

## Contenido

Datos del Alumno .....	2
Objetivo del Proyecto.....	3
Herramientas y Tecnologías usadas .....	4
Modelo de Red, Equipos y Sistemas Operativos.....	4
IDE y Herramientas de Desarrollo utilizadas .....	5
Lenguajes de Programación y de Marcas utilizados .....	5
SGBD o sistema de almacenamiento utilizado.....	5
Repositorio de Software.....	5
Framework y otras herramientas.....	5
Análisis y diseño del supuesto práctico.....	6
Modelo de clases.....	7
Modelo de casos de uso.....	7
Árbol de navegación: Página pública .....	8
Árbol de navegación: Interfaz de Administración.....	8
Modelo físico de datos .....	8
Presupuesto y financiación .....	9
Principales problemas encontrados.....	10
Bibliografía y enlaces utilizados .....	11

## Datos del Alumno

Jorge Fernández Cases

- Técnico Informático
- Programador de Aplicaciones Multiplataforma Novel
- Electromecánico Novel
- Peón de almacén de recambios de Renault.

## Objetivo del Proyecto

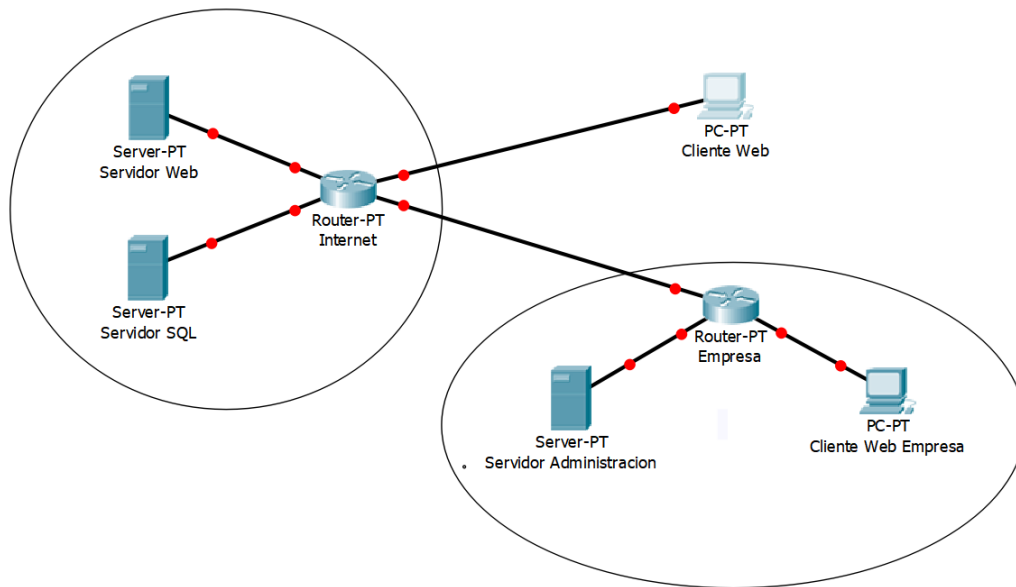
Creación de una página web para un restaurante de una localidad. Se pretende que sea un escaparate para que el establecimiento pueda mostrarse abiertamente. Estará dividido en dos partes diferenciadas:

- **Página pública:** Es la página web a la cual se tendrá acceso a través de cualquier equipo conectado a internet. Dispondrá de las siguientes secciones:
  - **Galería fotográfica:** Una selección de fotos tomadas al establecimiento y a sus platos, para tener un elemento visual atrayente.
  - **Información de contacto:** Datos de contacto del restaurante y dirección física, así como un mapa de Google para tener una mejor referencia acerca de cómo llegar.
  - **Carta:** Platos disponibles en el restaurante para ser pedidos, adjuntando información visual sobre alérgenos contenidos. Contiene un buscador de platos por su nombre. Inicialmente sólo se muestran los platos correspondientes a la temporada actual de platos. Existe la posibilidad de ver los platos sin el filtro de temporada.
- **Interfaz de administración:** Esta página web sólo es accesible desde los equipos administrativos del restaurante, y se encuentra en un equipo instalado en el restaurante. Permite las siguientes operaciones:
  - **Crear plato:** Introduce un plato de nueva creación del equipo de cocina.
  - **Modificar plato:** Cambia los datos de un plato que se haya modificado para resultar más atractivo.
  - **Borrar plato:** Elimina de la carta un plato que no se quiere volver a tener disponible por su baja aceptación.
  - **Mostrar plato:** Devuelve a la carta un plato que se haya retirado temporalmente por diversos motivos.
  - **Ocultar Plato:** Elimina temporalmente de la carta un plato por diversos motivos.
  - **Cambio de temporada:** Cambia el filtro inicial de platos a mostrar en la página web pública.

## Herramientas y Tecnologías usadas

### Modelo de Red, Equipos y Sistemas Operativos

La disposición ideal de equipos como servidores y clientes es la siguiente.



- **Servidor SQL:** Almacena las bases de datos.
  - S.O. Linux
  - Preferiblemente MySQL
- **Servidor Web:** Almacena la página web visible por todo el mundo.
  - S.O. Linux
  - Preferiblemente Apache, funcionando con HTML y PHP
  - Se conecta con el Servidor SQL para mostrar la carta
- **Cliente Web:** Usuarios de la web, que acceden a ella a través de Internet.
  - Cualquier S.O.
  - Cualquier navegador web visual que soporte HTML5
- **Servidor Administración:** Se utiliza para el mantenimiento de la base de datos de platos.
  - Independencia del S.O., preferiblemente Linux
  - Preferiblemente Apache, funcionando con HTML y PHP
  - Sólo accesible desde la red de la empresa
  - Se conecta con el Servidor SQL para permitir la administración de la carta
- **Cliente Web Empresa:** Usuarios con acceso a la web en Internet, y a la web de administración en la empresa
  - Cualquier S.O., preferiblemente para equipos de escritorio (Windows/Linux/Mac)
  - Cualquier navegador web visual que soporte HTML5

## IDE y Herramientas de Desarrollo utilizadas

[Notepad++](#): Editor de texto, para codificación.

[PuTTY](#): Control remoto modo texto, para la administración de los sistemas servidores.

[FileZilla](#): Gestor de conexiones FTP, para el intercambio de archivos entre los entornos de desarrollo y explotación

[Firefox](#), [Opera](#), [Chrome](#), [Edge](#): Navegadores web, para las comprobaciones de correcto funcionamiento.

## Lenguajes de Programación y de Marcas utilizados

[PHP](#) versión 7: Lenguaje de creación de páginas web pensado para tener una gran interacción con el usuario (web dinámicas). Tiene una elevada funcionalidad, ampliable mediante complementos. Cuando se solicita una página web a un servidor PHP, éste genera y envía una página HTML.

[HTML](#) versión 5: HTML es el lenguaje de páginas web por excelencia. Con el tiempo ha ampliado y mejorado sus funciones permitiendo la creación de sitios cada vez más vistosos y útiles. Como punto negativo, las páginas web nunca cambian (son estáticas).

[CSS](#) 3: Lenguaje diseñado para dar vistosidad a las páginas web HTML.

[JavaScript](#): Lenguaje utilizado para añadir operaciones y acciones a las páginas web. Pueden ir desde simples sumas hasta cambios completos de diseño, estilo y contenido. Estas operaciones se procesan en el navegador del sistema cliente.

## SGBD o sistema de almacenamiento utilizado

[MySQL](#) 5.6: Gestor de bases de datos con tablas relacionadas. Es de los sistemas más utilizados en aplicaciones web por sus velocidades de lectura de datos.

## Repositorio de Software

El código del proyecto y toda la documentación se encuentra [disponible en GitHub](#).

## Framework y otras herramientas

No se ha utilizado ninguna otra herramienta adicional.

## Análisis y diseño del supuesto práctico

Se ha pensado en un establecimiento pequeño y poco conocido para la creación de este sitio. Por ello, inicialmente no se dispone de una sección para eventos o para premios. Si se prospera y se gana popularidad que permitan celebrar dichos eventos, se puede agregar esa sección sin demasiado trabajo.

Para que el sitio no sea tan simple, dispone de un carrusel de imágenes con fotografías del lugar y algunos de sus platos. Pretende ser un elemento atrayente para potenciales clientes.

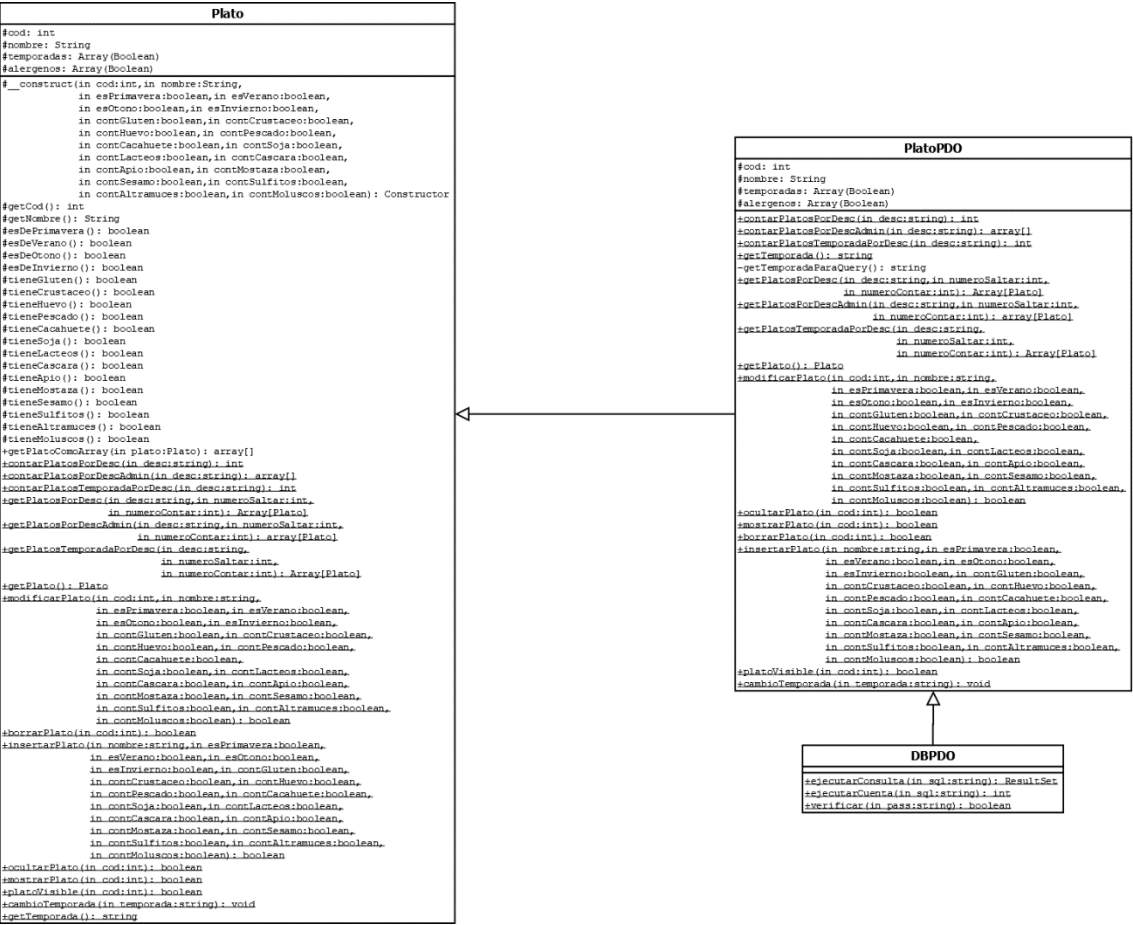
De nada sirve que los clientes quieran venir si no saben dónde está el restaurante, por lo que se incluye también la obligatoria sección “Nosotros”. En otros sitios tiene otros nombres, pero en todos cumple la misma función: Informar del emplazamiento del establecimiento y proporcionar datos de contacto.

La carta del restaurante está disponible en la web para consulta de los clientes. Esta carta mostrará los distintos alérgenos que contienen dichos platos (cumpliendo así con la normativa vigente). De igual manera, estos platos se filtrarán dependiendo de la estación que se seleccione para la carta del restaurante (por ejemplo, gazpacho en verano y sopas calientes en invierno).

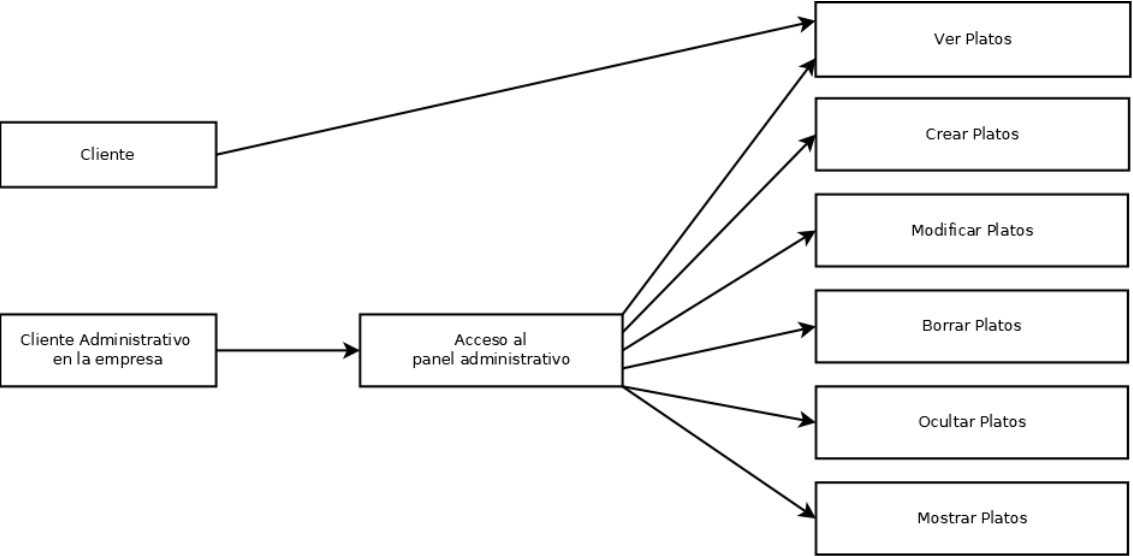
La simplicidad con la que se ha pensado en el sistema no está reñida con la seguridad. Para evitar que cualquier persona ajena al restaurante modifique la carta, la sección para este fin se ha separado, convertido en un sitio independiente y se ha instalado en un equipo del establecimiento. Sólo desde un equipo en el establecimiento puede accederse, y aun así es necesario introducir una contraseña para conseguir entrar.

Para el desarrollo de todo el entorno se ha utilizado software abierto buscando reducción de costes. Los posibles problemas y dudas pueden resolverse gracias a la gran comunidad que se ha creado alrededor de estas herramientas.

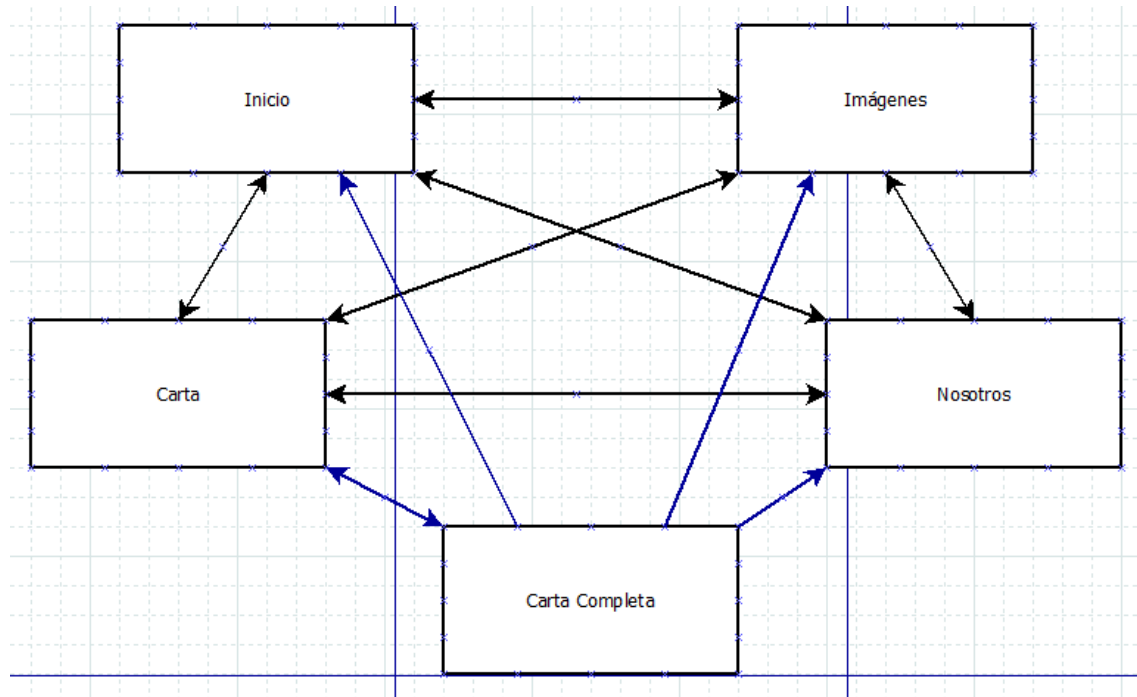
Modelo de clases



Modelo de casos de uso

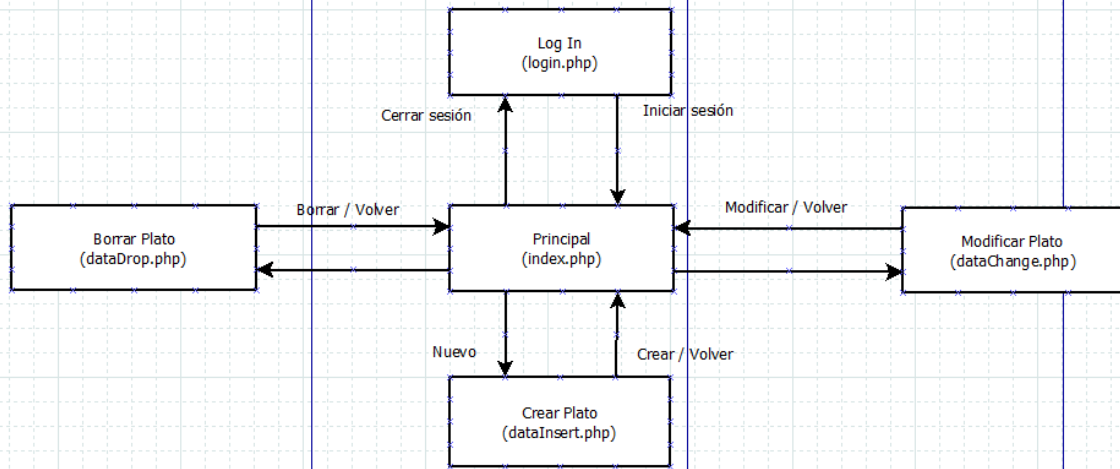


## Árbol de navegación: Página pública

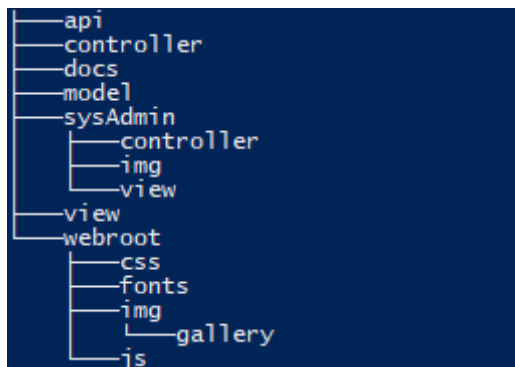


## Árbol de navegación: Interfaz de Administración

### Administrador de Platos



## Modelo físico de datos





## Presupuesto y financiación

Se calcula que el resultado visible tiene un coste de 800€ en mano de obra.

El coste del hosting externo se calcula en 14€ anuales.

El servicio de administración puede instalarse en un equipo controlado ya existente en la empresa. En el caso de querer un equipo dedicado, puede optarse por una Raspberry Pi debido al poco trabajo que soportará el mismo. La Raspberry completa costaría 60€, sin monitor puesto que se controlaría de forma remota.

## Principales problemas encontrados

**Entendimiento:** de poco sirve tener claro qué se quiere hacer y cómo si no se es capaz de expresarlo adecuadamente para que los demás también lo sepan.

**Bucle FOR en JavaScript:** El funcionamiento habitual de un proceso cíclico es realizar secuencialmente una lista de tareas. Esto implica que no se empieza una hasta que acaba la anterior. JavaScript lanza las tareas una detrás de otra sin esperar a la finalización de la anterior, lo que puede provocar problemas sobre todo con tareas que dependen de comunicaciones de red.

**Administración de tiempo:** Una mala organización provoca la dedicación de tiempo excesivo a tareas que no lo demandan, dejando tareas vitales demasiado ajustadas.

**Conectividad:** Aunque la prueba se ha desarrollado por completo sobre un solo equipo, el entorno real implica varias máquinas que deben trabajar en conjunto. Si no se realiza bien la configuración de todas ellas, la vista y mantenimiento de la carta no funcionarán.

## Bibliografía y enlaces utilizados

- Buscador de Imágenes de [Google](#)
- [Dia](#): Editor de diagramas de diversos tipos
- [Packet Tracer](#): Esquema de equipos
- Apuntes de DAW2 2016/2017