc);
```

Las variables urlOK y urlKO contienen la ruta a la que deberá redirigir si se ha realizado correctamente el pago, o si no.

## Página de Pago Virtual

The screenshot shows the Redsys Virtual Payment page. At the top, there is a navigation bar with the Redsys logo and a language selection dropdown set to 'Castellano'. Below this is a progress bar with four steps: 1. Seleccione método de pago (Select payment method), 2. Comprobación autenticación (Authentication check), 3. Solicitando Autorización (Authorization request), and 4. Resultado Transacción (Transaction result). The main content area is divided into sections:

- Datos de la operación**:
  - Importe: 10,00 €
  - Comercio: GreatFood (ESPAÑA)
  - Terminal: 999008881-1
  - Pedido: 1590772983
  - Fecha: 29/05/2020 19:24
- Pagar con Tarjeta**:
  - Nº Tarjeta:
  - Caducidad:  mm  aa
  - Cancelar
- Pagar con Tarjeta UnionPay**:
  -
- Otras formas de pago**:
  -

## Página de Confirmación del Pago



Esta es la Frase de Control que usted definió para identificar al Servidor de Autenticación

123456

A continuación le presentamos los datos de la operación de compra para la que se nos está solicitando autorización:

Nº Tarjeta	454881*****0004
Comercio	GreatFood
Descripción Producto	
Importe Operación	10.00 Euros
Fecha y Hora	29/05/2020 17:25:43

Rogamos informe su Código de Identificación Personal - CIP - en el siguiente apartado y pulse Aceptar para que el Servidor de Autenticación verifique su identidad.

Código de Identificación Personal CIP

.....

## Compra Finalizada

GreatFood Platos Contacto Área personal Logout



Código:

```
<div id="bienvenida">

    <div class="card">
        <div class="card-body ">
            <h4 class="card-title">Carrito procesado con éxito.</h4>
            <hr>
            <?php
                echo $_SESSION['resultadocarrito'];
            ?>
            <script type="text/javascript">
                deleteAllCookies2();
            </script>
            <br><br>
            <a class="btn btn-warning" href="../../index.php"> Volver a inicio </a>
        </div>
    </div>

</div>
```

Una vez se ha procesado la compra, se asegura de eliminar la cookie que recoge los pedidos que se van comprando.

Si la compra se ha realizado correctamente, se mandará un correo con los detalles del pedido realizado.

 GreatFood <contacto.greatfood@gmail.com>

para mí ▾

jose, su pedido se ha completado correctamente. Su repartidor@ Ana Lopez Banana entregará su pedido el 2020-05-30, su pedido es:

Numero de pedido	Fecha Pedido	Fecha de Entrega	Precio total
4	2020-05-29 19:27:27	2020-05-30 00:00:00	10
#	Nombre	Cantidad	Precio
#	Pizza Peperoni	1	10

Código:

```
$fechaentrega=$_SESSION['fecha_entrega'];
$dni = $_SESSION['dni'];
$_SESSION['resultadocarrito']=procesarCarrito($conn,$fechaentrega,$dni);

$array = mirar_historial($conn);

$_SESSION['array1']=$array[0];
$_SESSION['array2']=$array[1];

$nombre=$_SESSION["nombre"];
$dni=$_SESSION["dni"];
$correo_electronico=$_SESSION["correo"];
$repartidor=$_SESSION["resultadocarrito"];
$mensaje=$nombre.', su pedido se ha completado correctamente. '.$repartidor.', su pedido es: ';
$mensaje .='<html><body><table border="1">
| Numero de pedido | Fecha Pedido | Fecha de Entrega | Precio total |
| --- | --- | --- | --- |
| ' . $_SESSION['array1'][$i]['cod_pedido'] . ' | ' . $_SESSION['array1'][$i]['fecha_pedido'] . ' | ' . $_SESSION['array1'][$i]['fecha_entrega'] . ' | ' . $_SESSION['array1'][$i]['precio_total'] . ' |
| # | Nombre | Cantidad | Precio |

';
$i=sizeof($_SESSION["array1"])-1;

$mensaje .='<tr>
<td>' . $_SESSION['array1'][$i]['cod_pedido'] . '</td>
<td>' . $_SESSION['array1'][$i]['fecha_pedido'] . '</td>
<td>' . $_SESSION['array1'][$i]['fecha_entrega'] . '</td>
<td>' . $_SESSION['array1'][$i]['precio_total'] . '</td>
</tr>

<tr>
<th>#</th>
<th>Nombre</th>
<th>Cantidad</th>
<th>Precio</th>
</tr>';
for ($j = 0; $j < sizeof($_SESSION['array2']); $j++) {
if ($_SESSION['array2'][$j]['cod_pedido'] == $_SESSION['array1'][$i]['cod_pedido']) {

$mensaje .='<tr>
<td>' . $_SESSION['array2'][$j]['nombre'] . '</td>
<td>' . $_SESSION['array2'][$j]['cantidad'] . '</td>
<td>' . $_SESSION['array2'][$j]['precio'] . '</td>
</tr>';
}
}
$mensaje .='</tbody></table></body></html>';

enviarCorreo($correo_electronico,'Pedido completado en GreatFood',$mensaje,$nombre);
$_SESSION['fallido']="no";
```

Función que recorre el historial y recoge el último pedido para así mandar el correo con la información del pedido que se acaba de efectuar.

### Inserción del pedido en la base de datos

```
MariaDB [greatfood]> select * from pedidos;
+-----+-----+-----+-----+-----+
| dni  | cod_pedido | fecha_pedido | fecha_entrega | precio_total |
+-----+-----+-----+-----+-----+
| 12345678Z | 1 | 2020-06-01 09:17:52 | 2020-06-02 00:00:00 | 10 |
+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

### Error en el pedido



Código:

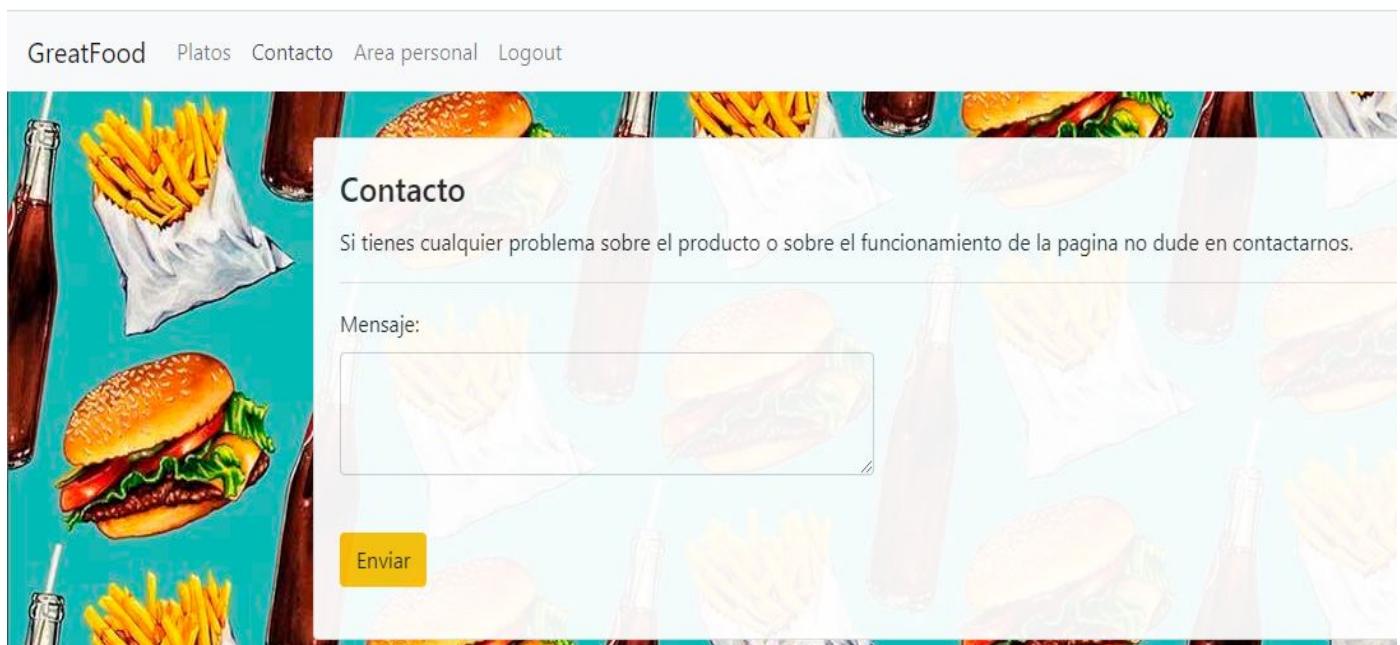
```
<div id="bienvenida">

    <div class="card">
        <div class="card-body ">
            <h4 class="card-title">No se ha podido procesar tu pedido.</h4>
            <hr>
            Revisa tus datos de pago.
            <script type="text/javascript">
                deleteAllCookies2();
            </script>
            <br><br>
            <a class="btn btn-warning" href="../index.php"> Volver a inicio </a>
        </div>
    </div>

</div>
```

Muestra por pantalla el error al procesar el pedido y borra el carrito que contenía el pedido que deseaba realizar

## Contacto



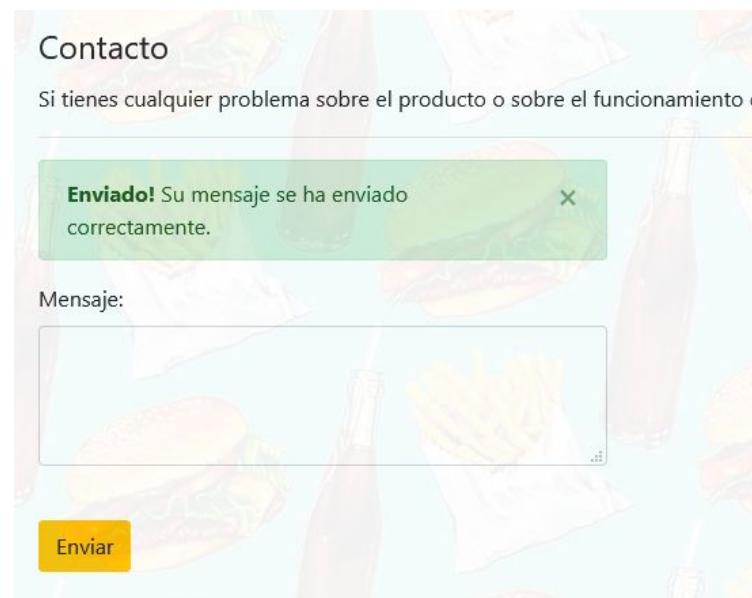
GreatFood Platos Contacto Area personal Logout

### Contacto

Si tienes cualquier problema sobre el producto o sobre el funcionamiento de la pagina no dude en contactarnos.

Mensaje:

Enviar



Contacto

Si tienes cualquier problema sobre el producto o sobre el funcionamiento c

**Enviado!** Su mensaje se ha enviado correctamente.

Mensaje:

Enviar

Código:

```
<h4 class="card-title">Contacto</h4>
    Si tienes cualquier problema sobre el producto o sobre el funcionamiento de la pagina no dude en contactarnos.
    <hr>

    <form action="#" method="POST">
        <div class="form-group" id="login">

            <div class="form-group blue-border-focus">
                <?php
                if ($_POST!=null) {
                    if (isset($_POST["enviar"]) && !empty($_POST["consulta"])) {
                ?>
                    <div class="alert alert-success alert-dismissible">
                        <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
                        <strong>Enviado!</strong> Su mensaje se ha enviado correctamente.
                    </div>
                <?php
                }
            }
            ?>
            <label for="exampleFormControlTextarea5">Mensaje:</label>
            <textarea class="form-control" id="exampleFormControlTextarea5" rows="3" name="consulta"></textarea>
        </div>
        <br>
        <input type="submit" value="Enviar" name="enviar" class="btn btn-warning " />
    </div>
</form>
```

Formulario que sólo recoge el mensaje de la consulta pues, la información general de el usuario que está enviando la consulta está recogida en variables de sesión y por eso no debe introducirlas.

#### Envío del correo con la consulta realizada

```
<?php
if ($_POST != null) {
    if (isset($_POST["envio"]) && !empty($_POST["consulta"])) {
        $nombre = $_SESSION["nombre"];
        $consulta = $_POST["consulta"];
        $correo_electronico = 'contacto.greatfood@gmail.com';
        $mensaje = '<html><body><h4>' . $nombre .
        ' ha enviado la siguiente consulta:</h4> <br><br>' . $consulta . '</body></html>';

        enviarCorreo($correo_electronico, 'Consulta ' . $nombre, $mensaje, $nombre);
    }
}
?>
```

Correo enviado con la consulta enviada

## Consulta jose Recibidos x



GreatFood <contacto.greatfood@gmail.com>  
para mí ▾

jose(josetorressmr@gmail.com) ha enviado la siguiente consulta:

Mensaje de Prueba Contacto GreatFood

Responder

Reenviar

Código:

```
<?php
if ($_POST!=null) {
    if (isset($_POST["enviar"]) && !empty($_POST["consulta"])) {
        $nombre=$_SESSION["nombre"];
        $consulta=$_POST["consulta"];
        $correo_electronico='contacto.greatfood@gmail.com';
        $mensaje='<html><body><h4>'.$nombre.'('.$_SESSION['correo'].') ha enviado la siguiente consulta:</h4> <br><br>'.$consulta.'</body></html>';
        enviarCorreo($correo_electronico,'Consulta '.$nombre,$mensaje,$nombre);

    }
}
?>
```

Envío del correo con el correo destinatario y los datos y consulta del usuario que lo manda.

## Área Personal



Código:

```
<h4 class="card-title">Area personal</h4>
<hr>
<a href="cambiarContraseña.php" class="btn btn-warning ">Cambiar Contraseña</a><br><br>
<?php
if ($_SESSION['resDNI']==$_SESSION['dni']) {
    ?>
<a href="historial_compras.php" class="btn btn-warning ">Historial Pedidos</a>
```

Si el usuario no tiene ningún pedido, no se muestra el botón del historial.

## Historial de Pedidos

Numero de pedido	Fecha Pedido	Fecha de Entrega	Precio total
4	2020-05-29 19:27:27	2020-05-30 00:00:00	10
#	Nombre	Cantidad	Precio
#	Pizza Peperoni	1	10

Código:

```
<?php
    require_once("../db/db.php");
    session_start();
    require_once("../models/historial.php");

    $array = mirar_historial($conn);

    $_SESSION['array1']=$array[0];
    $_SESSION['array2']=$array[1];
    require_once("../views/historial.php");
?>
```

Se crean 2 arrays cada uno con la información de una tabla (detallepedido y pedidos)

```
<?php

function mirar_historial($conn)
{
    $sql = "select * from pedidos where dni='$_SESSION[dni]'";
    $resultado = mysqli_query($conn, $sql);
    while ($row = mysqli_fetch_assoc($resultado)) {
        $array1[] = $row;
    }

    $sql2 =
"select detallepedido.cod_pedido,platos.nombre,detallepedido.cantidad, platos.precio from detallepedido,platos where detallepedido.cod_plato=platos.cod_plato AND cod_pedido in (select cod_pedido from pedidos where dni='$_SESSION[dni]' )";

    $resultado = mysqli_query($conn, $sql2);
    while ($row = mysqli_fetch_assoc($resultado)) {
        $array2[] = $row;
    }

    $final[0] = $array1;
    $final[1] = $array2;
    return $final;
}

?>
```

### Explicación:

Recorre los detallepedidos del usuario y recopila la información para llenar una tabla donde se visualiza el historial de pedidos del usuario

```
<?php

function comprobarPedidos($conn,$dni){


    $sql="select dni as p from pedidos where dni='".$dni."'";
    $result = mysqli_query($conn, $sql);
    $fila = mysqli_fetch_assoc($result);
    $dni = $fila['p'];

    if ($dni==null) {
        $dni="123456789023456789";
    }
    return $dni;

}
```

#### Explicación:

Función que comprueba si hay un pedido realizado para así tener que mostrar en el módulo de área personal el botón que muestra el historial de pedidos o no.

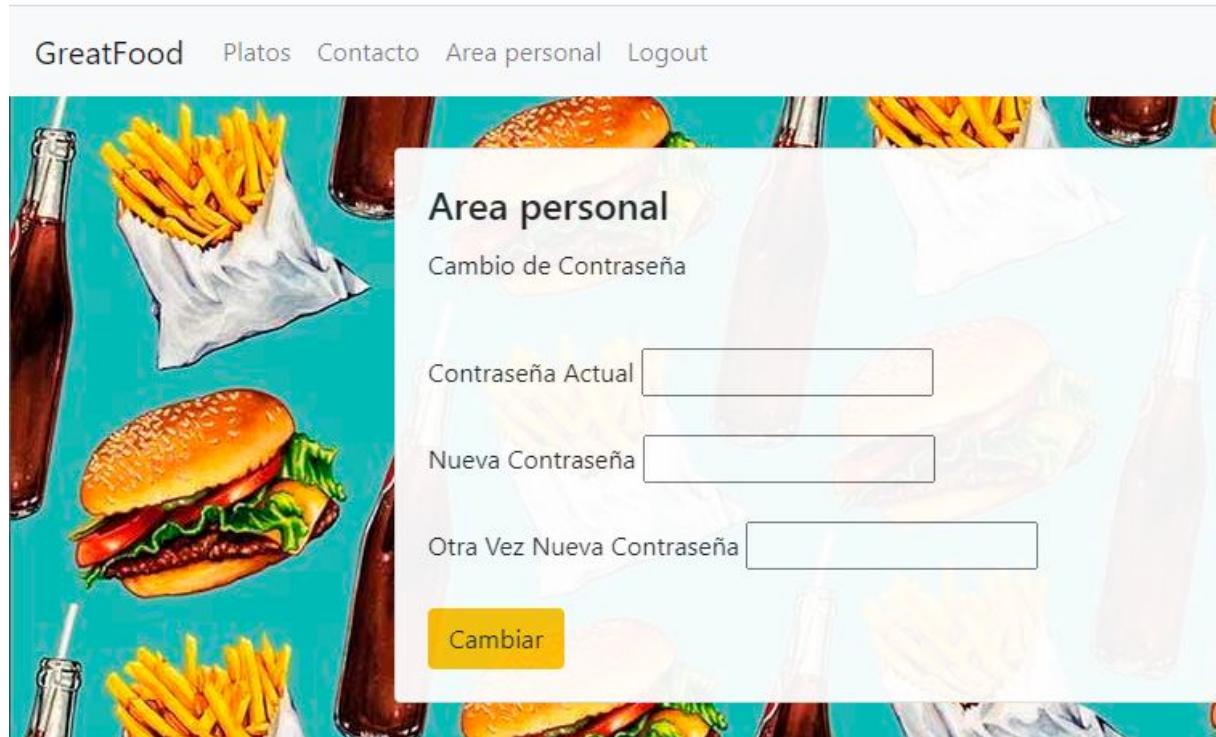
```

<h4 class="card-title">Historial</h4>
<p class="card-text ">Consulte sus pedidos</p>
<?php
?>
<table class="table table-dark table-bordered">
<thead>
<tr>
<th scope="col">Número de pedido</th>
<th scope="col">Fecha Pedido</th>
<th scope="col">Fecha de Entrega</th>
<th scope="col">Precio total</th>
</tr>
</thead>
<tbody>
<?php
//Aqui debajo irian el historial
//var_dump($array);
for ($i = 0; $i < sizeof($_SESSION["array1"]); $i++) {
echo "<script>console.log(" . $_SESSION["array1"][$i]["cod_pedido"] . ")</script>";
echo "<tr>
<th class='bg-primary' scope='row'>" . $_SESSION["array1"][$i]["cod_pedido"] . "</th>
<td class='bg-primary'>" . $_SESSION["array1"][$i]["fecha_pedido"] . "</td>
<td class='bg-primary'>" . $_SESSION["array1"][$i]["fecha_entrega"] . "</td>
<td class='bg-primary'>" . $_SESSION["array1"][$i]["precio_total"] . "</td>
</tr>
<tr>
<th class='bg-info' scope='col'>#</th>
<th class='bg-secondary' scope='col'>Nombre</th>
<th class='bg-secondary' scope='col'>Cantidad</th>
<th class='bg-secondary' scope='col'>Precio</th>
</tr>";
for ($j = 0; $j < sizeof($_SESSION["array2"]); $j++) {
if ($_SESSION["array2"][$j]["cod_pedido"] == $_SESSION["array1"][$i]["cod_pedido"]) {
echo "<tr>
<th class='bg-info' scope='row'>#"</th>
<td class='bg-info'>" . $_SESSION["array2"][$j]["nombre"] . "</td>
<td class='bg-info'>" . $_SESSION["array2"][$j]["cantidad"] . "</td>
<td class='bg-info'>" . $_SESSION["array2"][$j]["precio"] . "</td>
</tr>";
}
}
}
?>
</tbody>
</table>

```

Visualiza en el módulo los datos recopilados de la base de datos para mostrar el historial de pedidos del usuario

## Cambio de Contraseña



Código:

```
<div id="bienvenida">
    <div class="card">
        <div class="card-body ">
            <h4 class="card-title">Área personal</h4>
            <p class="card-text ">Cambio de Contraseña</p><br>
            <form action="" method="post">
                Contraseña Actual <input type="password" id="passwordVieja" name="passwordVieja"><br><br>
                Nueva Contraseña <input type="password" id="passwordNueva" name="passwordNueva" pattern="[A-Za-z0-9!?-]{6,12}"><br><br>
                Otra Vez Nueva Contraseña <input type="password" id="passwordNueva2" name="passwordNueva2" pattern="[A-Za-z0-9!?-]{6,12}"><br><br>
                <input name="cambiar" type="submit" value="Cambiar" class="btn btn-warning">
            <?php
                if ($_POST != null) {
                    if (isset($_POST["cambiar"])) {
                        require("../models/cambiarContraseña.php");
                    }
                }
            ?>
            </form>
        </div>
    </div>
</div>
```

Formulario que recoge el cambio de contraseña, validando la contraseña actual, para saber si es el usuario real o no, y la contraseña nueva dos veces por alguna posible equivocación

Código:

```
<?php
    $contraseñaActual=$_POST['passwordVieja'];
    $contraseñaNueva=$_POST['passwordNueva'];
    $dni=$_SESSION['dni'];
    $sql="SELECT password FROM login WHERE dni='$dni'";
    $result = mysqli_query($conn, $sql);
    $fila = mysqli_fetch_assoc($result);
    $passwordReal = $fila['password'];

    if($contraseñaActual==$passwordReal){

        $sql2="UPDATE login SET password='$contraseñaNueva' WHERE dni='$dni'";
        $_SESSION['passwordFail']=false;
        mysqli_query($conn, $sql2);

    } else {
        $_SESSION['passwordFail']=true;
        header("location:../controllers/cambiarContraseña.php");
    }
    mysqli_close($conn);
?>
```

Explicación:

Comprobación de si en contraseña actual es realmente la contraseña registrada en la base de datos por ese usuario, y si es cierta su posterior cambio.

## 7. Juego de ensayo y pruebas.

En este apartado comprobamos que funcionen todos los posibles casos, tanto los normales como los fallos controlados.

Al entrar el la página se puede acceder a todos los distintos módulos pero de una manera limitada, puesto que no tenemos contemplado que se puedan realizar pedidos sin estar logueados. Ej: Al acceder a el módulo de platos, éste muestra la variedad de comidas que están disponibles.

GreatFood Platos Contacto Login

Necesario [iniciar sesión/registrarse](#) Para añadir platos al carrito.

Platos

Pizza Peperoni Precio: 10€

Pizza 4 quesos Precio: 11€

Pizza Barbaco Precio: 15€

Si el usuario inicia sesión se vería así.

GreatFood Platos Contacto Área personal Logout

Platos

Pizza Peperoni Precio: 10€

Pizza 4 quesos Precio: 11€

Pizza Barbaco Precio: 15€

Añadir Añadir Añadir

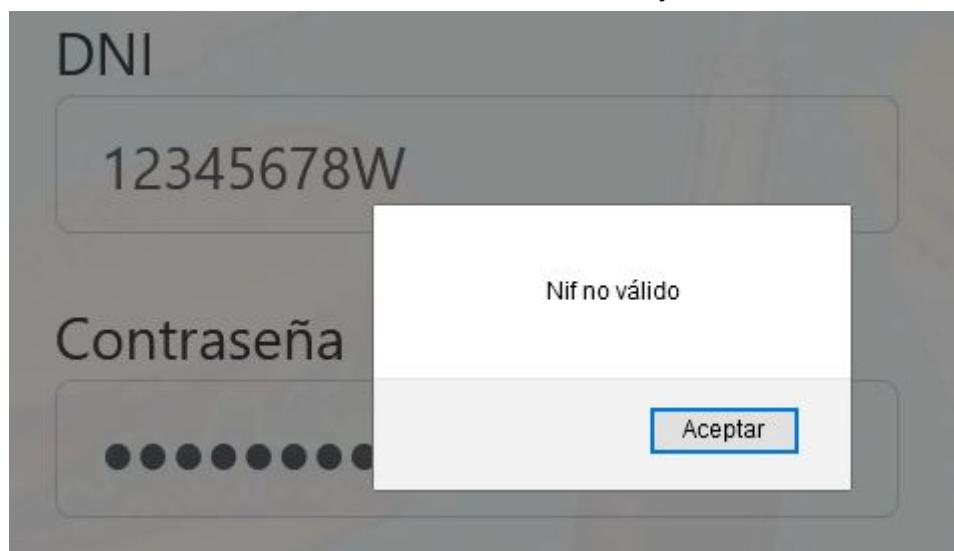
A la hora de registrarse, el usuario debe llenar varios campos con sus datos, estas son las diferentes validaciones:

#### DNI

Al registrarse hay que introducir un DNI válido, es decir, 8 números y 1 letra al final.

The screenshot shows a registration form titled "Zona de Registro" with a background image of fast food. The "DNI" field contains "1F", which is highlighted in red. A validation message box appears, stating "Haz coincidir el formato solicitado." and "Formato de DNI incorrecto". Other fields like "Nombre" (asdfghj), "Apellidos" (hjfghj), and "Direccion" (ghjkghjk) are also visible.

Si introducimos una letra no válida salta un mensaje.



## Zona de Registro

¡Rellena tus datos y pide cuando quieras!

DNI  
06030823Q

Contraseña  
•

**!** Haz coincidir el formato solicitado.  
Tu contraseña debe de tener entre 6 y 12 caracteres

Apellidos:  
a

Dirección:  
a

La única restricción que tiene para poner el código postal es que tiene que ser de 5 dígitos.

Apellidos:  
a

Dirección:  
a

Código Postal  
a

**!** Haz coincidir el formato solicitado.  
Ajalvir

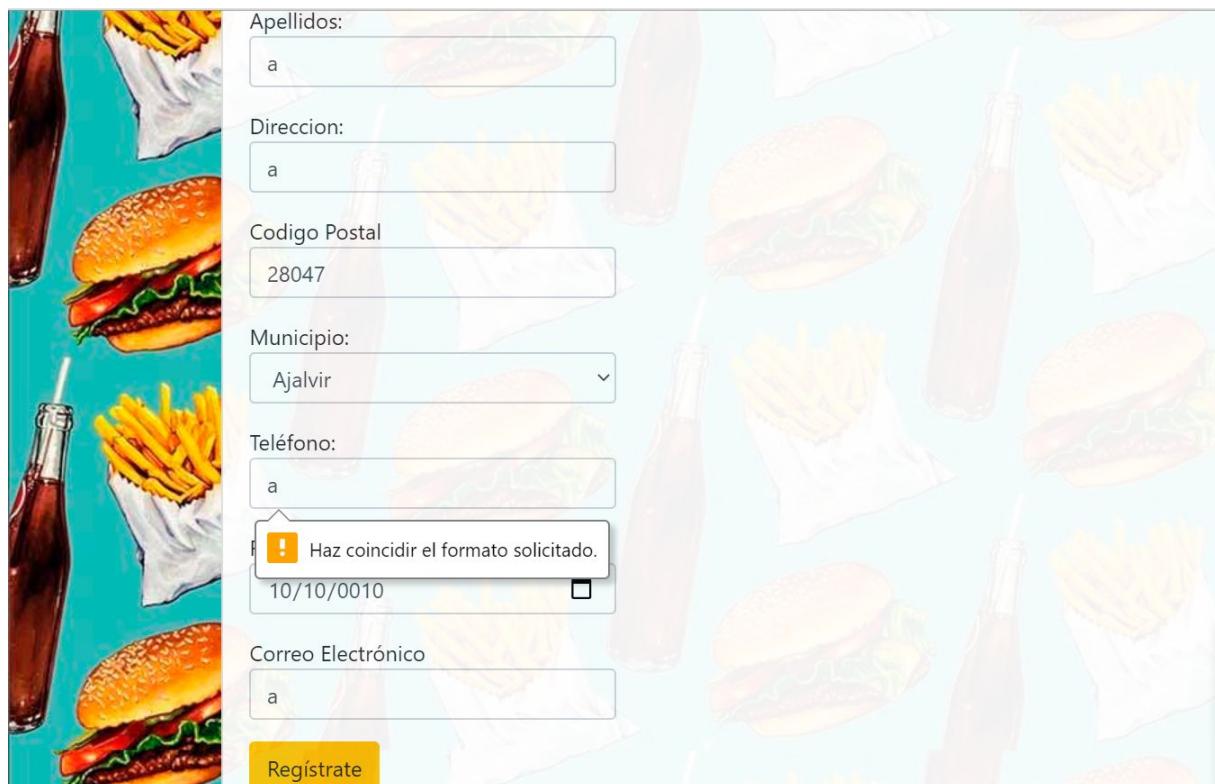
Teléfono:  
a

Fecha de Nacimiento:  
10/10/0010

Correo Electrónico  
a

Regístrate

Para introducir un número de teléfono correctamente la única restricción es que debe ser un número de 9 dígitos



Apellidos:

Dirección:

Código Postal:

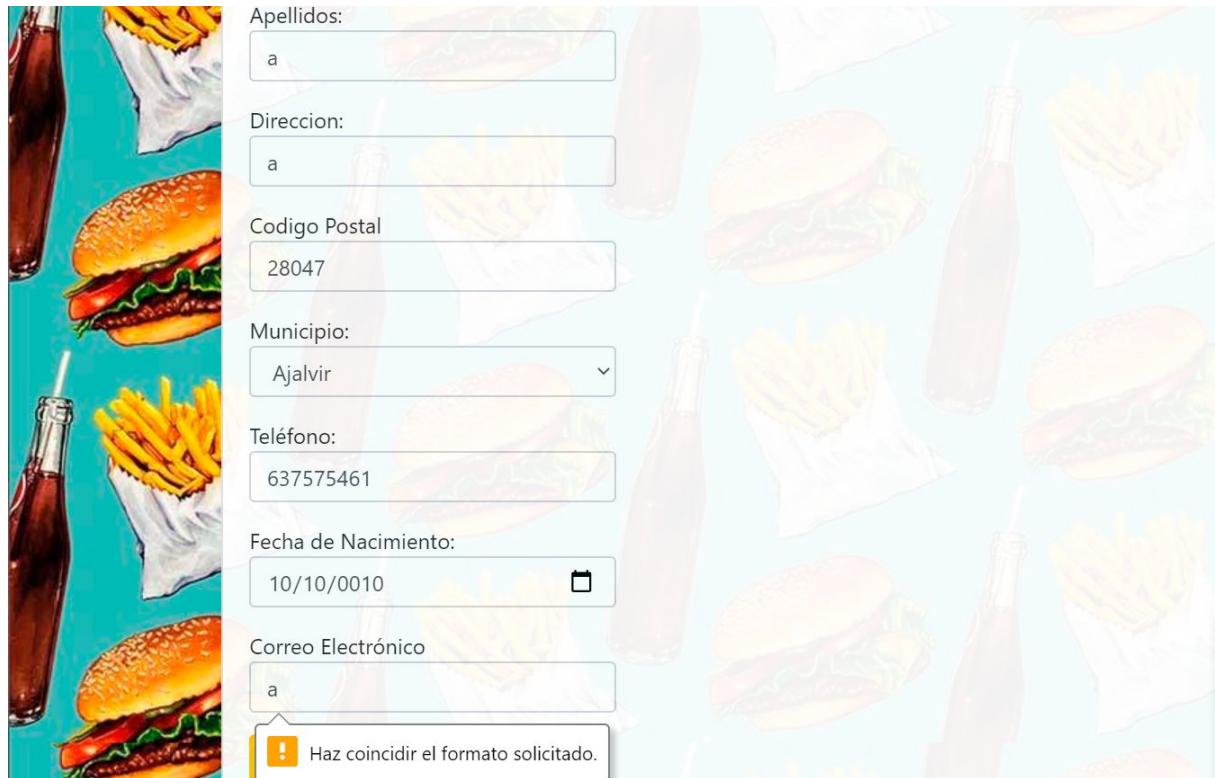
Municipio:

Teléfono:  
 Haz coincidir el formato solicitado.  
10/10/0010

Correo Electrónico:

```
<p>Teléfono:<input type="text" name="telefono"
    class="form-control" pattern="[679]{1}[0-9]{8}" required /></p>
```

La validación del correo electrónico es más compleja debido a que comprueba que tiene que tener al final un punto ej:`greatfood@great.es/.com/.net`, un `@` y que seguidamente vaya una cadena indeterminada que al final contenga un punto `greatfood@great.es`, etc...



Apellidos:

Dirección:

Código Postal:

Municipio:

Teléfono:

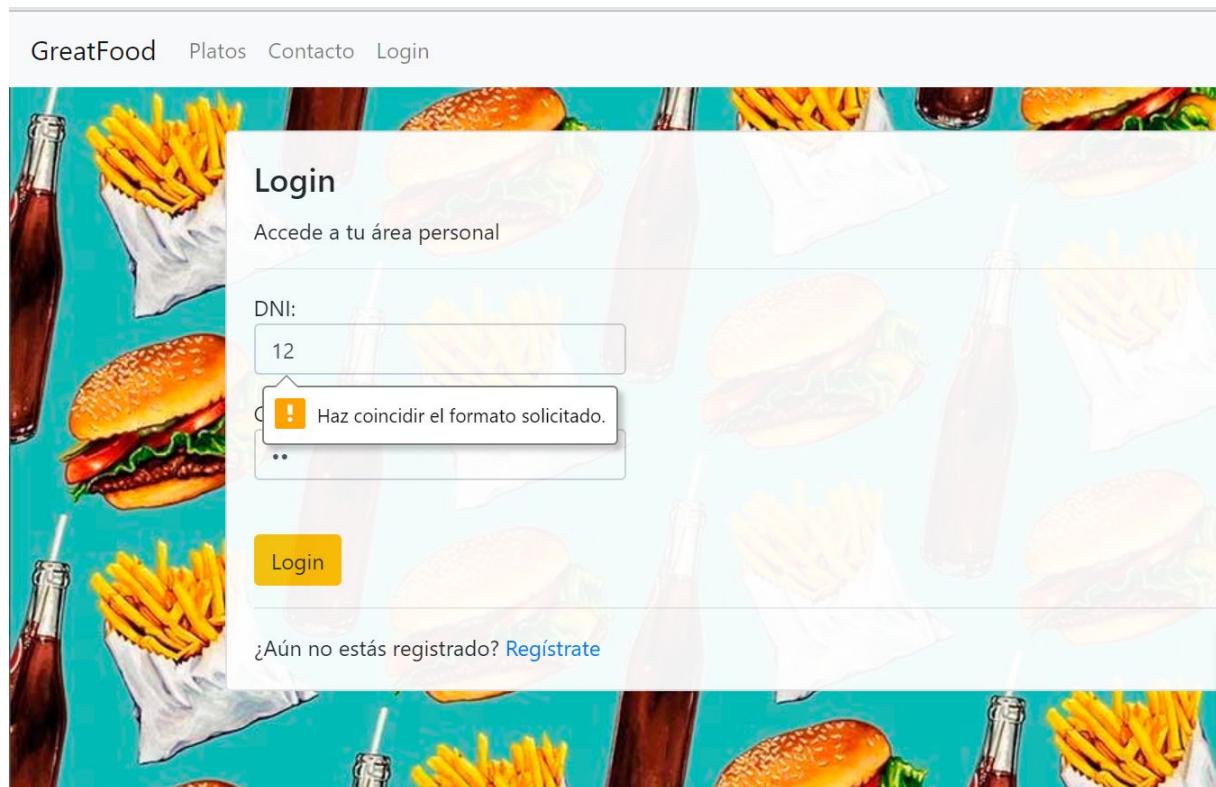
Fecha de Nacimiento:

Correo Electrónico:

! Haz coincidir el formato solicitado.

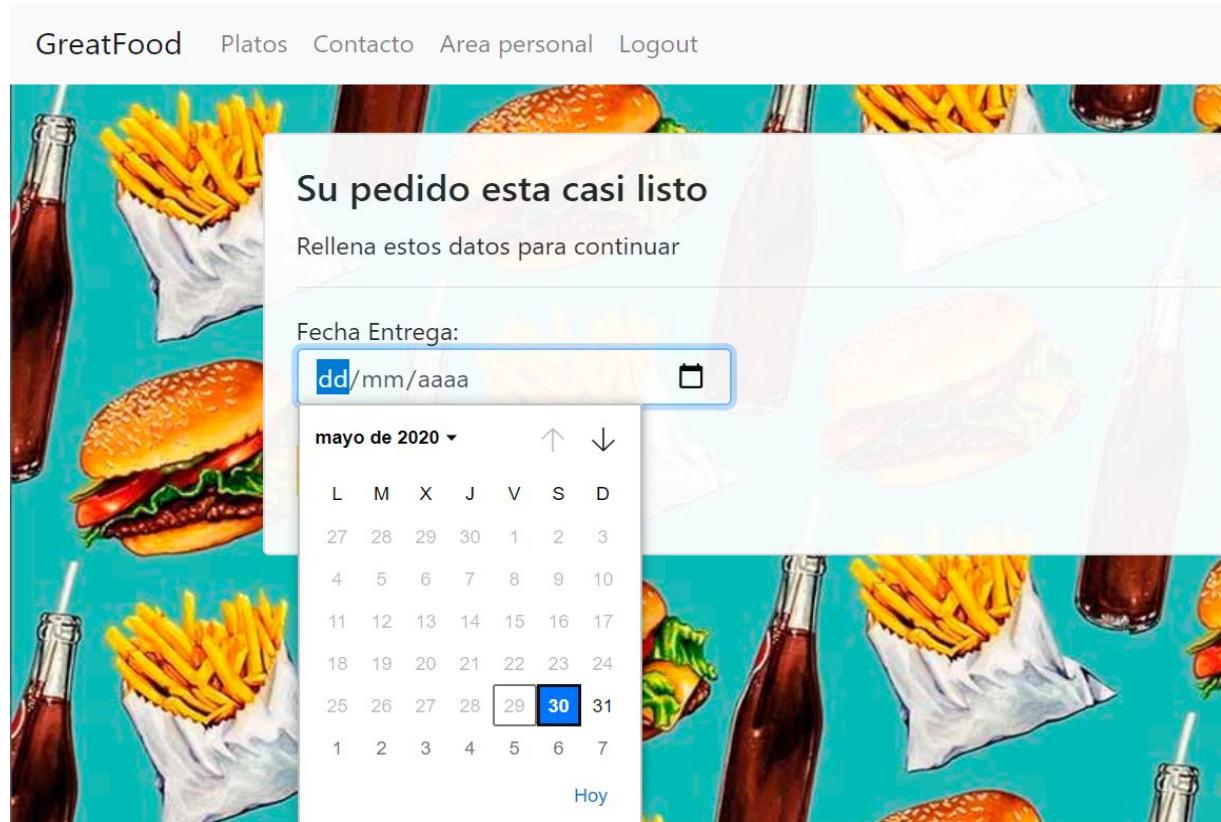
```
<p>Correo Electrónico<input type="text" name="mail" class="form-control"
  pattern="
    [a-zA-Z0-9!#$%&'*+/-=?^_`{|}~-]+(?:\.[a-zA-Z0-9!#$%&'*+/-=?^_`{|}~-]+)*@(?:[a-zA-Z0-9](?:[a-zA-Z0-9]*[a-zA-Z0-9])?\.\.)+[a-zA-Z0-9](?:[a-zA-Z0-9-]*[a-zA-Z0-9])?"$"
  required /></p>
```

El login también tiene las validaciones aunque en la base de datos no debería tener un número que no fuera un DNI válida



En este caso solo validamos la longitud del DNI, ya que donde realmente importa controlar la letra es a la hora de registrarse.

La fecha de entrega debe ser una fecha posterior al día que se realiza.

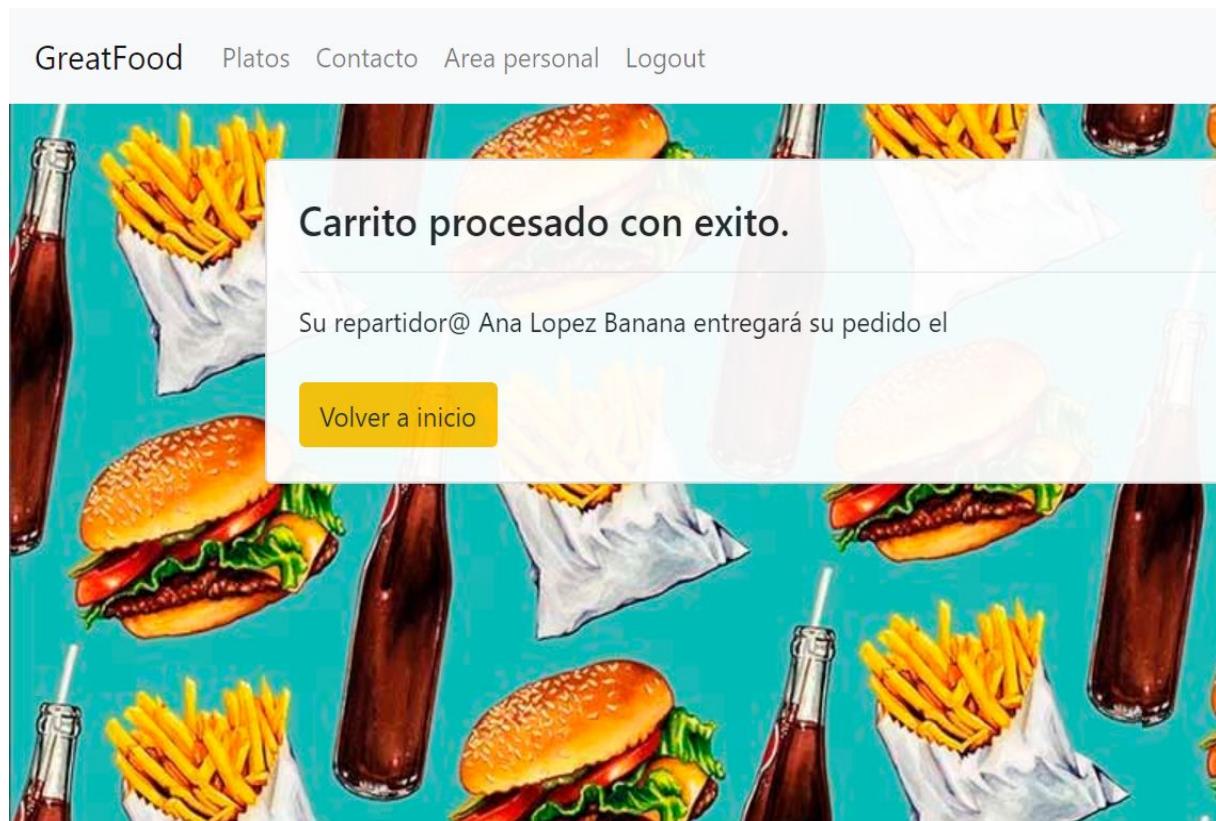


No deja elegir un dia anterior.

```
<form action="tarjeta_intermedio.php" method="POST" id="formfecha">
<hr>
<div class="form-group" id="login">
    Fecha Entrega: <input type="date" <?php
        $date_format = 'Y-m-d';
        $tomorrow = time() + (1 * 24 * 60 * 60);
        echo "min=\"".gmdate($date_format, $tomorrow).\"";
    ?> name="fecha_entrega" class="form-control" id="formfechaentrega" required><br>

    <input type="submit" value="Continuar" name="confirmar" class="btn btn-warning ">
</div>
</form>
```

## Caso positivo de Compra



Si Redsys recibe una tarjeta correcta con todos los datos correctos se procesa el pedido y llama a esta vista que muestra el repartidor encargado.

Dicha vista recibe ese resultado desde su correspondiente modelo y controlador.

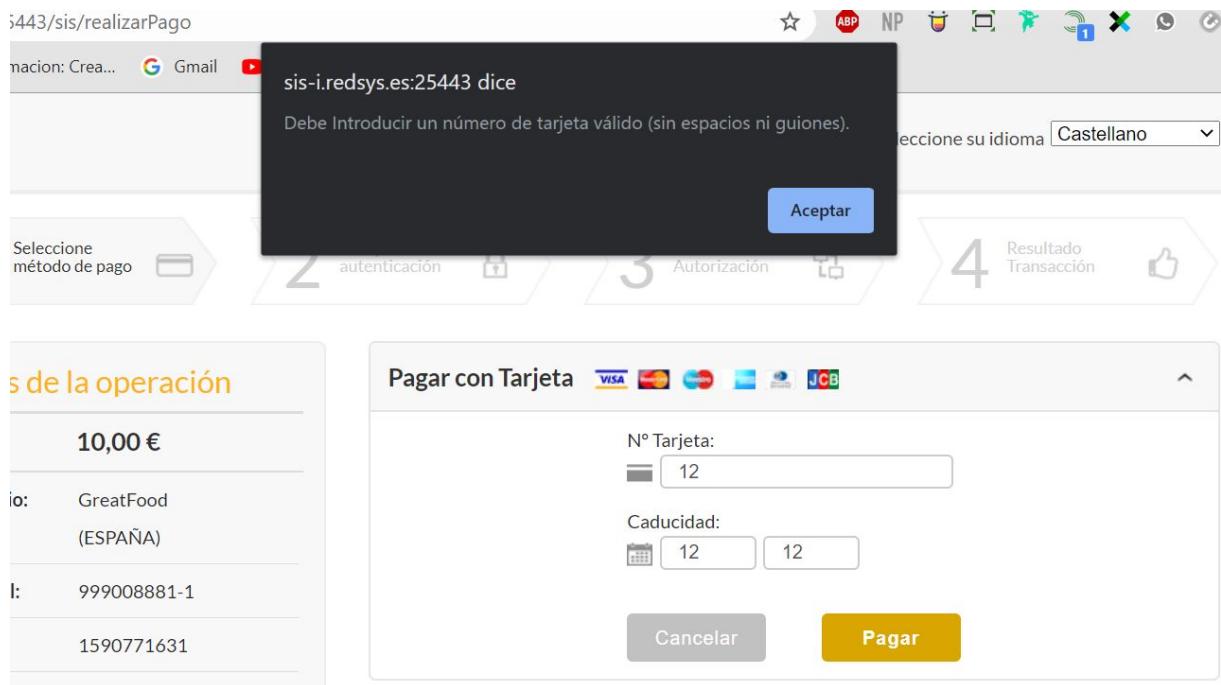
## Caso Negativo de Compra



En este caso llevamos a un controlador diferente al caso anterior para que no se procese el pedido y muestre que no se ha podido procesar.

Esta pantalla salta cuando introducimos una tarjeta incorrecta en Redsys.

La página externa para realizar el pago tiene sus propias validaciones y condiciones



Al estar en internet no tenemos control en lo que se muestra ahí, dado que nosotros solo le pasamos parámetros de entrada.

En este caso le pasamos parámetros que sirven para que los desarrolladores prueben el funcionamiento. En un caso real deberíamos poner los datos reales de nuestra empresa y nuestro banco.

## 8. Recursos y bibliografía utilizada.

Las fuentes que han sido consultadas para el desarrollo del proyecto son:

<https://www.php.net/>

<https://www.w3schools.com/>

<https://es.stackoverflow.com/>

<https://jqueryui.com/>

<https://getbootstrap.com/>

<https://developer.mozilla.org/es/>

<https://code.jquery.com/>

<https://developer.mozilla.org/es/docs/Web/JavaScript>

Además, también nos han servido de ayuda los apuntes proporcionados en clase por nuestros profesores.