

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería Informática**

# **TRABAJO FIN DE GRADO**

**Desarrollo de una aplicación web para la gestión de comandas  
en restaurantes**

**Autor: Daniel Ramos Trejo  
Tutora: Ana González Marcos**

**JUNIO 2017**



# **Desarrollo de una aplicación web para la gestión de comandas en restaurantes**

**AUTOR: Daniel Ramos Trejo**  
**TUTORA: Ana González Marcos**

**Dpto. Ingeniería Informática**  
**Escuela Politécnica Superior**  
**Universidad Autónoma de Madrid**  
**Junio de 2017**



# Resumen

El objetivo principal de este proyecto es el diseño e implementación de una aplicación web, que denominaremos *Note The Commands*. Se trata de una aplicación web que permite gestionar y administrar las comandas de un restaurante de manera sencilla y ágil. Debido a la poca experiencia en el sector de la hostelería, así como en su gestión y administración, el proyecto se ha desarrollado atendiendo las necesidades de un cliente en concreto. Además de las peticiones del cliente hemos realizado un estudio funcional de las diferentes aplicaciones existentes para nutrirnos de más información para desarrollar una buena y completa aplicación web.

Al comienzo del proyecto se han realizado un par de reuniones con el cliente para sobre todo para la captura de los requisitos tanto funcionales como no funcionales, en cuanto al diseño de la aplicación se concretó con el cliente que fuese libre asegurándonos de que la interfaz sea sencilla e intuitiva para facilitar su uso a todas las personas incluso aquellas que no disponen de grandes conocimientos en navegación web.

La herramienta que hemos desarrollado esta implementada en el lenguaje de programación Ruby empleando el framework Rails. Se trata de una de las herramientas más utilizadas en el desarrollo web en estos momentos y uno de los más sencillos de aprender. En cuanto a la base de datos hemos utilizado MySQL. Finalmente, una vez implementado toda la funcionalidad hemos realizado un proceso de QA, esto es, hemos asegurado la calidad del software mediante un conjunto de pruebas unitarias y de integración para verificar y validar que los resultados obtenidos son los esperados.

## Palabras clave

Note The Commands, aplicación web, Ruby on Rails, framework, QA, MySQL, lenguaje de programación y base de datos.



# Abstract

The main objective of this project is the design and implementation of a web application called *Note The Commands*. It is a web application to guarantee the management and administration of restaurant's commands in a simple and effective way. Due to the lack of experience of the author in the hospitality industry, as well as in its management and administration, the project has been developed taking care of the necessity of a specific client. In addition to the client's requests we have made a functional study of the different existing applications for more information to develop a good and comprehensive web application.

At the beginning of the project a couple of meetings were carried out with the client. The purpose of these meetings was to capture both functional and non functional requirements, as well as the design of the application that was concentered with the client. A mayor goal was fixed: the application should assure that the interface will be simple and intuitive to facilitate its use to people with not a great knowledge in web browsing.

Ruby on Rails (RoR) was the chosen framework to develop the web application. Nowadays, RoR is the mostly used tool in web development and one of the simplest to learn. Additionally, a cost-effective database solution was implemented using MySQL. Finally, the application was validated with a throughout Software Quality Assurance (QA) process. Therefore, sets of unit tests and integration were implemented to verify and validate that the results obtained should be the expected.

# Keywords

Note The Commands, web application, Ruby on Rails, framework, QA, MySQL, database and programming language



## *Agradecimientos*

Primero quiero agradecer a toda mi familia por todo el apoyo que me han transmitido desde el primer día que comencé mi andadura en la universidad porque sin ellos no hubiese sido posible. Por supuesto a mis padres que siempre han estado tanto en los buenos como en los malos momentos y también por todos esos consejos sabios que me han dado. Muchas gracias.

Agradecer a mi tutora Ana González por todo este tiempo de tutorías viendo la progresión del proyecto desde el inicio hasta el final, esos consejos para mejorar la aplicación. Gracias Ana, un placer.

Muchas gracias a la pequeña familia de ingenieros informáticos que hemos hecho en la universidad todos estos años, en especial al grupo de cinco que desde el minuto uno en esta andadura nos hemos apoyado y nos seguiremos apoyando por mucho tiempo, porque con trabajo y esfuerzo todo se consigue. Gracias amigos.

También quiero agradecer por todo su apoyo a mis amigos del día a día que siempre me apoyan, me escuchan y me dan ese empujón que en ocasiones me hace falta para tirar para adelante. Gracias chicas y chicos, sé que siempre estáis.

Por último, quiero mencionar a los culpables de haber llegado hasta aquí, porque han sido años difíciles, de caer y volver a levantarse, en ocasiones parecía que nada tenía sentido. Gracias a mi abuelo que sé que estaría muy orgulloso de lo que he conseguido, porque sé que siempre está ahí. También a mi madrina, mi abuela y mi luz hasta hace pocos días, siempre preocupada y orgullosa de su *Cuquin*. Os quiero mucho.



# INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Marco del proyecto.....	1
1.2	Motivación.....	1
1.3	Objetivos.....	1
1.4	Organización de la memoria.....	2
2	Estado del arte .....	3
2.1	Introducción.....	3
2.2	Sistemas existentes .....	3
2.2.1	Conclusiones sobre el estudio de las aplicaciones existentes.....	5
2.3	Estudio de tecnologías para desarrollo web .....	5
2.3.1	Tecnologías existentes .....	6
2.3.2	Tecnologías empleadas en nuestro proyecto .....	7
2.3.3	Conclusiones sobre las tecnologías empleadas en el proyecto.....	9
3	Diseño.....	11
3.1	Metodología.....	11
3.2	Bases de datos.....	12
3.2.1	Diagrama Entidad-Relación .....	12
3.3	Diagramas de casos de uso .....	15
3.4	Diagrama de secuencia .....	16
4	Desarrollo .....	17
4.1	Introducción.....	17
4.2	Conocimientos previos acerca de Ruby on Rails .....	17
4.3	Patrón Modelo-Vista-Controlador .....	18
4.4	Modelo.....	19
4.4.1	Modelo de Usuarios.....	20
4.5	Vista .....	21
4.5.1	Vista de localización de la mesa.....	21
4.6	Controlador.....	22
4.6.1	Controlador de incidencias .....	22
5	Integración, pruebas y resultados .....	25
5.1	Pruebas .....	25
5.1.1	Desarrollo del plan de pruebas .....	25
5.1.2	Prueba para generar una comanda .....	27
5.2	Resultados.....	33
6	Conclusiones y trabajo futuro.....	35
6.1	Conclusiones.....	35
6.2	Trabajo futuro .....	35
	Referencias .....	37
	Glosario .....	39
	Anexos.....	I
A	Acta de reunión con el cliente .....	I
B	Captura de requisitos .....	III
C	Manual de instalación del entorno.....	VI
D	Estructura del proyecto.....	VII
E	Costes de la aplicación .....	VIII
F	Manual de Usuario de NoteTheCommands.....	- 1 -



# INDICE DE FIGURAS

FIGURA 2-1: LOGO HOSTELTÁCTIL .....	3
FIGURA 2-2: LOGO DUAL LINK BAR & RESTAURANT .....	4
FIGURA 2-3: LOGO RUBY ON RAILS .....	8
FIGURA 2-4: LOGO MYSQL .....	8
FIGURA 2-5: COMMITS REALIZADOS POR EL PROGRAMADOR .....	9
FIGURA 3-1: ESQUEMA DEL CICLO DE VIDA CON REALIMENTACIÓN .....	11
FIGURA 3-2: DIAGRAMA DE ENTIDAD RELACIÓN .....	14
FIGURA 3-3: DIAGRAMA DE CASOS DE USO.....	15
FIGURA 3-4: DIAGRAMA DE SECUENCIA .....	16
FIGURA 4-1: ESQUEMA PATRÓN MODELO-VISTA-CONTROLADOR.....	18
FIGURA 4-2: EJEMPLO DEL CÓDIGO PARA CREAR LA BASE DE DATOS DE LA TABLA PRODUCTS .....	19
FIGURA 4-3: CÓDIGO PARA CREAR LA BASE DE DATOS DE LA TABLA JOIN COMMANDPRODUCTS .	20
FIGURA 4-4: MODELO DE USUARIOS.....	20
FIGURA 4-5: USO DE YIELD EN NUESTRO PROYECTO .....	21
FIGURA 4-6: LOCALIZACIÓN DE LA COMANDA .....	22
FIGURA 4-7: CONTROLADOR DE LAS INCIDENCIAS 1 .....	23
FIGURA 4-8: CONTROLADOR DE LAS INCIDENCIAS 2 .....	23
FIGURA 5-1: REGISTRO DE UNA MENOR DE EDAD.....	28
FIGURA 5-2: APUNTARSE A UN RESTAURANTE .....	28
FIGURA 5-3: SELECCIONAR LA MESA PARA LA COMANDA .....	29
FIGURA 5-4: ALERTA DE PROHIBICIÓN PARA LA VENTA A MENORES.....	29
FIGURA 5-5: ALERTA AL CERRAR SESIÓN .....	30
FIGURA 5-6: COMANDA DEL RESTAURANTE .....	30
FIGURA 5-7: COCINERO COCINA LOS PRODUCTOS DE COMIDA .....	31
FIGURA 5-8: DESGLOSE DE LA COMANDA PARA EL CAMARERO .....	31

FIGURA 5-9: CIERRE DE LA COMANDA POR EL CLIENTE.....	31
FIGURA 5-10: RELLENAR CUESTIONARIO POR PARTE DEL CLIENTE.....	32
FIGURA 5-11: IMPRESIÓN DEL TICKET PARA EL CLIENTE DE LA COMANDA .....	32
FIGURA 5-12: TICKET DE LA COMANDA .....	33
FIGURA 0-1: TAREAS ASIGNADAS Y DURACIÓN .....	VIII
FIGURA 0-2: DIAGRAMA DE GANTT DE LAS TAREAS .....	VIII
FIGURA 0-3: LOGO NOTETHECOMMANDS .....	- 1 -
FIGURA 0-4: VENTANA DE INICIO BIENVENIDA.....	- 1 -
FIGURA 0-5: VENTANA PARA DAR DE ALTA A UN USUARIO EN EL SISTEMA .....	- 2 -
FIGURA 0-6: ACCESO USUARIO REGISTRADO ‘LOGIN’ .....	- 2 -
FIGURA 0-7: VENTANA ¿HA OLVIDADO LA CONTRASEÑA? .....	- 3 -
FIGURA 0-8: VENTANA DE EJEMPLO DE AYUDA.....	- 3 -
FIGURA 0-9: VENTANA DE PERFIL DE USUARIO .....	- 4 -
FIGURA 0-10: VENTANA DE INICIO PARA EL ADMINISTRADOR .....	- 4 -
FIGURA 0-11: BARRA DE TAREAS DEL ADMINISTRADOR .....	- 5 -
FIGURA 0-12: TAREAS DEL ADMINISTRADOR.....	- 5 -
FIGURA 0-13: VENTANA PARA DAR DE ALTA UN NUEVO RESTAURANTE.....	- 6 -
FIGURA 0-14: VENTANA PARA DAR DE ALTA UN NUEVO EMPLEADO.....	- 6 -
FIGURA 0-15: VENTANA PARA VISUALIZAR LOS RESTAURANTES DEL SISTEMA.....	- 7 -
FIGURA 0-16: VENTANA PARA VISUALIZAR LOS CLIENTES DEL SISTEMA.....	- 7 -
FIGURA 0-17: VENTANA PARA VISUALIZAR LOS PRODUCTOS DEL SISTEMA.....	- 8 -
FIGURA 0-18: VENTANA DE INICIO DEL CLIENTE .....	- 9 -
FIGURA 0-19: VENTANA DE SELECCIÓN DE LA MESA.....	- 9 -
FIGURA 0-20: BARRA DE TAREAS DEL CLIENTE .....	- 10 -
FIGURA 0-21: VENTANA PARA AÑADIR PRODUCTOS O MENÚS A NUESTRA COMANDA.....	- 10 -
FIGURA 0-22: VENTANA PARA PEDIR LA CUENTA DE LA COMANDA.....	- 11 -

FIGURA 0-23: VENTANA DE TAREAS DEL CLIENTE.....	- 11 -
FIGURA 0-24: VENTANA DE LAS COMANDAS DEL CLIENTE.....	- 12 -
FIGURA 0-25: VENTANA DE LOCALIZACIÓN DEL RESTAURANTE .....	- 12 -
FIGURA 0-26: VENTANA PARA MANDAR EL CURRICULUM.....	- 13 -
FIGURA 0-27: VENTANA DE LOS BLOGS DEL RESTAURANTE.....	- 13 -
FIGURA 0-28: VENTANA PARA REALIZAR UNA RESERVA .....	- 14 -
FIGURA 0-29: VENTANA DE PEDIDOS ONLINE .....	- 14 -
FIGURA 0-30: VENTANA DE LOS MENÚS DEL RESTAURANTE .....	- 15 -
FIGURA 0-31: VENTANA CON EL DESGLOSE DEL MENÚ SELECCIONADO .....	- 15 -
FIGURA 0-32: VENTANA DE CREACIÓN DE UNA INCIDENCIA EN EL RESTAURANTE .....	- 16 -
FIGURA 0-33: VENTANA DE ESTADÍSTICAS DEL RESTAURANTE .....	- 16 -
FIGURA 0-34: VENTANA DE ESTADÍSTICAS DE TODAS LAS CATEGORÍAS.....	- 17 -
FIGURA 0-35: VENTANA DEL CHAT DEL USUARIO EN EL SISTEMA.....	- 17 -
FIGURA 0-36: VENTANA DE UNA CONVERSACIÓN DEL USUARIO .....	- 18 -
FIGURA 0-37: VENTANA PARA REDACTAR UN NUEVO CORREO .....	- 18 -
FIGURA 0-38: VENTANA DE LOS CORREOS ENVIADOS .....	- 19 -
FIGURA 0-39: VENTANA CON CUESTIONARIOS DEL RESTAURANTE.....	- 19 -
FIGURA 0-40: VENTANA DE INICIO DEL GERENTE .....	- 20 -
FIGURA 0-41: BARRA DE TAREAS DEL GERENTE .....	- 21 -
FIGURA 0-42: VENTANA DE TAREAS DEL GERENTE .....	- 21 -
FIGURA 0-43: VENTANA DE COMANDAS DEL RESTAURANTE.....	- 22 -
FIGURA 0-44: VENTANA DE PEDIDOS ONLINE DEL RESTAURANTE PARA EL GERENTE.....	- 22 -
FIGURA 0-45: VENTANA DE DETALLES DEL PEDIDO ONLINE PARA EL GERENTE.....	- 23 -
FIGURA 0-46: VENTANA DE ENVÍO DE CURRICULUM PARA EL GERENTE .....	- 23 -
FIGURA 0-47: VENTANA CURRICULA PARA EL GERENTE .....	- 24 -
FIGURA 0-48: VENTANA DE NOTICIAS PARA EL GERENTE DEL RESTAURANTE .....	- 24 -

FIGURA 0-49: VENTANA DE PRODUCTOS DEL RESTAURANTE .....	- 25 -
FIGURA 0-50: VENTANA PARA CREAR UN NUEVO PRODUCTO EN EL RESTAURANTE .....	- 25 -
FIGURA 0-51: VENTANA DEL STOCK DEL RESTAURANTE.....	- 26 -
FIGURA 0-52: VENTANA PARA REGISTRAR UN PRODUCTO EN EL STOCK DE UN RESTAURANTE..	- 26 -
FIGURA 0-53: VENTANA DE MENÚS YA CREADOS PARA EL RESTAURANTE .....	- 27 -
FIGURA 0-54: VENTANA PARA AGREGAR PRODUCTOS AL MENÚ SELECCIONADO .....	- 27 -
FIGURA 0-55: VENTANA PARA CREAR UN NUEVO MENÚ PARA EL RESTAURANTE.....	- 28 -
FIGURA 0-56: VENTANA DE CUESTIONARIOS CREADOS POR EL GERENTE .....	- 28 -
FIGURA 0-57: VENTANA DE ESTADÍSTICAS DE LAS QUE DISPONE EL GERENTE .....	- 29 -
FIGURA 0-58: VENTANA DE ESTADÍSTICAS DE LAS VOTACIONES QUE DISPONE EL GERENTE.....	- 29 -
FIGURA 0-59: VENTANA DE INCIDENCIAS PARA EL GERENTE DEL RESTAURANTE .....	- 30 -
FIGURA 0-60: VENTANA DE RESERVAS DEL RESTAURANTE.....	- 30 -
FIGURA 0-61: VENTANA DE RESERVAS DE HOY PARA EL RESTAURANTE .....	- 31 -
FIGURA 0-62: VENTANA DE INICIO DEL CAMARERO .....	- 32 -
FIGURA 0-63: BARRA DE TAREAS DEL CAMARERO .....	- 32 -
FIGURA 0-64: VENTANA DE TAREAS DEL CAMARERO.....	- 33 -
FIGURA 0-65: VENTANA DE COMANDAS PARA EL CAMARERO.....	- 34 -
FIGURA 0-66: VENTANA CON EL DESGLOSE DE LA COMANDA PARA EL CAMARERO .....	- 34 -
FIGURA 0-67: PRODUCTOS SERVIDOS POR EL CAMARERO .....	- 35 -
FIGURA 0-68: VENTANA RESERVAS DEL RESTAURANTE PARA EL CAMARERO .....	- 36 -
FIGURA 0-69: VENTANA DE INICIO DEL COCINERO .....	- 37 -
FIGURA 0-70: BARRA DE TAREAS DEL COCINERO .....	- 38 -
FIGURA 0-71: VENTANA DE TAREAS DEL COCINERO.....	- 38 -
FIGURA 0-72: VENTANA DE PRODUCTOS A COCINAR.....	- 39 -

# 1 Introducción

---

## 1.1 Marco del proyecto

Este Trabajo de Fin de Grado ha sido realizado por el estudiante Daniel Ramos Trejo. El objetivo del proyecto es construir una aplicación web de gestión de comandas de restaurantes. Para ello se ha realizado el diseño e implementación final mediante el uso de herramientas avanzadas de programación web. Se trata de una propuesta de Trabajo Fin de Grado (TFG) de carácter personal a sugerencia de un gerente de restaurante.

## 1.2 Motivación

Un mal endémico de muchos restaurantes es la tardanza en ser atendidos por el servicio dejando a los comensales en ocasiones desamparados a la espera. Hoy en día cada vez más los “tiempos muertos” deben dejar de existir.

Actualmente, existen algunas aplicaciones web que realizan esta función de ayudar al restaurante a mejorar la comunicación con sus clientes. La mayoría de las aplicaciones están centradas en la gestión interna por parte de los empleados del restaurante, dejando de lado un amplio grupo de funcionalidades, como realizar reservas o pedidos online, que el usuario puede realizar de forma personal desde una web.

Por tanto, la motivación para desarrollar este proyecto se basa en la construcción de una aplicación web que permita al cliente interactuar, sin discriminar a usuarios por sus conocimientos tecnológicos, y poder agilizar el trabajo administrativo y de servicio del restaurante.

## 1.3 Objetivos

El objetivo de este TFG es la definición, implementación y validación de un sistema informático web cuyo propósito principal es gestionar restaurantes con la interacción de los clientes.

Las principales funcionalidades que presenta la aplicación son:

- **Gestor de la aplicación:** Se trata del creador, por tanto, propietario de la aplicación encargado de dar de alta restaurantes y empleados del mismo.
- **Registros:** La aplicación permitirá registrar usuarios que posteriormente se apuntarán a los restaurantes que estén dados de alta en el sistema. El administrador de la aplicación será el encargado de dar de alta los restaurantes, así como, asignar el rol de cada uno de los usuarios empleados.
- **Gestión de comandas:** Permite al usuario registrado en un restaurante realizar una comanda desde la mesa, seleccionando previamente su ubicación, y añadiendo productos o menús a la misma. Dicha comanda la visualizará el camarero (si son

bebidas), el cocinero (si son comidas) y el gerente que podrá ver todas las comandas de su restaurante.

- **Gestión de blog:** El gerente podrá crear, modificar, eliminar y comentar una noticia en el blog, mientras que el cliente solo podrá comentar.
- **Gestión de incidencias:** Permite a cualquier usuario crear una incidencia que solo el gerente podrá visualizar, editar y borrar.
- **Gestión de productos y stock:** La aplicación permitirá solo al gerente crear, editar, visualizar y eliminar un producto de su restaurante. A su vez podrá añadir productos al stock de su restaurante.
- **Gestión de menús:** La aplicación permitirá que el gerente pueda crear, modificar, agregar y eliminar productos de un menú para su posterior visualización por parte de los clientes.
- **Gestión de cuestionarios:** La aplicación dispondrá de un apartado donde el gerente de cada restaurante pueda crear y eliminar cuestionarios que los clientes realizarán de forma anónima mientras le traen la cuenta.
- **Módulo de mensajería:** Se dispondrá de un módulo de mensajería privada en el cual los usuarios podrán visualizar su bandeja de entrada y de salida, y podrán enviar mensajes al gerente o al restaurante en concreto.
- **Módulo de chat:** La aplicación dispondrá de mensajería instantánea en la que los clientes de un restaurante podrán chatear con otros clientes que estén registrados en el mismo restaurante.
- **Gestión de pedidos y reservas online:** Permitirá al cliente realizar un pedido online y también, tanto el cliente como los empleados del restaurante, podrán realizar reservas.

## **1.4 Organización de la memoria**

La memoria consta de los siguientes capítulos:

- En el **capítulo 2** se han estudiado dos aplicaciones para coger cierta orientación de los sistemas existentes en el sector de la hostelería y poder nutrirnos de información para añadirlos a nuestra aplicación.
- En el **capítulo 3** se describe el diseño de la aplicación implementada incluyendo el diseño de la base de datos y los distintos diagramas de casos de uso y de secuencia.
- En el **capítulo 4** vemos el mecanismo para el desarrollo de la aplicación y la forma en que hemos implementado las distintas tecnologías.
- El **capítulo 5** se dedica a la batería de pruebas. Se han realizado pruebas exhaustivas de cada uno de los módulos y las pruebas de integración del proyecto.
- En el **capítulo 6** se describen las conclusiones obtenidas al finalizar el proyecto y las posibles líneas de trabajo futuro.

## 2 Estado del arte

---

### 2.1 Introducción

Actualmente existen multitud de aplicaciones web orientadas a la gestión de restaurantes y por ello antes de empezar a desarrollar se ha realizado una investigación de mercado para evaluar las características intrínsecas de los productos existentes e identificar la percepción del consumidor. Durante el estudio se observó que hay aplicaciones más orientadas a los profesionales/empleados del sector. El objetivo de este estudio era brindar de información confiable sobre estos sistemas y así poder analizar los pros y contras de los mismos.

### 2.2 Sistemas existentes

A continuación, se analizarán los sistemas existentes que más se asemejan a nuestro proyecto:

- **Hosteltáctil** (<http://www.hosteltactil.com>)



**Figura 2-1: Logo Hosteltáctil**

Se trata de un potente software de gestión creado específicamente para el sector de la hostelería, permitiendo un control total de todas y cada una de las áreas de trabajo de tu empresa, desde la gestión de salas y mesas, pasando por la gestión de recursos humanos (RR.HH), cocina, almacén, compras, proveedores, escandallos, hasta la gestión de clientes [1].

Hosteltáctil ha sido elegida para el análisis por su semejanza, en cuanto a funcionalidad, con nuestra aplicación web.

El punto de acceso a la aplicación se realiza a través de un Terminal Punto de Venta (TPV), presentando dos modos de uso: gerente y empleados (camareros y cocineros). Es el gerente del restaurante el que controla quien accede a la información que genera su sistema de gestión *Hosteltáctil* a través de un módulo donde se determina el nivel de acceso de los empleados. Con esto determina quién y qué puede ver cada empleado asegurando así la protección y la confidencialidad de los datos.

Hosteltáctil tiene un módulo llamado “telecomanda” que permite gestionar, agilizar y mejorar el servicio en sala. Este servicio será de uso exclusivo del camarero para anotar las comandas, realizar recomendaciones,

anotar comentarios a cocina sobre los productos servidos y cobrar desde cualquier punto de negocio. Dispone de una gran cantidad de dispositivos PDA, Tablets para Android y sistemas de pedidos portátiles *Orderman*.

En cuanto al gerente del restaurante se le ofrece la posibilidad de poder gestionar su negocio sin tener que estar físicamente en él, pudiendo acceder desde cualquier parte del mundo. Se trata de una gestión remota llamada “Hosteltáctil Cloud” que ofrece información sobre el funcionamiento económico y operativo (facturas diarias, cierres de cajas, ventas por horas, movimiento de caja entre otras) de tu empresa las 24 horas del día/365 días del año.

Todas estas funcionalidades se han nutrido de la experiencia y la profesionalidad de empresarios de renombre en la hostelería, reforzando así el sistema de *Hosteltáctil*. Una solución especializada que permite un control y gestión de todos los procesos de trabajo de la empresa hostelera de forma sencilla, rápida, ágil, segura y personalizada.

Como aspecto negativo a tener en cuenta es que el cliente tendrá que esperar al camarero para añadir productos a su comanda, surgiendo así lo que se conoce como “tiempos muertos” en los que el cliente espera a ser atendido.

- **Dual Link Bar & Restaurant** (<http://www.dual-link.es/>)



**Figura 2-2: Logo Dual Link Bar & Restaurant**

*Dual Link Bar and Restaurant* es una aplicación web *Dual Link Manager* para gestión de restaurantes donde puedes hacer cualquier gestión de TPV estés donde estés. Al ser una aplicación web, si tienes una conexión a internet podrás tener el control absoluto de todo lo que suceda en tu local, controlando cada aspecto de manera remota, sin tener que estar presente en el local [2].

Entre las principales funcionalidades es que presenta un funcionamiento local, es decir, no necesita disponer de un servidor externo donde almacenar los datos. Se trata de una aplicación para iPad la cual presenta varios tipos de licencia que comentaremos a continuación. Otra de las funcionalidades a destacar es que el gerente del restaurante podrá gestionar uno o tantos negocios como posea.

*Dual Link Bar and Restaurant* dispone de varias licencias con variedad de precios para cada una de las jerarquías de la restauración. Una primera licencia de TPV que realiza las funciones de servidor de datos y sincronización del resto de aplicaciones (presenta un coste de 1,5 €/ día, 547,50 € al año). Una segunda licencia de Comandero que dispone de todas las funcionalidades del

TPV Dual Link, excepto la administración, y puede adquirir tantas licencias como necesite (presenta un coste de 0,25 €/ día, 91,25 € al año por Comandero). Una tercera licencia de TPV esclavo, TPV adicional con las mismas funcionalidades que la primera licencia, pero sin la sección de administración, pudiendo adquirir tantas licencias como sea preciso (presenta un coste de 0,50 €/día, 182,50 € al año por TPV esclavo). Una cuarta y última licencia de *Chef* donde visualizará los productos que tienen que ser elaborados, sin necesidad de usar papel ni esperar a que el cocinero llegue hasta la cocina con la comanda (presenta un coste de 0,50 €/día, 182,50 € al año por *Chef*).

Tras analizar la versión de iPad sin comprar la licencia, se observa que aparentemente tiene una interfaz sencilla, aunque puede llegar a ser complicada para un usuario poco familiarizado con este tipo de aplicaciones, volviéndose poco interactiva. Como ocurre con la anterior aplicación carece de la posibilidad de interacción del cliente con su comanda.

### **2.2.1 Conclusiones sobre el estudio de las aplicaciones existentes**

Como ya hemos señalado se han analizado dos de los servicios más destacados hoy en día, ambos comparten objetivos con nuestro proyecto. Se caracterizan por estar orientadas a la gestión de comandas de un restaurante y tener una interfaz más o menos sencilla para el usuario que lo manipula, centrado en (personal profesional del restaurante).

Es importante destacar que no están orientadas a que los clientes puedan interactuar con la aplicación para realizar algunas de las funcionalidades que le permitirían agilizar el servicio del mismo.

Otro de los inconvenientes que presentan estos sistemas es que es el propio gerente del restaurante el que se encarga de decidir qué información será visible para cada usuario empleado.

Con esta recopilación de información hemos determinado mejorar los puntos que se han mencionado: (1) realizar una interfaz más sencilla de cara a cualquier usuario con bajos o medios niveles de navegación web, (2) orientar la aplicación añadiendo la interactividad con los clientes, por ejemplo, en la elaboración de su comanda y (3) distribuir las funcionalidades dependiendo de los roles de los empleados del restaurante (camarero, cocinero) así como el propio gerente con el fin de minimizar el trabajo de reparto de roles.

### **2.3 Estudio de tecnologías para desarrollo web**

En este apartado se realizará un análisis de las principales tecnologías para el desarrollo web más destacadas hoy en día y así poder seleccionar la tecnología que más se adecue a los requisitos de nuestro proyecto.

A continuación, mostraremos las tecnologías existentes para el desarrollo de aplicaciones web y posteriormente citaremos cual ha sido nuestra elección para el desarrollo de nuestro proyecto.

### 2.3.1 Tecnologías existentes

Entre las tecnologías existentes para el desarrollo web cabe destacar PHP, JSF y .NET. Aunque poco a poco van naciendo nuevas tecnologías que van ganando terreno a las tradicionales como puede ser Django entre otras.

- **Lenguaje de programación: PHP.**

Lenguaje de programación de código de lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin coste alguno [3].

Posee parecido con los lenguajes más comunes de programación estructurada, como C y Perl, permitiendo a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta, permitiéndoles a su vez involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

- **Framework de desarrollo Java: JSF.**

Tecnología y framework para aplicaciones Java basadas en web que simplifican el desarrollo de interfaces de usuario en aplicaciones Java EE. Se trata de la sucesión de JSP (JavaServer Pages), por lo tanto, está programado en lenguaje Java [4].

Al estar programado en lenguaje Java, estamos hablando de que JSF es un framework Modelo Vista Controlador (MVC) basado en el API de Servlets que proporciona un conjunto de componentes en forma de etiquetas definidas en páginas XHTML mediante el framework Facelets utilizándolas como vistas, objetos JavaBean como modelos y métodos de esos objetos como controladores. Presenta una arquitectura claramente separada entre lógica de la aplicación y presentación facilitando la conexión con el código de la aplicación.

- **Framework: .NET.**

Framework para aplicaciones web desarrollado y comercializado por Microsoft, utilizado por programadores y diseñadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Se trata de una propuesta rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones web [5].

- **Framework de desarrollo Python: Django.**

Se trata de un framework de desarrollo web de alto nivel de código abierto, escrito en Python, que respeta el patrón de diseño MVC. La meta fundamental de Django es la de facilitar la creación de sitios web complejos y estar enfocado a sistemas multiplataforma [6].

Otra ventaja de Django es que permite realizar una aplicación web de forma rápida, utilizando la filosofía DRY (*Don't Repeat Yourself*), esto es, reducción de duplicación de código a la hora de programar. Además, las consultas a la base de datos son más sencillas e intuitivas, evitando consultas infernales a las tablas de la base de datos.

Para realizar un buen proyecto con este framework es de vital importancia tener conocimiento de Python ya que puede ser algo complicado encontrar información o documentación para su desarrollo.

### 2.3.2 Tecnologías empleadas en nuestro proyecto

Es aquí donde citaremos las tecnologías que hemos decidido utilizar para la elaboración de nuestro proyecto. Las tecnologías seleccionadas se han escogido previo estudio de mercado, viendo cuáles son las tecnologías de desarrollo web más demandadas y con mayor proyección en la actualidad.

Después de este estudio se tomó la decisión de realizar el proyecto en el framework RoR, más conocido como Ruby on Rails o Rails.

- **Framework de desarrollo Ruby: Ruby on Rails.**

Framework de aplicaciones web de código abierto orientada a objetos, escrito en el lenguaje de programación Ruby, siguiendo el paradigma del patrón MVC. Ruby permite la metaprogramación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran legible [7].

Presenta un conjunto de librerías, automatismos y convenciones para construir aplicaciones web que acceden a bases de datos [8].

Combina la simplicidad con la posibilidad de desarrollar aplicaciones complejas en menos código que con otros frameworks. Al igual que comentamos en Django, uno de los puntos fuertes de RoR es el de utilizar la filosofía DRY para evitar duplicidad de código.

Rails se distribuye a través de un gestor de paquetes para el lenguaje de programación Ruby llamado RubyGems que proporcionan un formato estándar y autocontenido para la distribución de programas o bibliotecas en Ruby [9].

Encontrar información de este framework resulta bastante difícil porque hay pocos manuales y por ello requiere de conocimientos de programación para la elaboración de un buen proyecto.

Para el desarrollo de este TFG se ha utilizado este framework con la versión 5.1, utilizando para la parte de back-end el lenguaje de programación Ruby. Cabe destacar que se incrusta fácilmente en el código HTML.



Figura 2-3: Logo Ruby on Rails

- **Lenguaje de programación: HTML5.**

Se trata de la quinta versión del lenguaje básico de la *World Wide Web* HTML. Este lenguaje de etiquetas es un estándar a cargo de W3C y es la base utilizada para la creación de páginas web [10]. Se ha utilizado este lenguaje en la versión HTML5.

- **Lenguaje de hojas de estilos: CSS3.**

CSS o hoja de estilos en cascada es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado, de esta forma separamos el estilo y el contenido de las vistas [11]. Se ha utilizado para definir la presentación de los documentos de este proyecto y la versión empleada ha sido la 3.0.

- **Lenguaje de programación: JavaScript.**

Para dar un efecto dinámico a la web hemos empleado JavaScript (abreviado comúnmente JS) incrustado junto con el código HTML. Se trata de un lenguaje de programación interpretado, esto es, analiza y ejecuta en la medida en que sea necesaria y está orientada a objetos [12].

- **Desarrollo y manejo de bases de Datos: MySQL 5.7 y Workbench 6.3.**

La definición de la base de datos se ha realizado en el lenguaje de programación SQL utilizando el sistema de gestión MySQL [13]. *Ruby on Rails* tiene la peculiaridad de que al crear un proyecto de cero utiliza por defecto la base de datos SQLite3, pero se ha decidido utilizar el sistema de gestión MySQL ya que funciona muy bien para proyectos de gran envergadura de datos como puede ser el TFG que se ha desarrollado. Se ha utilizado la versión 5.7 para Linux.

Para realizar la interacción con la base de datos se ha utilizado una herramienta visual llamada *Workbench* que permite administrar y mantener una base de datos MySQL [14]. Se ha utilizado la versión 6.3.



Figura 2-4: Logo MySQL

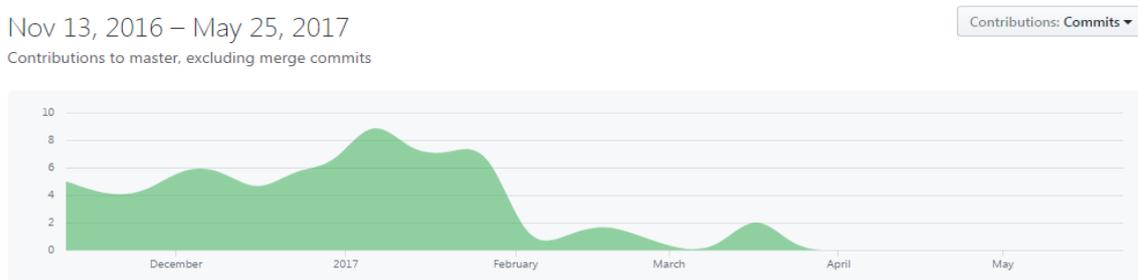
- **Plataforma de control de versiones: GitHub.**

Plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git [15]. GitHub, entre otros, es una de las herramientas más utilizadas por las grandes empresas para alojar el código de su proyecto en un repositorio privado.

Por ello he utilizado esta plataforma para alojar mi proyecto ya que es una herramienta muy útil, ágil y de fácil manipulación para el control de versiones. A continuación, se muestra un diagrama detallado con los distintos *commits* realizados a lo largo de la vida del proyecto. Los comandos utilizados para subir las versiones desde la terminal de Linux a nuestro repositorio son:

- Preparar para subir todos los archivos modificados: `'git add .'.`
- Dar nombre a la versión que vamos a subir: `'git commit -m "nombre_version"'`.
- Subir al repositorio: `'git push origin master'`.
- Comprobar que se ha subido correctamente: `'git status'`.

En la Figura 2-5 se muestra el histograma de *commits* que se han realizado a lo largo del desarrollo del proyecto. Como se puede apreciar, existe una fase importante de desarrollo que se corresponde con los primeros meses del proyecto, de Noviembre a Enero.



**Figura 2-5: Commits realizados por el programador**

### 2.3.3 Conclusiones sobre las tecnologías empleadas en el proyecto

A la vista del estudio realizado, concluimos que las tecnologías seleccionadas para realizar el proyecto son las herramientas empleadas hoy en día en el desarrollo de aplicaciones web en empresas punteras de ámbito mundial.

En cuanto al lenguaje de programación RoR, al presentar modularización hace sencilla su mantenibilidad en el futuro. El lenguaje Ruby en si es bastante intuitivo y a la vez sencillo de aprender partiendo de la buena base de programación adquirida al cursar el grado.

Por otro lado, la integración del lenguaje Ruby en los ficheros HTML es sencilla y el uso de dicho lenguaje ahorra líneas de código con respecto a otro tipo de lenguajes de programación web.

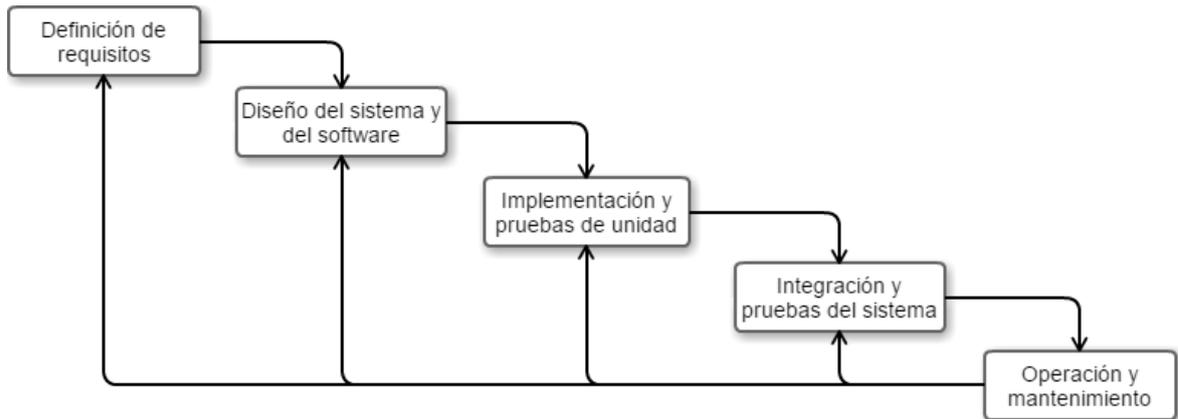


# 3 Diseño

---

## 3.1 Metodología

Para el desarrollo del proyecto se ha seguido un ciclo de vida en cascada con realimentación desde la fase inicial hasta la final, facilitando la ordenación y agrupación de las etapas del proceso de desarrollo de nuestro proyecto, permitiendo modificar aspectos que surgieron en alguna de las fases.



**Figura 3-1: Esquema del ciclo de vida con realimentación**

La Figura 3-1 esquematiza el ciclo de vida en cascada y a continuación, detallaremos las tareas realizadas en cada una de las fases del desarrollo:

- **Definición de requisitos:** Fase en la que se analizaron las necesidades del cliente [ANEXO A]. Como ya se mencionó en apartados anteriores, este proyecto es una petición real de un cliente. Se realizaron reuniones con el cliente para capturar los requisitos funcionales y especificar los hitos del proyecto [ANEXO B].
- **Diseño del sistema y del software:** Fase en la que se han esbozado los diagramas de estado, entidad-relación para la base de datos y el diagrama de secuencia del proyecto.
- **Implementación y pruebas de unidad:** En esta fase se ha llevado a cabo la codificación de las bases de datos del proyecto, los módulos y sus respectivas pruebas unitarias.
- **Integración y pruebas del sistema:** Fase de realización de la estrategia de pruebas, los casos de prueba y desarrollo de las mismas para cerciorarnos del correcto funcionamiento del proyecto, *Quality Assurance* (QA).
- **Operación y mantenimiento:** Se trata de una de las fases destinadas a arreglar errores que hayan ido surgiendo y realizar nuevas mejoras del proyecto.

## 3.2 Bases de datos

Para que los datos persistan en la vida de la aplicación se ha diseñado una base de datos relacional en MySQL que almacena todos los datos relacionados con los usuarios, comandas, restaurantes, mensajería, votaciones, stock, incidencias, menús, productos, reservas, takeaways y chat. Por ello se ha realizado un diseño robusto ya que el almacenamiento y la facilidad de extracción de datos en las consultas es una de las partes más importante del proyecto.

### 3.2.1 Diagrama Entidad-Relación

A continuación, se muestra en la Figura 3-2 el diagrama entidad-relación del sistema de almacenamiento de la aplicación, pero antes haremos un breve repaso de las entidades de nuestra base de datos:

**restaurants:** Entidad formada por los restaurantes del sistema. Cabe destacar que al registrar un nuevo restaurante se guardan automáticamente la latitud y longitud de su ubicación en el mapa para poder visualizarlo desde la propia aplicación. El número máximo de restaurantes a registrar no está prefijado.

**users:** Entidad formada por un conjunto de usuarios que pueden estar registrados en la aplicación. Los tipos de usuarios que pueden registrarse son el administrador de la aplicación, usuarios clientes, usuarios cocineros, usuarios camareros y usuarios gerente. Al registrar un usuario su contraseña será cifrada y almacenada en base de datos. El número de usuarios en el sistema no está limitado.

**restaurant\_users:** Entidad *Join* en la que se almacenan los roles de los usuarios en cada restaurante, junto con los identificadores del restaurante y del usuario. Un mismo usuario puede estar registrado en un mismo restaurante como cliente o como cualquier cargo de la jerarquía (cocinero, camarero o gerente).

**products:** Entidad para los productos de cada uno de los restaurantes registrados en el sistema. El producto puede ser de los siguientes tipos: "RACION", "CALDOS", "PESCADOS", "CARNES", "ENSALADAS", "BOCADILLO", "POSTRE", "BEBIDA", "BEBIDAS ALCOHOLICAS", "APERITIVO", "TACO", "BURRITO", "WRAP", "GENERAL", "EXTRAS". El número de productos en el restaurante no está limitado.

**stocks:** Entidad para las existencias o reservas de productos que se encuentran en cada uno de los restaurantes del sistema. Cada restaurante puede tener un único stock, formado por un número de productos no limitado.

**questionnaires:** Entidad para las cuestiones que los gerentes de cada restaurante realizan para que los clientes los rellenen y así nutrirse de información de forma anónima. Cada restaurante puede tener un número de cuestionarios no limitado.

**votings:** Entidad para las votaciones que dan los clientes a cada uno de los cuestionarios de los restaurantes del sistema. Cada cuestionario solo admite una única respuesta por cliente.

**incidences:** Entidad para las incidencias que los usuarios del sistema realizan para cualquier restaurante. La categoría de las incidencias puede ser: "COMIDA", "MOBILIARIO", "COCINA", "BAÑO", "TERRAZA", "BARRA", "SALON", "ATENCION PERSONAL", "ELECTRICIDAD", "GENERAL" y la prioridad de la incidencia puede ser: "MUY BAJA", "BAJA", "MEDIA", "ALTA", "MUY ALTA". Los restaurantes pueden tener un número de incidencias no limitado.

**emails:** Entidad para los correos de los clientes registrados en el sistema. Cada usuario puede tener un número de emails tanto de entrada como de salida no limitado.

**curriculum:** Entidad para los *curricula* de los usuarios de los restaurantes. Cada restaurante tiene un número de *curricula* no limitado.

**blogs:** Entidad en la que se almacenaran todas las noticias que los gerentes crean para su restaurante. Cada restaurante puede tener un número de noticias no limitado.

**blog\_users:** Entidad *Join* en la que se almacenan los comentarios de los usuarios sobre una noticia de un restaurante determinado, junto con los identificadores de usuario y blog. Un usuario puede realizar un número de comentarios no limitado en el blog.

**takeaway:** Entidad formada por los pedidos para llevar de cada cliente. Los restaurantes pueden tener un número de pedidos no limitado.

**products\_takeaway:** Entidad *Join* en la que se almacenan la cantidad de productos que se seleccionan para cada uno de los pedidos, junto con los identificadores del pedido y del producto. No hay límite de productos en el pedido, dejando responsable al cliente.

**conversations:** Entidad para las conversaciones abiertas entre dos usuarios en el chat del sistema. Estas conversaciones se pueden dar entre todos los clientes registrados en el sistema y el número de conversaciones no está limitado.

**personal\_messages:** Entidad para los mensajes personales entre dos usuarios del sistema. Estos mensajes personales en una misma conversación no están limitados.

**reservation:** Entidad para las reservas que los usuarios realizan en un restaurante determinado del sistema. Una misma persona no puede reservar dos veces un mismo día. El número de reservas en un restaurante no está limitado.

**menus:** Entidad para los menús de los restaurantes del sistema. El menú puede ser de los siguientes tipos: "LUNES", "MARTES", "MIERCOLES", "JUEVES", "VIERNES", "SABADO", "DOMINGO", "ESPECIAL", "INFANTIL". El número de menús en un restaurante no está limitado.

**menu\_products:** Entidad *Join* en la que se almacenan los productos de un menú con sus respectivos identificadores y la categoría del producto que puede ser "ENTRANTE", "PRIMER PLATO", "SEGUNDO PLATO", "POSTRE", "BEBIDA", siendo el número de productos del menú no limitado.

**commands:** Entidad para las comandas de los restaurantes. Los restaurantes pueden estar formadas por varias comandas, siendo el número de comandas no limitado.

**command\_menus:** Entidad *Join* que almacena los menús de la comanda con sus respectivos identificadores, siendo el número de menús en la comanda no limitado.

**comand\_products:** Entidad *Join* que almacena los productos de una comanda con sus respectivos identificadores, siendo el número de productos en la comanda no limitado.

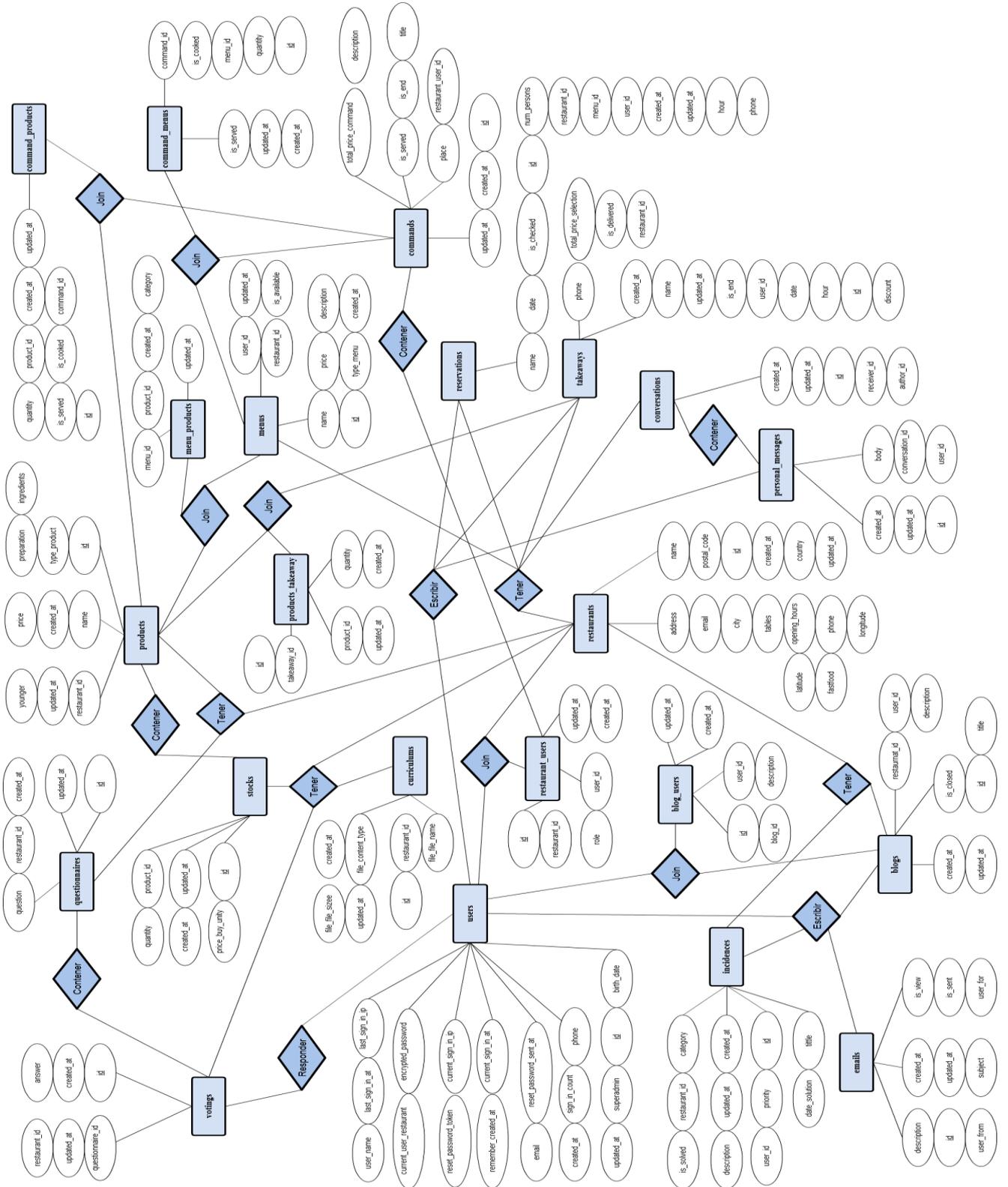


Figura 3-2: Diagrama de Entidad Relación

### 3.3 Diagramas de casos de uso

Los diagramas de casos de uso representan gráficamente el comportamiento de los usuarios en el sistema, así como los pasos detallados para realizar una actividad. A continuación, se muestra el diagrama de casos de uso del proyecto.

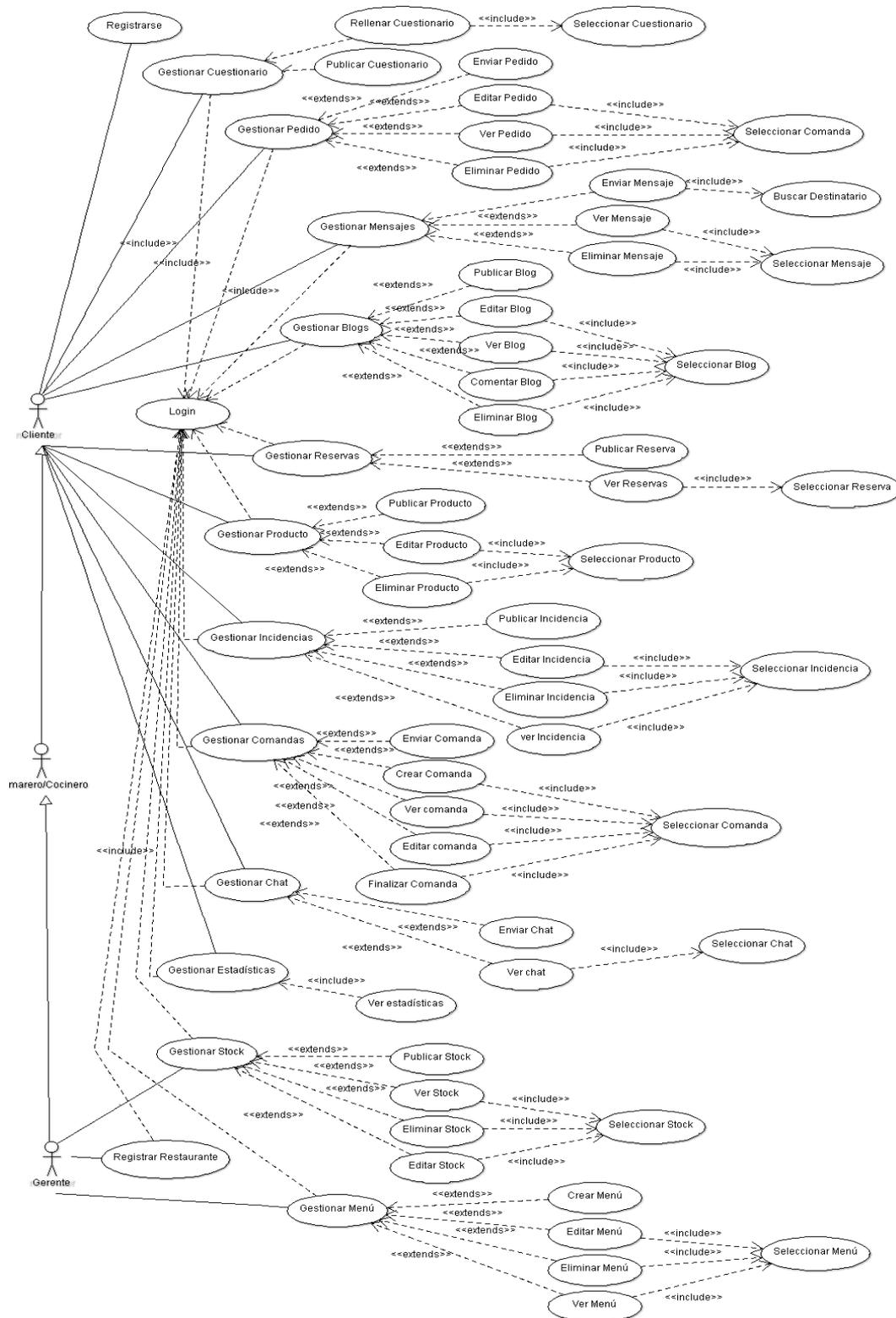


Figura 3-3: Diagrama de casos de uso

### 3.4 Diagrama de secuencia

Los diagramas de secuencia representan gráficamente la iteración de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

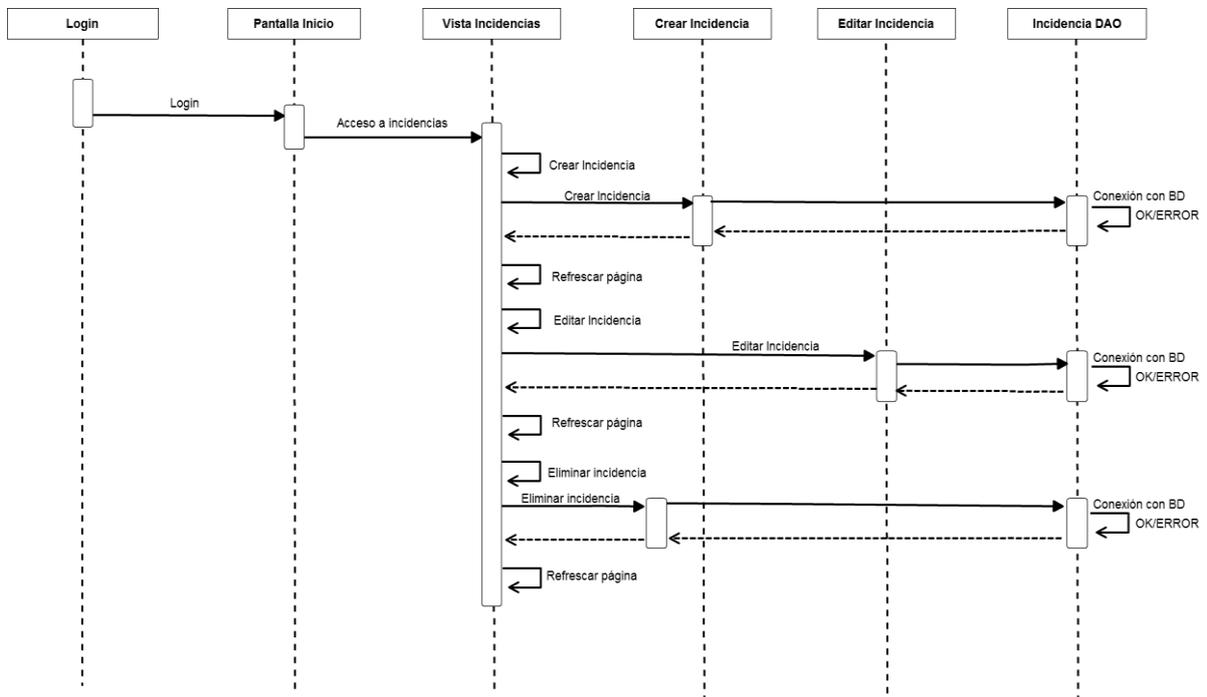


Figura 3-4: Diagrama de secuencia

Para acceder a cada uno de los módulos lo primero que hay que hacer es loguearse en el sistema. Una vez logueados desde la página de inicio, se selecciona en la barra de tareas la pestaña 'Gestiones → incidencias'. En la vista de las incidencias se puede: (1) pulsar el botón crear incidencia donde redireccionará a otra página para rellenar el formulario de la incidencia y una vez terminada se invoca al controlador correspondiente y se publica actualizando la vista de las incidencias, (2) pulsar el botón editar de una incidencia seleccionada en la vista de las incidencias que nos redireccionará a otra página para editar la información de la incidencia y una vez terminado se invoca al controlador correspondiente, se publica y se actualiza la página de la vista de las incidencias con la incidencia modificada y (3) eliminación de una incidencia, para ello se selecciona la incidencia y se pulsa eliminar, se invoca al controlador correspondiente y se actualiza la página de las vistas de incidencias. Todas estas peticiones que hemos mencionado como crear, editar y eliminar una incidencia realizarán su respectiva operación en la base de datos, dando como resultado *OK* en caso de que se haya realizado la operación satisfactoriamente o *ERROR* en caso contrario.

# 4 Desarrollo

---

## 4.1 Introducción

En esta sección se va a explicar cómo se han implementado los diferentes componentes de la aplicación. Recordamos que para el desarrollo hemos utilizado el lenguaje de programación interpretado, reflexivo y orientado a objetos “Ruby” y el framework “On Rails” para el acceso a la base de datos. Hemos elegido como herramienta de desarrollo para nuestra aplicación el editor de código multiplataforma “Sublime Text” y el Sistema Operativo Ubuntu 16.0.

## 4.2 Conocimientos previos acerca de Ruby on Rails

Al no haber utilizado una plataforma de desarrollo como *NetBeans*, *eclipse...*, sino un editor de textos, antes de comenzar a explicar la estructura del sistema es importante tener algunos conocimientos previos para la ejecución de un proyecto en RoR, que nos servirá de lección para visualizar los principales pasos que hemos ido siguiendo para la puesta en marcha de nuestro proyecto. Primero veremos una guía de instalación de RoR [ANEXO C] y a continuación mostraremos unas breves indicaciones de los pasos seguidos para la iniciación de nuestro proyecto:

1. **Creación de un nuevo proyecto:** Para ello se tendrá que ejecutar en la terminal, en el directorio donde queramos crear el proyecto, el siguiente comando: *rails new <nombre\_proyecto>*. Esto genera un directorio con una serie de subcarpetas [ANEXO D]. Los posteriores comandos se deberán de realizar desde la terminal en el directorio del proyecto que se acaba de crear.
2. **Inicialización del servidor web de Rails:** Para que el servidor de aplicaciones este corriendo es necesario ejecutar el siguiente comando desde la terminal: *rails server*.
3. **Utilización del comando Rake:** Previo a la utilización de este comando, se ha debido definir la estructura de una base de datos en el directorio */db/migrate* como, por ejemplo, la tabla *Products* de la Figura 4-2. En este caso utilizando el manual del comando en el que se podrá ver las distintas combinaciones de comandos disponibles, aunque solo mencionaremos tres de ellos que son lo que hemos utilizado para la gestión de la base de datos del proyecto:
  - a. **Comando para la creación de la base de datos:** *rake db:create*
  - b. **Comando para la migración de la base de datos:** *rake db:migrate*
  - c. **Comando para cargar los datos del fichero *db/seeds.rb*:** *rake db:seed*
  - d. **Comando para el borrado de la base de datos:** *rake db:drop*

Estos han sido los principales pasos a seguir para la creación de nuestro proyecto.

### 4.3 Patrón Modelo-Vista-Controlador

Para el desarrollo de nuestra aplicación web se ha decidido seguir un patrón de arquitectura software de MVC que consiste en separar los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones [16]. A continuación, se muestra una breve explicación de cada una de las capas de este patrón:

- **Modelo:** Capa donde se trabaja con los datos, habitualmente contenidos en una base de datos, por lo que en los modelos tendremos las funciones que accederán a las tablas y harán las correspondientes consultas.
- **Vista:** Como su propio nombre indica, contiene el código de nuestra aplicación que va a producir la visualización de las interfaces al usuario, es decir, el código que permitirá renderizar los estados de nuestra aplicación en HTML.
- **Controlador:** Contiene el código necesario para responder a las acciones que se solicitan en la aplicación. En realidad, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación.

Esta modularización del código, donde en cada una de las capas se realiza una tarea determinada, facilita la localización de cualquier problema dada su distribución por capas, permitiendo en un futuro mejorar el desarrollo y sobre todo un mantenimiento del mismo de manera sencilla.

En la Figura 4-1 se muestra una representación esquemática y aclarativa del mencionado patrón MVC:

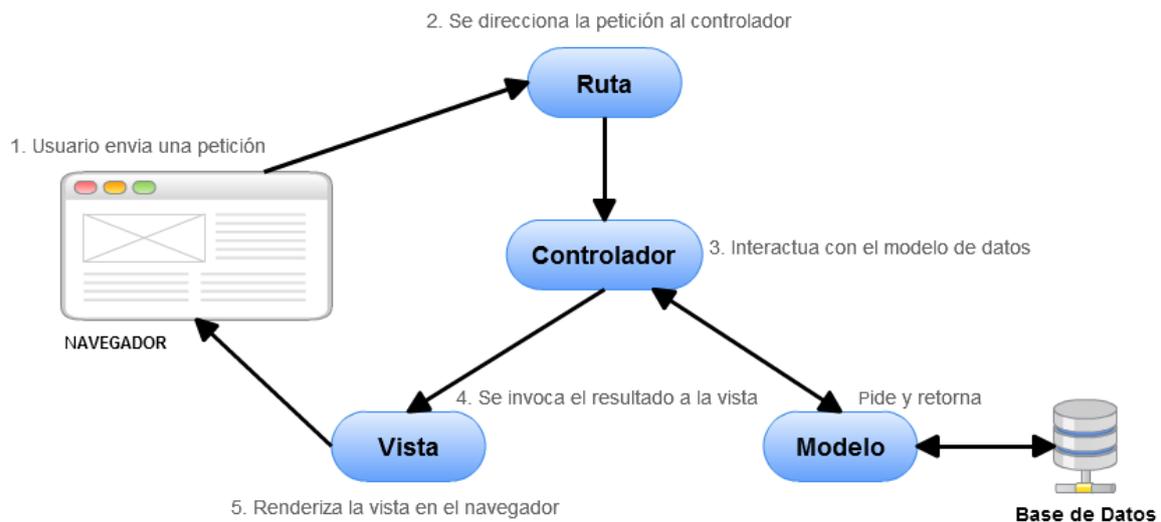


Figura 4-1: Esquema patrón Modelo-Vista-Controlador

## 4.4 Modelo

El diseño de la aplicación se ha realizado desde la parte más interna hasta la capa más visible, es decir, desde la base de datos hasta la interfaz de usuario. Se tomó esta decisión ya que lo más importante de un buen desarrollo es que la base del proyecto, en este caso, la base de datos tiene que estar perfectamente definida, facilitando así la implementación de los demás módulos y dejando la posibilidad de realizar cambios basados en la lógica de negocio y en la interfaz.

Como ya mostramos en el apartado anterior, la base de datos de nuestra aplicación *NoteTheCommands* está formada por 22 tablas cada una de las cuales almacena información generadas a diario por los restaurantes registrados en el sistema.

En cuanto al modelado de datos distinguimos dos tipos, una de unión entre tablas y otra de registros. Las tablas de unión, conocidas como *join\_table*, se utilizan para enlazar o unir varias tablas mediante las claves externas que referencian las dos tablas unidas, mientras que las tablas de registro almacenan cada uno de los estados que se vayan introduciendo a lo largo de la vida del sistema.

Antes de comenzar a ver la información almacenada en la base de datos, cabe mencionar que se pensó en la posibilidad de implementar la lógica de negocio en la propia base de datos, empleando *Triggers* para que se ejecuten cada vez que se llama al disparador desde la tabla asociada al *Trigger*. A pesar de saber que se trata de una implementación muy eficiente de manejar los datos, se ha decidido que esta funcionalidad se desarrolle en la lógica de la aplicación facilitando su mantenimiento, portabilidad y reusabilidad.

Cada tabla creada en la base de datos tiene asignado por defecto como clave primaria un identificador numérico único llamado *id*, cuya finalidad es tener mayor eficiencia a la hora de trabajar con la tabla. En la Figura 4-2 se muestra un ejemplo sencillo de creación de una tabla *Products* para nuestro sistema:

```
class CreateProducts < ActiveRecord::Migration[5.0]
  def change
    create_table :products do |t|
      t.string :name, null: false
      t.integer :type_product, null: false
      t.text :ingredients
      t.text :preparation
      t.float :price, null: false

      t.boolean :younger, null: false

      t.references :restaurant

      t.timestamps
    end
  end
end
```

Figura 4-2: Ejemplo del código para crear la base de datos de la tabla *Products*

Las tablas *Join* no presentan un identificador único por tanto en caso de ser necesario, se ha añadido a mano como muestra el ejemplo de la tabla *CommandProducts* en la Figura 4-3.

```
class CreateCommandProducts < ActiveRecord::Migration[5.0]
  def change
    create_join_table :command, :products do |t|
      t.primary_key :id
      t.index :product_id
      t.index :command_id

      t.integer :quantity, null: false
      t.boolean :is_served, default: false
      t.boolean :is_cooked, default: false

      t.timestamps
    end
  end
end
```

Inserción manual de la Clave Primaria en la Tabla *CommandProducts* que relaciona un Producto y una Comanda determinada

Figura 4-3: Código para crear la base de datos de la tabla *Join CommandProducts*

Como hemos podido observar en las tablas de la tabla de la Figura 4-2 y la Figura 4-3 se aprecia que hemos añadido el campo *timestamps*. Este campo genera dos atributos, un campo *created\_at* que almacena automáticamente la fecha y la hora del sistema en la inserción del elemento en la tabla y un campo *updated\_at* que almacena automáticamente la fecha y la hora de la última modificación del elemento en base de datos.

#### 4.4.1 Modelo de Usuarios

Una vez vista la estructura y desarrollo de la base de datos del sistema pasaremos a ver un ejemplo de la capa modelo y ver como interactúa dicha capa con la base de datos generada. Para ello hemos elegido el modelo de los usuarios donde se explicará en detalle el funcionamiento del código en la Figura 4-4.

```
class User < ApplicationRecord
  # Include default devise modules. Others available are:
  # :confirmable, :lockable, :timeoutable and :omniauthable
  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :trackable, :validatable

  #Relacion uno a muchos
  has_many :restaurant_users
  has_many :emails
  has_many :restaurants, through: :restaurant_users
  has_many :authored_conversations, class_name: 'Conversation', foreign_key: 'author_id'
  has_many :received_conversations, class_name: 'Conversation', foreign_key: 'received_id'
  has_many :personal_messages, dependent: :destroy

  #Validacion de los paramtros insertados en el registro
  validates :user_name, length: { maximum: 30 }

  def age
    (Time.zone.now - self.birth_date.to_datetime).to_i / 1.year
  end
end
```

Figura 4-4: Modelo de Usuarios

Como se observa en la Figura 4-4, tenemos la variable *has\_many* para generar la asociación de uno a muchos, la variable *validates* para validar los parámetros introducidos en los formularios que se rellenan en la vista del usuario y por último la definición del método *age* que te devuelve la edad del usuario mediante la fecha de nacimiento previamente introducida.

## 4.5 Vista

El desarrollo de las vistas de la aplicación está implementado en ficheros HTML, JavaScript y CSS: (1) HTML que nos permite mostrar el contenido de la página web, (2) JavaScript que nos permite dotar de dinamismo a la interfaz y (3) CSS que nos permite dar estilo a la página.

Para dar funcionalidad a la aplicación se deberá incrustar el lenguaje Ruby en las páginas HTML y por ello cabe destacar la palabra reservada en RoR '*yield*' que nos permite ejecutar código dentro de un método incluso dentro de otro bloque. Esto lo hemos utilizado para crear un patrón igual para todas las vistas compuesto por el logo de la empresa, el menú de código de barras y el pie de página, dejando que el cuerpo sea donde se cargue las demás vistas de la aplicación. En la Figura 0-6 del manual de usuario se ilustra la página de inicio con todas las zonas aquí señaladas.

```
</ul>
</nav>
<section>
  <%= yield %>
</section>
</br>
<footer>
  <div class="eps">
    <%= image_tag('logoEPS.png', :alt => 'logoEPS')%>
  </div>
  <div class="uam">
    <%= image_tag('logoUAMblanco.png', :alt => 'logoUAM')%>
  </div>
  <p>- UAM - EPS - Ingeniería Informática - TFG - Daniel Ramos Trejo -</p>
  <p>Contacto: daniel.ramost@estudiante.uam.es</p>
</footer>
```

Palabra reservada en Ruby on Rails que ejecuta código de otro bloque en el cuerpo de la página

Figura 4-5: Uso de *yield* en nuestro proyecto

### 4.5.1 Vista de localización de la mesa

A continuación, veremos un ejemplo de un fichero HTML como, por ejemplo, el correspondiente a rellenar los datos para la localización de la mesa de una comanda del restaurante. Este fichero hará de bloque para la palabra reservada '*yield*' que mencionamos anteriormente, es decir, la vista localización será el cuerpo de la página. En la Figura 4-6 se muestra el bloque de la vista localización y una breve descripción del código.

```

<div class="muestra-tabla">
<table class="pretty">
  <tr>
    <th><%= "Lugar" %></th>
    <th><%= "Mesa" %></th>
    <th><%= "Precio total de la comanda" %></th>
    <th><%= "Restaurante" %></th>
    <th><%= "Usuario" %></th>
  </tr>
  <%= @locations.each do |location| %>
    <tr>
      <td><%= Location::LOCATION_CATEGORY[location.place] %></td>
      <td><%= location.sit %></td>
      <td><%= location.total_price_command %></td>
      <td><%= @current_restaurant %></td>
      <td><%= current_user.user_name %></td>
      <td><%= link_to "Edit Location", edit_location_path(location) %></td>
    </tr>
  <%= end %>
</table>
</div>
<div class="muestra-opciones">
  <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>
  <script type="text/javascript">
    jQuery(document).ready(function() {
      jQuery("a.mostrar").click(function() {
        jQuery("div.chatbox").toggle("slow");
        return false;
      });
    });
  </script>
  <a href="#" class="mostrar">Añadir una nueva mesa</a><br/>
  <div class="chatbox">
    <%= render 'form' %>
    <p><%= link_to "Back to List", restauran_users_path %></p>
  </div>
</div>

```

Atributos de la tabla para la localización de una mesa

Recorrer todas las localizaciones del restaurante

Imprimir los valores de cada iteración de todas las localizaciones del restaurantes

Script para marcar como ocupada, seleccionada o vacía una mesa del restaurante

Redirección al fichero para visualizar las mesas del restaurante

Figura 4-6: Localización de la comanda

## 4.6 Controlador

Para el desarrollo del controlador hemos declarado variables globales que utilizaremos en cada una de las vistas de los módulos que hemos implementado. Estas variables globales pueden contener el resultado de una consulta a base de datos y que posteriormente en la vista recorreremos para imprimir los distintos valores de la consulta.

### 4.6.1 Controlador de incidencias

En este fichero cada una de las definiciones son métodos que se utilizarán en las páginas HTML de las vistas que definen el propio método, es decir, en el caso de *index*, todo lo que se declare en él serán las variables globales que podremos utilizar en la vista *index.html* de incidencias y así sucesivamente. Las Figuras 4-7 y 4-8 muestran el caso del controlador de las incidencias.

```

class IncidencesController < ApplicationController
  def index
    @incidences = Incidence.where("restaurant_id = ?", "#{@current_restaurant.id}").order("priority DESC")
    .paginate(:per_page => 7, :page => params[:page])
    @incidences_solved = Incidence.where("restaurant_id = ? AND is_solved = true", "#{@current_restaurant.id}").order("priority DESC")
    .paginate(:per_page => 7, :page => params[:page])
    @incidences_pending = Incidence.where("restaurant_id = ? AND is_solved = false", "#{@current_restaurant.id}").order("priority DESC")
    .paginate(:per_page => 7, :page => params[:page])

    @incidences_waiter = Incidence.where("restaurant_id = ? AND category = 0", "#{@current_restaurant.id}").order("priority DESC")
    .paginate(:per_page => 7, :page => params[:page])
    @incidences_solved_waiter = Incidence.where("restaurant_id = ? AND is_solved = true AND category = 0", "#{@current_restaurant.id}").
    .order("priority DESC").paginate(:per_page => 7, :page => params[:page])
    @incidences_pending_waiter = Incidence.where("restaurant_id = ? AND is_solved = false AND category = 0", "#{@current_restaurant.id}").
    .order("priority DESC").paginate(:per_page => 7, :page => params[:page])
    @incidence = Incidence.new
  end

  def show
    @incidence = Incidence.find(params[:id])
  end

  def new
    @incidence = Incidence.new
  end

  def create
    @incidence = Incidence.new(incidence_params)
    if @incidence.save
      flash[:notice] = "Successfully created incidence."
      redirect_to @incidence
    else
      render :action => 'new'
    end
  end

  def edit
    @incidence = Incidence.find(params[:id])
  end
end

```

Figura 4-7: Controlador de las incidencias 1

```

def update
  @incidence = Incidence.find(params[:id])
  if @incidence.update_attributes(incidence_params)
    flash[:notice] = "Successfully updated incidence."
    redirect_to @incidence
  else
    render :action => 'edit'
  end
end

def destroy
  @incidence = Incidence.find(params[:id])
  @incidence.destroy
  flash[:notice] = "Successfully destroyed incidence."
  redirect_to incidences_url
end

private

def incidence_params
  params.require(:incidence).permit(:category, :title, :priority, :description, :is_solved, :date_solution, :restaurant_id, :user_id)
end
end

```

Figura 4-8: Controlador de las incidencias 2

En la Figura 4-7 se observa, en la definición del método *index*, la declaración de variables globales que vamos a utilizar en las vistas. Estas variables globales son consultas que están formadas, en este caso de ejemplo, por la tabla *Incidnece* sobre la que se van a aplicar los siguientes métodos: (1) método *where* para filtrar por los valores que nos interese, (2) método *order* para ordenar el resultado de la consulta y (3) método *paginate* que página la tabla mostrando un número de registros específico. Para este último método hemos utilizado la gema *will\_paginate* (API simple que proporciona consultas paginadas en RoR). Estas gemas de Ruby son paquetes de librerías para Ruby que se instalan en el sistema y quedan listas para ser usadas, con un simple *'require'* o con mecanismos que aporta el propio sistema de gemas para Ruby.



# 5 Integración, pruebas y resultados

---

## 5.1 Pruebas

El objetivo de las pruebas es comprobar que todos los componentes del sistema funcionan correctamente. Para ello se ha llevado a cabo un plan de *Quality Assurance* (QA) formado por pruebas de inspección de código, testeo unitario, test de integración, pruebas de la interfaz gráfica, prueba de stress, test funcional y test de aceptación.

### 5.1.1 Desarrollo del plan de pruebas

Uno de los aspectos importantes del desarrollo de software es lo que conocemos hoy en día como *Quality Assurance*, QA, o lo que es lo mismo, aseguramiento de la calidad del software. Se trata de un conjunto de actividades de evaluación de las distintas etapas del proceso de desarrollo para garantizar que el producto final sea de calidad. El concepto de calidad se presta a múltiples interpretaciones, pero siempre implica que el software satisfaga las necesidades del cliente.

El plan de QA atraviesa el proceso de desarrollo desde el nacimiento de la idea hasta la implementación del software. En las primeras etapas, se verifica que los objetivos estén bien planteados y que los requerimientos sean precisos. En las fases de diseño y codificación, se vigila el cumplimiento de los estándares fijados. Finalmente, se revisa que el software en funcionamiento respete los requerimientos solicitados y que la entrega al cliente se haga en las condiciones adecuadas [17].

En el plan que se ha diseñado queremos cerciorarnos del correcto funcionamiento de todos los componentes como son: la base de datos, los controladores y la parte de las vistas de la aplicación. Para ello seguiremos el siguiente plan:

#### **Prueba de inspección de código**

Esta prueba consiste en una inspección minuciosa de los módulos, cada vez que se ha terminado la programación, para detectar posibles errores de código que puede observarse a simple vista. Esta revisión es importante sobre todo en la capa de controlador ya que es ahí donde se realizan las *queries* de las consultas y donde se filtran los datos con los que se va a trabajar y posteriormente mostrar en la página web.

Al realizar dichas pruebas se han encontrado errores como por ejemplo al definir las variables globales en Ruby, que se distingue de las variables normales en que se pone un '@' delante del nombre de la variable y esta puede ser visible desde cualquier parte del código perteneciente a su módulo. También se han encontrado errores del estilo de impresión por pantalla, ya que Ruby cuando quiere imprimir el contenido de una variable utiliza `<%= variable %>` y si quitamos el '=' no imprime el contenido, evitando así la impresión de caracteres o variables utilizados para debuggear.

## **Testeo unitario**

Estas pruebas unitarias se encargan de probar cada módulo de forma individual. Se ha hecho especial hincapié en este tipo de pruebas en los módulos que tienen conexiones a la base de datos y sus respectivos controladores para asegurarnos de su correcto funcionamiento. Se distinguen dos tipos de pruebas que son las de caja blanca y las de caja negra. Las pruebas de caja blanca están orientadas a las funciones internas de un módulo y dichas pruebas se han realizado desde la consola de Rails (para abrir la consola basta con ejecutar desde la terminal *'rails console'*). Las pruebas de caja negra se han realizado desde la propia web cerciorándonos del cumplimiento de todos los requisitos funcionales.

Al realizar dichas pruebas en nuestro código se han encontrado errores en las consultas, que se realizan en los controladores, para acceder a la base de datos. Estos errores consistían, por ejemplo, que para toda consulta realizada había que filtrar por el restaurante en el que el usuario se encontraba logueado, evitando así que los productos de un restaurante estuvieran visibles en las tablas de todos los restaurantes.

## **Test de integración**

Una vez terminadas las pruebas unitarias de los módulos, se han realizado las pruebas de integración. Dichas pruebas tienen como objetivo probar que todos los elementos unitarios, una vez unidos en bloque, en su conjunto funcionan correctamente. Primero se ha comprobado que la integración de los módulos ha sido satisfactorio y seguidamente se han realizado test integrando todos los módulos para cerciorarnos de su correcto funcionamiento.

Al realizar estas pruebas no se han encontrado problemas relevantes ya que, al pasar las pruebas unitarias correctamente, las de integración funcionaron a la primera.

## **Prueba de la interfaz gráfica**

Las pruebas de interfaz gráfica se encargan de comprobar la correcta visualización de cada uno de los elementos de la página web como, por ejemplo, la correcta ubicación de los botones, que una imagen se vea pixelada, que no pierda la forma si la agrandamos o reducimos la ventana del navegador, etc.

Al realizar dichas pruebas en nuestra aplicación se han detectado errores en la carga de imágenes o en la vista de algunas tablas que se salían del contenedor. También algunos botones dependiendo del tamaño de la ventana del navegador se ubicaban en zonas donde no deberían como, por ejemplo, los botones de *'dar de alta un restaurante'*.

## **Prueba de stress**

Las pruebas de stress se caracterizan por poner a prueba la resistencia de la aplicación enviándoles una cantidad de peticiones excesiva, para buscar un posible colapso de la aplicación.

Para ello se ha creado un script con unas 2000 inserciones a las distintas tablas. Posteriormente se ha lanzado el script desde la línea de comandos y hemos verificado el correcto funcionamiento de la prueba viendo que las tablas contienen los datos introducidos y observando que los *Logs* del sistema son los esperados y no hay errores de inserción.

### **Test funcional**

Esta prueba consiste en comprobar que el software ofrece las funciones solicitadas en la captura de requisitos.

Para ello se ha ido comprobando que se cumplen cada uno de los requisitos capturados previamente.

### **Test de aceptación**

Se trata de la última prueba de todas ya que es la que realiza el cliente en nuestro caso para verificar que el producto satisface sus expectativas. En el momento de la defensa de este TFG, esta prueba no se ha realizado, pero se llevará a cabo.

## **5.1.2 Prueba para generar una comanda**

Como la aplicación dispone de varios tipos de usuario (cliente, camarero, cocinero o gerente), utilizaremos un borde de color distintivo para diferenciarlos visualmente. En caso de que la ventana sea accedida por cualquier usuario no se visualizará color de borde. Los colores elegidos son los que se muestran en cada una de las palabras identificadoras del tipo de usuario.

**Cliente**      **Cocinero**      **Camarero**      **Gerente**      **Administrador**

Para la realización de esta prueba se va a representar una secuencia de trabajo concurrente entre los distintos perfiles, se va a ilustrar esta secuencia de la Figura 5-1 a la Figura 5-12. Lo primero que haremos será registrarnos como nuevo cliente del sistema, para ello en la pantalla de inicio pulsaremos el botón registrar como nuevo usuario y rellenamos el formulario (vamos a realizar un registro especial ya que se trata de una persona menor de edad con lo que ello puede implicar y que veremos más adelante). La Figura 5-1 muestra cómo se vería la página relativa a un nuevo registro.



**Figura 5-1: Registro de una menor de edad**

Seguidamente nos apuntaremos al restaurante *El vidueño*, que será donde vamos a realizar la comanda para la prueba (Figura 5-2, observad que la Figura tiene un recuadro en tono azul correspondiendo con un perfil de usuario tipo cliente).



**Figura 5-2: Apuntarse a un restaurante**

A continuación, seleccionamos la mesa donde nos vamos o hemos sentado, dándole un nombre a la comanda y una breve descripción como por ejemplo si hay alguien celiaco, o alérgico a un producto (Figura 5-3).



Figura 5-3: Seleccionar la mesa para la comanda

Una vez seleccionada la mesa pasaremos a agregar los productos que vamos a consumir, para ello vamos añadiendo productos y es aquí donde entra en juego la edad del cliente. En este caso si intentamos agregar un producto que no está a la venta para los menores de edad, como es el caso de una cerveza con alcohol nos mostrará un mensaje de alerta y no nos permitirá añadir dicha consumición a la comanda (ver Figura 5-4). En caso de introducir un producto sin una cantidad, no se nos agregará el producto a la lista de productos de la comanda. También podemos ver que si no se han servido los productos no nos mostrará el botón de 'Cerrar Comanda', el cual manda un aviso al camarero diciéndole que la comanda ha sido finalizada y poder gestionar la cuneta de la comanda.



Figura 5-4: Alerta de prohibición para la venta a menores

En el caso de que el cliente quiera salir de la aplicación teniendo una comanda abierta se le mostrará un mensaje de alerta comunicándole que tiene comandas abiertas y no le permitirá cerrar sesión (Figura 5-5).

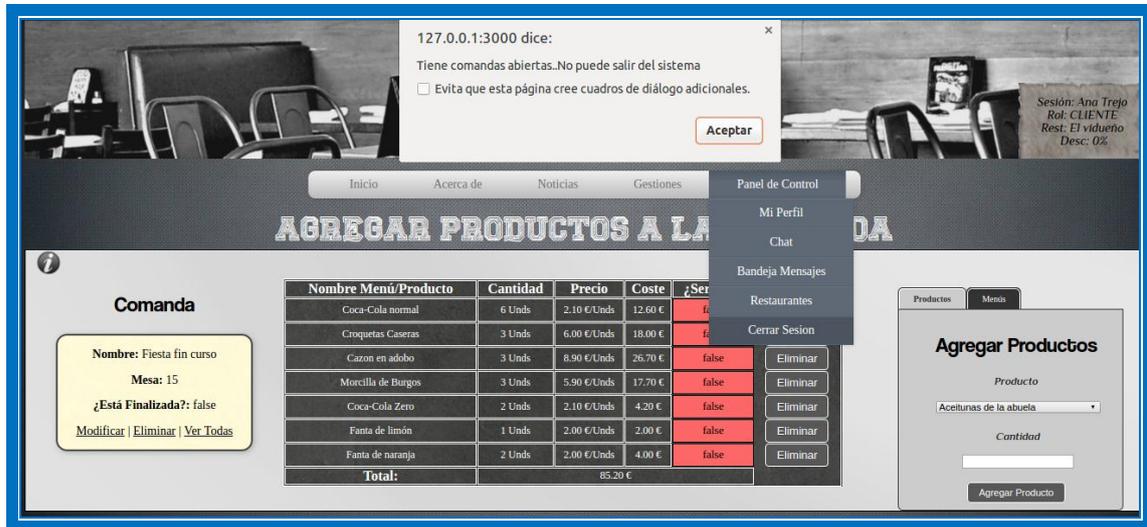


Figura 5-5: Alerta al cerrar sesión

Si nos centramos, ahora en el perfil de camarero del restaurante, con contorno representado en color verde – Figura 5-6 -, puede visualizar las comandas que hay abiertas y también servir los productos dependiendo de si están o no cocinados. Como es obvio las bebidas no hacen falta cocinarlas por eso todas ellas deberían mostrarse con la bandera de cocinado a true para que le cocinero lo pueda servir.



Figura 5-6: Comanda del restaurante

Si pulsamos el botón 'Ver Comanda' se podrán visualizar los productos de la comanda y prepararlos para servir. Una vez listos y servidos en la mesa pulsaremos el botón 'Servir'. Para los productos que son comida y necesitan que los prepare el cocinero, con contorno representado en color marrón – Figura 5-7 -, tendremos que esperar a que el cocinero habilite la bandera de *cocinado* a true para poder servirlo a la mesa.



Figura 5-7: Cocinero cocina los productos de comida

A continuación, vemos el desglose de la comanda tras pulsar en el botón 'Ver Comanda' y se podrá ver que hay productos que están cocinados, pero no servidos, pudiendo ser estos servidos a la mesa.



Figura 5-8: Desglose de la comanda para el camarero

Una vez servido todos los productos de la comanda, el cliente podrá o bien seguir añadiendo productos, o bien visualizar y pulsar el botón para cerrar la comanda, viendo el desglose de la lista de productos pedidos en la comanda y su coste total (Figura 5-9).



Figura 5-9: Cierre de la comanda por el cliente

Después de pulsar el botón ‘Cerrar Comanda’ nos aparecerá un breve cuestionario del restaurante para aportar información al gerente de forma anónima (Figura 5-10). Este cuestionario va a permitir al agente realizar toma de decisiones basadas en las opiniones de sus propios clientes.

Sesión: Ana Trejo  
Rol: CLIENTE  
Rest: El viduero  
Desc: 0%

Inicio Acerca de Noticias Gestiones Panel de Control

### NUEVA VOTACIÓN

¿Qué le ha parecido la aplicación?

Muy mal Mal Aceptable Buena Muy buena Excelente

¿Qué le ha parecido el servicio del camarero?

Muy mal Mal Aceptable Buena Muy buena Excelente

¿Qué le ha parecido el restaurante?

Muy mal Mal Aceptable Buena Muy buena Excelente

¿Qué tal ha estado la comida?

Muy mal Mal Aceptable Buena Muy buena Excelente

**Figura 5-10: Rellenar cuestionario por parte del cliente**

Mientras el cliente rellena el formulario, el camarero puede revisar que todo se ha servido y podrá pulsar el botón ‘Imprimir Comanda’ para imprimir el ticket que deberá entregar al cliente y que éste abone el coste de la comanda, bien en efectivo o bien con tarjeta.

Sesión: Cristina Ruiz  
Rol: CAMARERO  
Rest: El viduero  
Desc: 0%

Inicio Acerca de Noticias Gestiones Panel de Control

### NÚMERO COMANDA: 6

**Comanda**

Nombre: Fiesta fin curso  
Mesa: 15  
¿Está Finalizada?: true

**Imprimir Factura**

Nombre Menú/Producto	Cantidad	Precio	Coste	¿Servido?	¿Cocinado?
Coca-Cola normal	6 Unids	2,10 €/Unid	12,60 €	true	true
Croquetas Galeras	3 Unids	6,00 €/Unid	18,00 €	true	true
Cazon en adobo	3 Unids	8,90 €/Unid	26,70 €	true	true
Morcilla de Burgos	3 Unids	5,90 €/Unid	17,70 €	true	true
Coca-Cola Zero	2 Unids	2,10 €/Unid	4,20 €	true	true
Fanta de limón	1 Unid	2,00 €/Unid	2,00 €	true	true
Fanta de naranja	2 Unids	2,00 €/Unid	4,00 €	true	true
<b>Total:</b>			<b>85,20 €</b>		

**Figura 5-11: Impresión del ticket para el cliente de la comanda**

En la Figura 5-12 se muestra el ticket de la comanda de prueba que hemos estado realizando.

Restaurante El vidueño			
Dirección: Avenida de Andalucía, 2 San Sebastian de los Reyes, Madrid 28702			
Tel: 916519761			
Fecha: 2017-05-02 11:57:54			
Camarero: Cristina Ruiz			
Factura: 6			
Nombre: Fiesta fin curso			
Mesa: 15			
Descuento Adicional: 0%			
Descripción	Cantidad	P.Unidad	Importe
- Coca-Cola normal	6	2.10€	12.60€
- Croquetas Caseras	3	6.00€	18.00€
- Cazon en adobo	3	8.90€	26.70€
- Morcilla de Burgos	3	5.90€	17.70€
- Coca-Cola Zero	2	2.10€	4.20€
- Fanta de limón	1	2.00€	2.00€
- Fanta de naranja	2	2.00€	4.00€
<b>Total:</b>	*		85.20€
		Con descuento:	85.20€
			*21% Iva incluido
***** GRACIAS POR SU VISITA *****			

**Figura 5-12: Ticket de la comanda**

## 5.2 Resultados

Tras realizar todas estas pruebas hemos obtenido unos resultados satisfactorios utilizando todos los métodos nombrados anteriormente y por tanto hemos obtenido una versión del sistema totalmente operativa y que asegura la calidad del software de la funcionalidad especificada en los requisitos tanto funcionales como no funcionales. Entre los objetivos del plan de pruebas que hemos realizado cabe destacar:

- Descubrimiento de errores en la función, la lógica o la implementación de cualquier representación del software.
- Verificar que el software alcanza los requisitos.
- Garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos.
- Conseguir un software desarrollado en forma uniforme.

Todos estos procedimientos son indispensables para asegurar la calidad de un producto y evitar así errores que en producción podrían ser catastróficos. También cabe destacar que un buen desarrollo software ayuda en el futuro en su mantenibilidad y sobre todo ahorraría sobrecostes del proyecto.



# 6 Conclusiones y trabajo futuro

---

## 6.1 Conclusiones

La elaboración de este proyecto me ha permitido realizar un proceso de Ingeniería del Software para el desarrollo de una aplicación web. Dicho trabajo ha consistido en el diseño e implementación de una herramienta para facilitar la gestión de un restaurante.

Al ser una aplicación encargada por un cliente, muchas de las ideas implementadas son gracias a la experiencia tras muchos años en la hostelería y por ello las primeras reuniones fueron muy importantes. También el estudio del arte nos ha ayudado a captar algunas ideas de aplicaciones ya existentes, aunque hemos podido observar que éstas están orientadas a los empleados y gerente, pero no tanto a los clientes del restaurante.

Este trabajo me ha proporcionado multitud de conocimientos de programación web y sobre todo el uso del framework RoR que es una herramienta completamente nueva para mí y que he tenido que aprender partiendo de cero. Por ello el tiempo estipulado para la elaboración del proyecto se ha extendido algo más de lo esperado [ANEXO E].

## 6.2 Trabajo futuro

Tras el desarrollo del proyecto se ha obtenido una primera versión operativa y que cumple con los requisitos mínimos de funcionalidad básica. A pesar de ello se ha definido un conjunto de ampliaciones para mejorar ciertos módulos del sistema. A continuación, se especificarán las mejoras que se quieren realizar a medio plazo:

1. Mejorar la parte de diseño del mapa de un restaurante para la selección de la comanda. Se procederá a solicitar, al restaurante que se va a dar de alta, una maqueta con la ubicación del mobiliario en el restaurante.
2. En la inclusión de un menú completo en una comanda (incluidos todos los productos que forman el menú) se deberá poder elegir entre los productos del menú que el cliente desee.
3. Crear un sistema que gestione y de prioridad a las incidencias de un restaurante, evitando que sea el propio usuario el que la determine.
4. Generar más estadísticas como, por ejemplo, que el gerente pueda visualizar los gastos de su restaurante y así tener un mayor control de los mismos.
5. Incluir una selección de idiomas para cualquier tipo de usuario pueda moverse por la aplicación sin problemas.
6. Debido a que muchos restaurantes no van a tener la posibilidad de instalar Tablet's en las mesas, se desea desarrollar una versión móvil para que cualquier usuario con dicho dispositivo pueda acceder a ella.



## Referencias

---

- [1] Hosteltáctil. Available: <http://www.hosteltactil.com>. [Último acceso: Enero 2017].
  - [2] Dual Link Bar & Restaurant. Available: <http://www.dual-link.es/>. [Último acceso: Enero 2017].
  - [3] PHP. Available: <https://es.wikipedia.org/wiki/PHP>. [Último acceso: Enero 2017].
  - [4] JSF. Available: [https://es.wikipedia.org/wiki/JavaServer\\_Faces](https://es.wikipedia.org/wiki/JavaServer_Faces). [Último acceso: Enero 2017].
  - [5] ASP.NET. Available: <https://es.wikipedia.org/wiki/ASP.NET>. [Último acceso: Enero 2017].
  - [6] Django. Available: <https://www.djangoproject.com/>. [Último acceso: Enero 2017].
  - [7] Ruby on Rails. Available: [https://es.wikipedia.org/wiki/Ruby\\_on\\_Rails](https://es.wikipedia.org/wiki/Ruby_on_Rails). [Último acceso: Enero 2017].
  - [8] Ruby on Rails. Available: <http://rubyonrails.org.es/>. [Último acceso: Febrero 2017].
  - [9] RubyGems. Available: <https://es.wikipedia.org/wiki/RubyGems>. [Último acceso: Febrero 2017].
  - [10] HTML5. Available: <https://www.w3.org/TR/html5/>. [Último acceso: Febrero 2017].
  - [11] CSS3. Available: <https://www.w3.org/TR/css-style-attr/>. [Último acceso: Febrero 2017].
  - [12] JavaScript. Available:
  - [13] MySQL. Available: <https://www.mysql.com/>. [Último acceso: Febrero 2017].
  - [14] Workbench. Available: <https://www.mysql.com/products/workbench/>. [Último acceso: Febrero 2017].
  - [15] GitHub. Available: <http://conociendogithub.readthedocs.io/en/latest/data/introduccion/>. [Último acceso: Febrero 2017].
  - [16] MVC. Available: <https://es.wikipedia.org/wiki/Modelo%20%93vista%20%93controlador>. [Último acceso: Marzo 2017]
- QA. Available: <http://www.4rsoluciones.com/blog/que-es-un-plan-de-qa-2/> [Último acceso: Abril 2017].



## Glosario

---

<b>API</b>	<i>Application Programming</i> . Interface. Interfaz de programación entre componentes software.
<b>RoR</b>	<i>Ruby on Rails</i> . Framework de aplicaciones web de código abierto escrito en lenguaje de programación Ruby.
<b>Framework</b>	Estructura de soporte definida utilizada para organizar y desarrollar un proyecto software.
<b>MVC</b>	<i>Model-View-Controller</i> . Patrón de arquitectura software que separa los datos y la lógica de una aplicación de la interfaz de usuario y el modelo encargado de gestionar los eventos y las comunicaciones
<b>MySQL</b>	Sistema de gestión de bases de datos relacional, multihilo y multiusuario.
<b>SQL</b>	<i>Structured Query Language</i> . Lenguaje declarativo de acceso a base de datos relacionales.
<b>HTML</b>	<i>HyperText Markup Language</i> . Lenguaje de marcado para elaboración de páginas web.
<b>CSS3</b>	<i>Cascading Style Sheet 3</i> . Lenguaje usado para definir y crear la presentación de un documento estructurado.
<b>JavaScript</b>	Lenguaje de programación interpretado, utilizado principalmente en el lado del cliente para el desarrollo web dinámico.
<b>Jquery</b>	Biblioteca de JavaScript.
<b>SQLite</b>	SQLite es una biblioteca escrita en lenguaje <a href="#">C</a> que implementa un sistema de gestión de bases de datos transaccionales <a href="#">SQL</a> auto-contenido, sin servidor y sin configuración.



## Anexos

---

### A Acta de reunión con el cliente

# ACTA DE REUNIÓN

---

**ASISTENTES:** Daniel Ramos Trejo (jefe de la empresa desarrolladora)  
Bonifacio Fernández (dueño del restaurante)

**LUGAR:** Restaurante El Vidueño

**FECHA:** 18/10/2016

#### ASUNTOS

En la presente reunión hemos hablado de varios temas, principalmente sobre la funcionalidad de la aplicación que el cliente, en este caso, solicita que tenga para la gestión de su restaurante. Entre los temas tratados cabe destacar que el cliente que va al restaurante pueda realizar su comanda desde la mesa sin necesidad de que el camarero intervenga, que el camarero reciba las comandas en su terminal y que el gerente pueda realizar un control de los productos del restaurante entre otras funcionalidades como generación de estadísticas con las opiniones de los clientes del restaurante. El nombre de la aplicación se quedó en que se llamaría “*Note The Commands*”, anote las comandas en inglés. La duración de la reunión fue de una hora y media aproximadamente con un gran interés por parte del cliente para que nuestra empresa desarrolle el proyecto.

#### TEMAS TRATADOS

1. Objetivo
  - Desarrollo de una aplicación web para anotar las comandas de cada una de las mesas del restaurante.
2. Obtención de Requisitos
  - a) Funcionales:
    - El cliente podrá crear su comanda desde la mesa habiéndose registrado o no previamente en el sistema (el cliente que se registre podrá obtener beneficios en forma de descuentos, por ejemplo).
    - El cliente le enviará la comanda para que sea procesada a través de la aplicación.
    - El sistema permitirá loguearse tanto a clientes, como al camarero y al gerente del restaurante.
    - El cliente solicitará la cuenta a través de la aplicación cuando haya decidido que ha terminado.
    - El cliente rellenara un breve cuestionario de forma anónima para que el gerente pueda sacar sus propias conclusiones.

- El camarero recibirá en su terminal las comandas que los clientes han solicitado y las procesará.
  - El camarero marcará como servido aquellos productos que ha servido ya a la mesa correspondiente.
  - El gerente podrá realizar un control del stock del restaurante para prevenir posible falta de productos.<sup>1</sup>
  - El gerente también podrá realizar los cuestionarios que los clientes cuando podan la cuenta realizaran de forma anónima sobre el trato recibido por el camarero, el estado de la comida o cual ha sido el plato que más le ha gustado.
  - El gerente podrá crear los menús de la semana o podrá realizar un menú especial previo acuerdo con un cliente que ha reservado una mesa.
- b) No funcionales
- La aplicación debe tener una interfaz gráfica sencilla e intuitiva.
  - En caso de introducir mal los datos deberá mostrar mensajes de error.
  - Principalmente se utilizará la aplicación en el navegador de Google Chrome.
  - Se solicitará usuario y contraseña para acceder al sistema a los camareros y al gerente, pero no es obligatorio para el cliente del restaurante.
3. Posibles mejoras
- El cliente pueda realizar una reserva de una mesa para una hora determinada para un número determinado de personas y con un menú que el mismo elija.
  - El gerente podrá colgar noticias sobre los nuevos menús de la semana cual ha sido el plato estrella.
  - El cliente podrá reportar una incidencia por ejemplo del estado del aire acondicionado, si una silla está rota, si hay una bombilla fundida, etc.
  - En caso d que el cliente se haya registrado en la aplicación y haya seleccionado que desea recibir noticias, recibirá las noticias que el gerente vaya haciendo, por ejemplo, si es la semana de la gastronomía, si es la semana del cocido, si es la semana de los callos madrileños, etc...
  - El cliente podrá solicitar una hoja de reclamaciones por algo que le haya pasado en el restaurante.

Nombre del autor del acta: Daniel Ramos Trejo  
 Fecha de edición del acta: 19/10/2016  
 Revisado y modificado: Ana González Marcos  
 Fecha de revisión: 3/11/2016

## **B Captura de requisitos**

Una vez realizado el análisis del estado del arte y evaluado las necesidades del cliente en las distintas reuniones que hemos tenido, se van a enumerar los requisitos funcionales y no funcionales de nuestro proyecto.

- **Requisitos funcionales del desarrollador del sistema:**

**RF.1** Se permitirá que el desarrollador, en este caso el administrador global de la aplicación, registre mediante un conjunto de datos o dé de baja restaurantes en el sistema.

**RF.2** El administrador de la aplicación dará de alta a los empleados en un restaurante con una jerarquía (Gerente, camarero, cocinero e incluso cliente). Los empleados deben estar registrados previamente como usuarios en el sistema.

**RF.3** Tendrá la posibilidad de visualizar en formato tabla todos los productos registrados en todos los restaurantes del sistema, también podrá ver todos los usuarios registrados en el sistema.

- **Requisitos funcionales comunes a los usuarios del sistema:**

**RF.4** El sistema permitirá darse de alta a usuarios mediante un conjunto de datos personales suministrados por el cliente. También permitirá darse de baja a un usuario que lo desee.

**RF.5** El sistema permitirá a un usuario registrado darse de alta en un mismo restaurante como gerente, camarero cocinero o cliente del mismo, pudiendo tener las funcionalidades de toda la jerarquía.

**RF.6** Se permitirá a su vez loguearse a usuarios registrados con puntos de entrada diferentes dependiendo de su Jerarquía.

**RF.7** Cada usuario registrado podrá seleccionar un restaurante de una lista de restaurantes registrados.

**RF.8** El sistema permitirá a los usuarios visualizar su perfil y modificar sus datos personales.

**RF.9** El sistema mostrará la ubicación exacta del restaurante mediante un pequeño mapa de *Google*.

**RF.10** El sistema dispondrá de un blog único donde los usuarios podrán ver y comentar las noticias que el gerente publique.

**RF.11** Se dispondrá de un correo electrónico propio donde los usuarios registrados en el sistema podrán intercambiar información mediante emails.

**RF.12** El sistema tendrá una bolsa de empleo para cada restaurante que permitirá a un usuario registrado que esté interesado en trabajar subir su curriculum vitae. Solo el gerente podrá visualizarlos.

- **Requisitos funcionales del cliente:**

**RF.13** Cada cliente de un restaurante podrá seleccionar una mesa de entre las disponibles y crear su propia comanda sin necesidad de la intervención del camarero.

**RF.14** El cliente de una mesa podrá añadir y eliminar productos de su comanda (siempre y cuando no lo haya servido por el camarero).

**RF.15** El sistema permitirá cerrar una comanda y mandar una solicitud de aviso para que el camarero le traiga la cuenta de su comanda.

**RF.16** Una vez solicitada la cuenta, el cliente podrá realizar un pequeño cuestionario de forma anónima para valorar ciertos aspectos del restaurante.

**RF.17** Se dispondrá de un sistema chat para todos los clientes registrados en el sistema para que puedan comunicarse en tiempo real.

**RF.18** El sistema permitirá a los usuarios asociados a un restaurante ver mediante gráfico y de forma estadística cuales son los productos más consumidos en un rango de tiempo.

**RF.19** Se dispondrá de un gestor de incidencias donde el cliente de un restaurante podrá crear y enviar una incidencia o bien al camarero, si la categoría de la incidencia es de comida, o al gerente para el resto de incidencias.

**RF.20** El sistema ofrecerá la posibilidad de realizar pedidos online por parte de un cliente registrado en un restaurante que tenga en su lista de restaurantes asociados.

**RF.21** El sistema dispondrá de una sección de reservas para un determinado restaurante.

**RF.22** El cliente podrá visualizar los menús que el gerente realice para el restaurante.

- **Requisitos funcionales del camarero:**

**RF.23** El camarero podrá visualizar una comanda que está activa para poder servir los productos y así marcarlos como servidos.

**RF.24** El sistema permitirá imprimir un recibo con los productos de una comanda cuando el cliente lo solicite.

- **Requisitos funcionales del cocinero:**

**RF.25** El cocinero recibirá los productos que tenga que cocinar que, cuando los termine de cocinar, mandará un aviso al camarero para que este lo pueda servir y marcarlo como servido.

- **Requisitos funcionales del gerente:**

**RF.26** El gerente podrá publicar noticias en el blog del restaurante al que esté asociado.

**RF.27** Se le permitirá crear cuestionarios para el restaurante al que este asociado para que los clientes de forma anónima puedan aportar información relevante.

**RF.28** Se le proporcionará una sección de estadísticas donde el gerente podrá ver las votaciones de los clientes de forma gráfica y en un rango de fechas que quiera para así poder tomar sus propias decisiones de negocio.

**RF.29** El sistema permitirá que el gerente registre productos nuevos en el restaurante.

**RF.30** Dispondrá de un stock donde el gerente podrá llevar un control de los productos del restaurante.

**RF.31** El gerente podrá crear, eliminar un menú para el restaurante.

**RF.32** Permitirá que el gerente añada productos para un determinado menú del restaurante.

- **Requisitos no funcionales:**

**RF.1** La aplicación web dispondrá de una interfaz gráfica sencilla e intuitiva para la iteración de los usuarios independientemente de su categoría sea sencilla.

**RF.2** La interfaz mostrará mensajes de error mediante pop-ups si se realiza una acción errónea o los datos introducidos son incorrectos.

**RF.3** El idioma de la aplicación será castellano por defecto.

**RF.4** Se podrá utilizar en un navegador Chrome principalmente.

**RF.5** Para el acceso al sistema se solicitará un usuario y una contraseña.

**RF.6** La aplicación web desarrollada dispondrá de un manual de usuario [ANEXO F].

## **C Manual de instalación del entorno**

A continuación, mostraremos los pasos seguidos para la instalación de las herramientas de RoR en nuestra computadora con Sistema Operativo Linux/Ubuntu. Toda la instalación la hemos realizado desde la terminal, por tanto, indicaremos los comandos introducidos para la instalación:

1. Primero se instalarán los paquetes requeridos por Ruby Version Manager (RVM):  
\$ sudo apt-get install -y git-core subversion
2. Instalación de la firma:  
\$ gpg2 --keyserver hkp://keys.gnupg.net --recv-keys  
409B6B1796C275462A1703113804BB82D39DC0E3
3. Instalación rápida de RVM:  
\$ \curl -sSL https://get.rvm.io | bash -s stable --ruby  
\$ rvm install 1.9.3
4. Comprobación de la correcta configuración:  
\$ type rvm | head -n -1
5. Instalación de Ruby:  
\$ rvm install 2.3.1
6. Instalación de Rails:  
\$ gem install rails --version 5.0.0
7. Comprobación de las versiones instaladas tanto de Ruby como de RVM:  
\$ ruby -v  
\$ rvm -v
8. Instalación de la librería XML:  
\$ sudo apt-get install libxml2-dev
9. Instalación de las gemas de Ruby on Rails:  
\$ sudo gem install bundle
10. Instalación node JS:  
\$ sudo apt-get install nodejs

Finalmente, instalaremos el editor de textos utilizado para el desarrollo de la aplicación *'sublime text'*. Los pasos seguidos son los siguientes:

```
$ sudo add-apt-repository ppa:webupd8team/sublime-text-3  
$ sudo apt-get update  
$ sudo apt-get install sublime-text-installer
```

## D Estructura del proyecto

El proyecto se llama *NoteTheCommand* y está desglosado en las siguientes subcarpetas:

- **app/** → Contiene los controllers, models, views, helpers, mailers y assets para tu aplicación. Te centrarás en esta carpeta por el resto de esta guía.
- **config/** → Configura las reglas de ejecución de la aplicación, rutas, base de datos y más. Este tema es cubierto en mayor detalle en [Configuring Rails Applications](#).
- **config.ru** → Configuración Rack para servidores basados en Rack usados para iniciar la aplicación.
- **db/** → Contiene el esquema actual de tu base de datos, así como las migraciones de la base de datos.
- **doc/** → Documentación detallada de tu aplicación.
- **Gemfile** **Gemfile.lock** → Estos archivos te permiten especificar qué dependencias de gemas son necesitadas para tu aplicación Rails. Estos archivos son usados por la gema Bundler, ver [Sitio web de Bundler](#).
- **lib/** → Módulos extendidos para tu aplicación.
- **log/** → Archivos de Log de tu aplicación.
- **public/** → La única carpeta vista por el mundo tal como es. Contiene los archivos estáticos y assets compilados.
- **Rakefile** → Este archivo localiza y carga tareas que pueden ser ejecutadas desde la línea de comandos. Las listas de tareas son definidas a través de los componentes de Rails. En vez de cambiar el Rakefile, deberías agregar tus propias tareas, añadiendo archivos al directorio lib/tasks de tu aplicación.
- **README.rdoc** → Este es un breve manual de instrucciones para tu aplicación. Deberías editar este archivo para comunicar a otros lo que tu aplicación hace, cómo configurarla y demás.
- **script/** → Contiene el script de Rails que inicia tu aplicación y contiene otros scripts usados para deployar o correr tu aplicación.
- **test/** → Pruebas unitarias, fixtures y otras pruebas. Éstos son cubiertos en [Testing Rails Applications](#).
- **tmp/** → Archivos temporales (como caché, PID y archivos de sesiones).
- **vendor/** → Lugar para código de terceros. En una típica aplicación Rails, ésta incluye librerías y plugins.

## E Costes de la aplicación

Para la administración del proyecto se ha realizado un presupuesto con todos los gastos y recursos utilizados para el desarrollo de nuestra aplicación web utilizando la herramienta *OpenProj*. Los recursos empleados son o bien materiales o bien de trabajo profesional:

### Materiales empleados

- Red eléctrica → 0.20 €/hora.
- Portátil ASUS F541U → 750€.
- Ratón Microsoft → 30€.
- Gastos de impresión → 45€.

### Personal cualificado

- Junior en Desarrollo Web y QA → 50€/hora -- 60€/hora extra.

### Fecha de inicio y fin de proyecto previsto

17/10/2016 – 31/05/2017

A continuación, mostraremos la gráfica de Gantt donde se expone el tiempo de dedicación previsto para las diferentes tareas a lo largo del tiempo total del proyecto.

	🕒	Nombre	Duración
1	🕒	Captura de requisitos	0,75 days?
2	🕒	Diseño del diagramas del proyecto	8,5 days?
3	🕒	Instalación herramientas de trabajo	1,438 days?
4	🕒	Diseño y programacion de la Base de Datos	21,125 days?
5	🕒	Programacion de la aplicacion en Ruby on Rails	83,312 days?
6	🕒	Realizacion de las pruebas para la QA	16 days?
7	🕒	Generacion de la documentacion de la aplicacion software	25 days?
8	🕒	Presentacion del producto al cliente	1 day?

Figura 0-1: Tareas asignadas y duración

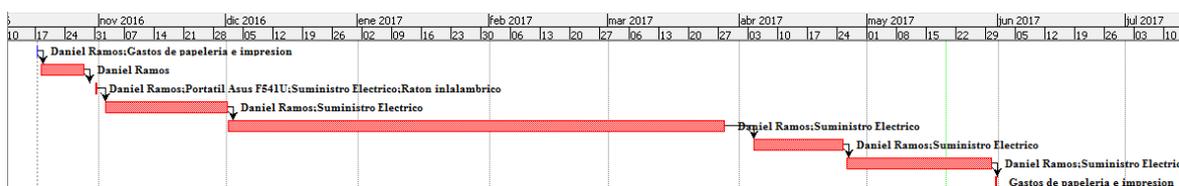


Figura 0-2: diagrama de Gantt de las tareas

Tras realizar el diagrama de Gantt con las fechas de las tareas y los recursos empleados en cada una de las tareas hemos obtenido un total de horas dedicadas al

proyecto de 162,75 días, a una media de 4 horas. A continuación, desglosaremos el total de las horas:

### **Desglose de las horas dedicadas en el desarrollo del proyecto sin demoras**

- Duración total del proyecto en días (sin tener en cuenta los retrasos en algunas tareas) → 157,125 días.
- Horas dedicadas al día → 4 horas.
- Total horas dedicadas al proyecto → 628,5 horas.

### **Desglose de las horas dedicadas en el desarrollo del proyecto con demoras**

- Duración total del proyecto en días (teniendo en cuenta los retrasos en algunas tareas) → 162,75 días.
- Horas dedicadas al día → 4 horas.
- Total horas dedicadas al proyecto → 651 horas.

Hay que tener en cuenta que el proyecto se ha demorado 5,625 días (162,75 días - 157,125 días). Por tanto, los gastos del proyecto son:

### **Desglose de los gastos del proyecto**

- Luz → 651 horas \* 0,20 €/hora = 130,20 €.
- Portátil → 750 €.
- Ratón Microsoft → 30 €.
- Gastos de impresión → 45 €.
- Desarrollador Junior:
  - Coste sin demoras:  
157,75 días → 631 horas.  
50 €/hora \* 631 horas = 31550 €.
  - Coste demoras (horas extra):  
162,75 días - 157,125 días = 5,625 días. → 22,5 horas  
60 €/hora \* 22,5 horas = 1350 €.

Con estos datos se va a calcular el coste total del proyecto sin IVA y lo vamos a desglosar a continuación:

### **Coste de la aplicación sin IVA**

- Total sin IVA → 130,20 € (luz) + 750 € (portátil) + 30 € (ratón) + 45 € (impresión) + 31550 € (desarrollador Junior) + 1350 € (desarrollador Junior horas extra) = 33855,20 €.

Finalmente aplicándole el 21% de IVA nos quedaría un coste total desglosado de la siguiente manera:

### **Coste de la aplicación con IVA**

- Total con 21% IVA → 7109,60 € (21%) + 33855,20 € = **40964,80 €**.

## F Manual de Usuario de NoteTheCommands

### Aspectos generales de las ventanas.

Cada usuario dependiendo de la jerarquía que tenga en la aplicación tendrá distinta funcionalidad, por ello, como mencionamos en el apartado de pruebas, utilizaremos un borde de color distintivo para diferenciarlos visualmente.

**Administrador**      **Cliente**      **Gerente**      **Camarero**      **Cocinero**

En caso de que la ventana sea accedida por cualquier usuario de la jerarquía no se visualizará con borde de color.

### Logo de la aplicación.



Figura 0-3: Logo NoteTheCommands

### Ventana de inicio y bienvenida a todos los usuarios.

Al acceder a la web, se muestra la pantalla inicial donde nos da la bienvenida al sistema y nos permite, o bien loguearnos si somos usuarios registrados en el sistema, o bien registrarnos como nuevos usuarios, o bien acceder sin registro al sistema. Este último método de acceso a la aplicación, llamado ‘usuario invitado’, no nos permitirá obtener ventajas en forma de descuento final a la hora de realizar el pago de una comanda o de un pedido.

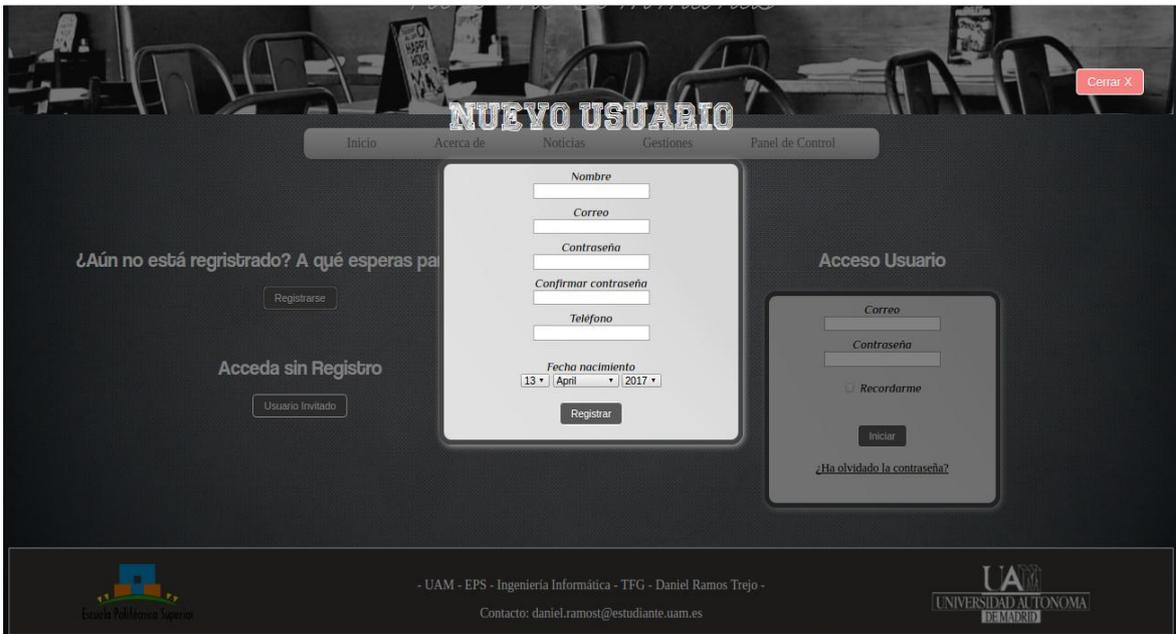


Figura 0-4: Ventana de inicio bienvenida

En la Figura 0-6 podemos observar la presencia de una barra de menú desplegable donde una vez logueados veremos las distintas tareas que podemos realizar en la aplicación dependiendo de la jerarquía del usuario con el que nos hayamos logueado.

***Dar de alta un nuevo usuario en el sistema.***

Para dar de alta un nuevo usuario habría que pulsar el botón ‘*Registrarse*’, y en la nueva ventana rellenar el formulario con los datos del nuevo usuario.



**Figura 0-5: Ventana para dar de alta a un usuario en el sistema**

***Acceso a un usuario registrado.***

Cualquier usuario que quiera acceder al sistema deberá rellenar el formulario de acceso con sus datos que son el correo electrónico y la contraseña.



**Figura 0-6: Acceso usuario registrado ‘login’**

## ¿He olvidado mi contraseña?

En caso de no acordarte de la contraseña el sistema nos permitirá cambiarla redireccionándonos a una nueva ventana solicitándonos el correo electrónico, donde nos enviará los pasos a seguir para restaurar la contraseña.



Figura 0-7: Ventana ¿Ha olvidado la contraseña?

## Ayuda en cualquiera de las ventanas de la aplicación

Además de este manual de usuario, dispondremos de un botón de ayuda en el que al pasar el puntero del ratón o pulsar sobre él nos mostrará información acerca de la ventana en la que nos encontremos como podemos observar en la Figura 0-8:



Figura 0-8: Ventana de ejemplo de ayuda

## Perfil de usuario.

Si pulsamos en la barra de tareas en 'Panel de Control → Mi Perfil' podremos visualizar el perfil de usuario que esta logueado, pudiendo modificarlo si así lo desea.



**Figura 0-9: Ventana de perfil de usuario**

## ADMINISTRADOR

*Pantalla de inicio del administrador logueado.*

A continuación, mostraremos la pantalla que le aparecería al administrador nada más loguearse en el sistema. En ella dispondremos del botón 'Dar de Baja' para dar de baja un empleado de un restaurante, una barra con las tareas que puede realizar el administrador y una tabla con los empleados registrados en el sistema el cual dispone de un buscador para filtrar los empleados por el nombre del restaurante al que pertenecen.



**Figura 0-10: Ventana de inicio para el administrador**

### Barra de tareas para el administrador.

A continuación, mostraremos las distintas funcionalidades que presenta la barra de tareas del administrador del sistema.



Figura 0-11: Barra de tareas del administrador

### Tareas del administrador.

Si pulsamos en la barra de tareas en 'Inicio → Tareas' podremos visualizar las tareas que el administrador puede realizar sobre la aplicación. Estas tareas son las mismas que las que aparecen en la barra de tareas de la Figura 0-11. Cada una de estas tareas funcionan como un botón, por lo que si pulsamos cualquiera de ellas redirecciona a la ventana correspondiente. Algunas de estas funcionalidades las hemos explicado anteriormente como en la Figura 0-5 para el registro de un empleado en el sistema.



Figura 0-12: Tareas del administrador

### Registrar un nuevo restaurante.

Si pulsamos en la barra de tareas en 'Gestión → Dar de alta Restaurante' podremos pulsar en el botón 'Registrar Restaurante' y nos mostrará un formulario que deberemos rellenar con los datos del nuevo restaurante que queramos dar de alta. Como

explicamos en capítulos anteriores, las coordenadas del restaurante se calcularán y se almacenarán automáticamente con el registro del restaurante mediante el uso de una gema de RoR llamada *geocoder*.



Figura 0-13: Ventana para dar de alta un nuevo restaurante

### Dar de alta empleado

Si pulsamos en la barra de tareas en 'Gestión → Dar de alta Empleado' podremos pulsar el botón 'Añadir Empleado' para registrar un nuevo empleado en un restaurante determinado.

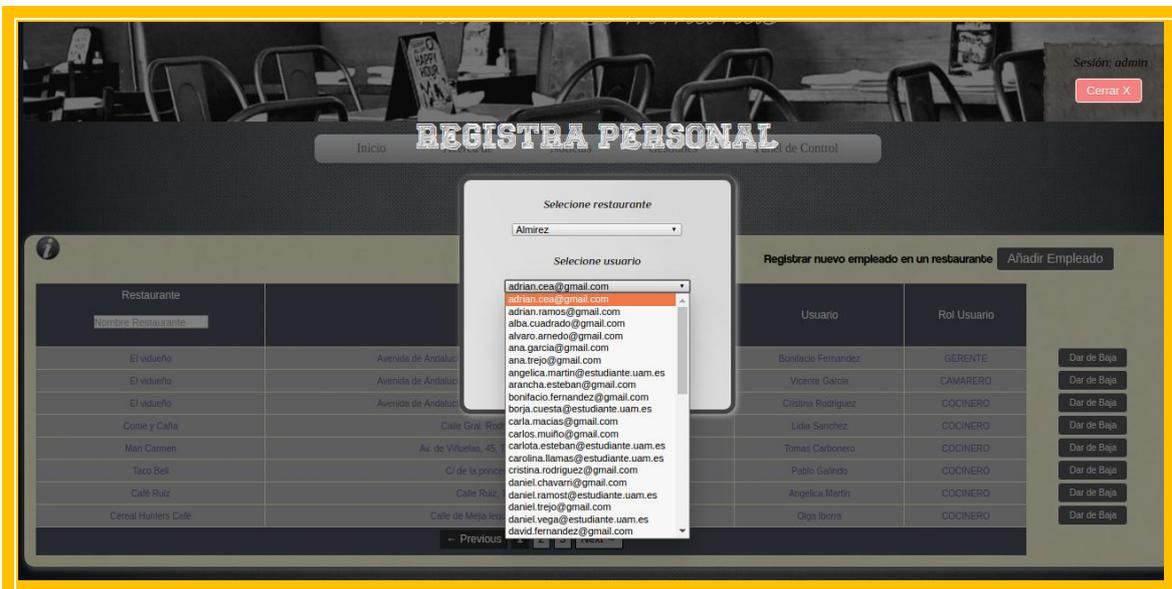


Figura 0-14: Ventana para dar de alta un nuevo empleado

### Visualización de los restaurantes registrados.

Si pulsamos en la barra de tareas en ‘*Gestión → Restaurantes Registrados*’ podremos visualizar en una tabla todos los restaurantes registrados en el sistema. Dicha tabla dispondrá de un filtro para buscar un restaurante por su nombre.



The screenshot shows a web interface with a navigation bar at the top containing 'Inicio', 'Acerca de', 'Noticias', 'Gestiones', and 'Panel de Control'. Below the navigation bar is a header with the title 'RESTAURANTES REGISTRADOS'. The main content area features a table with the following columns: 'Nombre', 'Correo electrónico', 'Teléfono', 'País', 'Código postal', 'Coord. Latitud', 'Coord. Longitud', 'Horario apertura', and 'Fecha registrado'. A search filter is present in the 'Nombre' column. The table lists 15 restaurants, including KFC Sol, McDonald's, Burger King, El vidueño, Taco Bell, Mari Carmen, Come y Caña, Almiraz, Asador Castilla, Amarras, Café Ruiz, and Casa Juaneca. At the bottom of the table, there are navigation controls: '-- Previous', '1', '2', and 'Next --'.

Nombre	Correo electrónico	Teléfono	País	Código postal	Coord. Latitud	Coord. Longitud	Horario apertura	Fecha registrado
KFC Sol	KFC@kfc.com	915227749	España	28013	40.4165	-3.70459	L-D: 10:00-00:00	12 January 2016 - 13:29:42
McDonald	mcdonalds@mcdonalds.com	913581384	España	28046	40.4802	-3.68708	L-D: 07:00-00:30	13 January 2016 - 13:29:42
Burger King	customerservicespain1@burgerkingcm.com	916587171	España	28700	40.5481	-3.63402	L-D: 07:00-00:30	17 February 2016 - 13:29:42
El vidueño	elvidueno@elvidueno.com	916519761	España	28702	40.5538	-3.62945	L-D: 07:00-00:30	28 February 2016 - 13:29:42
Taco Bell	tacobelli@tacobelli.com	916534361	España	28008	40.425	-3.71238	L-D: 09:00-24:00	14 May 2016 - 13:29:42
Mari Carmen	maricamen@maricamen.com	918042950	España	28760	40.5975	-3.70017	L-D: 12:00-23:30	28 June 2016 - 13:29:42
Come y Caña	comeycaña@comeycaña.com	915535953	España	28003	40.4435	-3.71426	L-D: 12:00-23:30	13 July 2016 - 13:29:42
Almiraz	almiraz@almiraz.com	912291688	España	28100	40.5551	-3.65302	L-D: 12:00-01:00	08 September 2016 - 13:29:42
Asador Castilla	asadorcastilla@asadorcastilla.com	916516011	España	28703	40.5572	-3.62575	L-D: 12:00-18:00, 20:00-24:00	06 October 2016 - 13:29:42
Amarras	amarras@amarras.com	918278398	España	28100	40.5494	-3.65831	L-D: 10:00-24:00	13 November 2016 - 13:29:42
Café Ruiz	cafeeruz@cafeeruz.com	914461232	España	28004	40.4583	-3.70478	L-D: 16:00-02:00	23 December 2016 - 13:29:42
Casa Juaneca	casajuaneca@casajuaneca.com	918435702	España	28770	40.6482	-3.59559	L-D: 8:00-23:00	23 December 2016 - 13:29:42

Figura 0-15: Ventana para visualizar los restaurantes del sistema

### Visualización de los clientes registrados.

Si pulsamos en la barra de tareas en ‘*Gestión → Clientes Registrados*’ podremos visualizar en una tabla todos los clientes registrados en el sistema. Dicha tabla dispondrá de un filtro para buscar el cliente por su nombre.



The screenshot shows a web interface with a navigation bar at the top containing 'Inicio', 'Acerca de', 'Noticias', 'Gestiones', and 'Panel de Control'. Below the navigation bar is a header with the title 'CLIENTES REGISTRADOS'. The main content area features a table with the following columns: 'Nombre', 'IP último inicio', 'IP Actual', 'Correo electrónico', 'Inicios de sesión', 'Teléfono', and 'Fecha de alta'. A search filter is present in the 'Nombre' column. The table lists 10 clients, including admin, Daniel Ramos, Pilar Ezqurra, Vicente Teruel, Erika Luna, Maria Guinea, Daniel Vega, Ana Trejo, Fernando Sanchez, and Carlota Esteban. At the bottom of the table, there are navigation controls: '-- Previous', '1', '2', '3', '4', '5', '6', and 'Next --'.

Nombre	IP último inicio	IP Actual	Correo electrónico	Inicios de sesión	Teléfono	Fecha de alta
admin	127.0.0.1	127.0.0.1	admin@admin.com	4		10 May 2017 a las 17:13:00
Daniel Ramos	127.0.0.1	127.0.0.1	daniel.ramost@estudiante.uam.es	3	630789317	10 May 2017 a las 17:13:00
Pilar Ezqurra	127.0.0.1	127.0.0.1	pilar.ezqurra@gmail.com	1	545632120	10 May 2017 a las 17:13:00
Vicente Teruel	127.0.0.1	127.0.0.1	vicente.teruel@estudiante.uam.es	2	666232232	10 May 2017 a las 17:13:01
Erika Luna	127.0.0.1	127.0.0.1	erika.luna@gmail.com	1	666322147	10 May 2017 a las 17:13:01
Maria Guinea	127.0.0.1	127.0.0.1	maria.guinea@gmail.com	1	663251123	10 May 2017 a las 17:13:01
Daniel Vega	127.0.0.1	127.0.0.1	daniel.vega@estudiante.uam.es	2	666895412	10 May 2017 a las 17:13:01
Ana Trejo	127.0.0.1	127.0.0.1	ana.trejo@gmail.com	1	663336522	10 May 2017 a las 17:13:01
Fernando Sanchez	127.0.0.1	127.0.0.1	fernando.sanchez@estudiante.uam.es	1	652137894	10 May 2017 a las 17:13:01
Carlota Esteban	127.0.0.1	127.0.0.1	carlota.esteban@estudiante.uam.es	1	665221452	10 May 2017 a las 17:13:02

Figura 0-16: Ventana para visualizar los clientes del sistema

## Visualización de los productos registrados.

Si pulsamos en la barra de tareas en ‘Gestión → Productos Registrados’ podremos visualizar en una tabla todos los restaurantes registrados en el sistema. Dicha tabla dispondrá de un filtro para buscar un producto por su nombre.



Nombre	Restaurante	Categoría	Ingredientes	Precio	Fecha de alta	Fecha modificación
<input type="text" value="Nombre del producto"/>						
Pan de trigo	El vidueño	GENERAL	1 cucharada de azúcar. 1 cucharada de levadura seca de panadero. 100 ml agua tibia. 212 g masa madre...	0.9 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Alubias Blancas	El vidueño	CALDOS	500 gramos judías blancas. 150 gramos beicon en un trozo. 1 punta jamón. 150 gramos chorizo. 4 patatas...	4.5 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Croquetas Caseras	El vidueño	RACION	200 g de jamón serrano. 1 cebolleta. 1 diente de ajo. 100 g de harina. 100 g de mantequilla (o 100 ml de a...	6.0 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Gambas blancas a la plancha	El vidueño	RACION	Gambas Blancas. Agua. Hielo y Sal	13.0 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Pan con Jamon Iberico con tomate	El vidueño	RACION	Pan, Jamon Iberico de Bellota de las dehesas Extremeñas, tomate, sal y aceite de oliva virgen extra	12.5 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Anchoas marinadas	El vidueño	PESCADOS	6 anchoas. 1 diente de ajo. 2 rodajas finas de pan. zumo de 1/2 limón. 1 cucharada de aceite. una pizca d...	8.5 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Pimientos asados con ventresca	El vidueño	PESCADOS	pimientos de asar. Aceite de Oliva Virgen Extra. Sal. Pieza de Ventresca	7.5 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Migas Extremeñas	El vidueño	RACION	1 hogaza de pan candeal. 3 lonchas de pancetta. 1 chorizo. 1 cabeza de ajo. 1 vaso (de vino) de aceite. 1 ...	6.5 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Cocido Madrileño	El vidueño	CALDOS	500 gr. de garbanzos. 1/2 repollo. 1/2 kg. de zanahorias pequeñas. 6 patatas medianas. 1/2 kg. de morcil...	3.5 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Callos a la Madrileña	El vidueño	CALDOS	1/2 Kg. de callos 1 Kg. de morro de vaca 100 g de tocino 1 hueso de jamón 2 morcillas 2 chorizos caseros...	11.5 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Sopa de verduras	El vidueño	CALDOS	1 carcasa pollo campero. 2 muslos enteros de pollo campero. 3 cebollas medianas. 3 zanahorias. 1 chirivía...	6.5 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC
Dorada a la española	El vidueño	PESCADOS	6 doradas a la española. 1 vaso pequeño aceite. 2 cabezas ajos. perejil. sal	9.9 €/Unidad	2016-11-13 13:29:42 UTC	2016-11-13 13:29:42 UTC

Figura 0-17: Ventana para visualizar los productos del sistema

## Perfil del usuario.

Si pulsamos en la barra de tareas en ‘Panel de Control → Mi Perfil’ podremos visualizar el perfil del usuario que esta logueado, en este caso el del administrador, pudiendo modificarlo si es necesario. La ventana será similar a la de la Figura 0-9.

## Cerrar sesión del usuario administrador.

Si pulsamos en la barra de tareas en ‘Panel de Control → Cerrar Sesión’ podremos salir de la sesión del administrador.

## CLIENTE

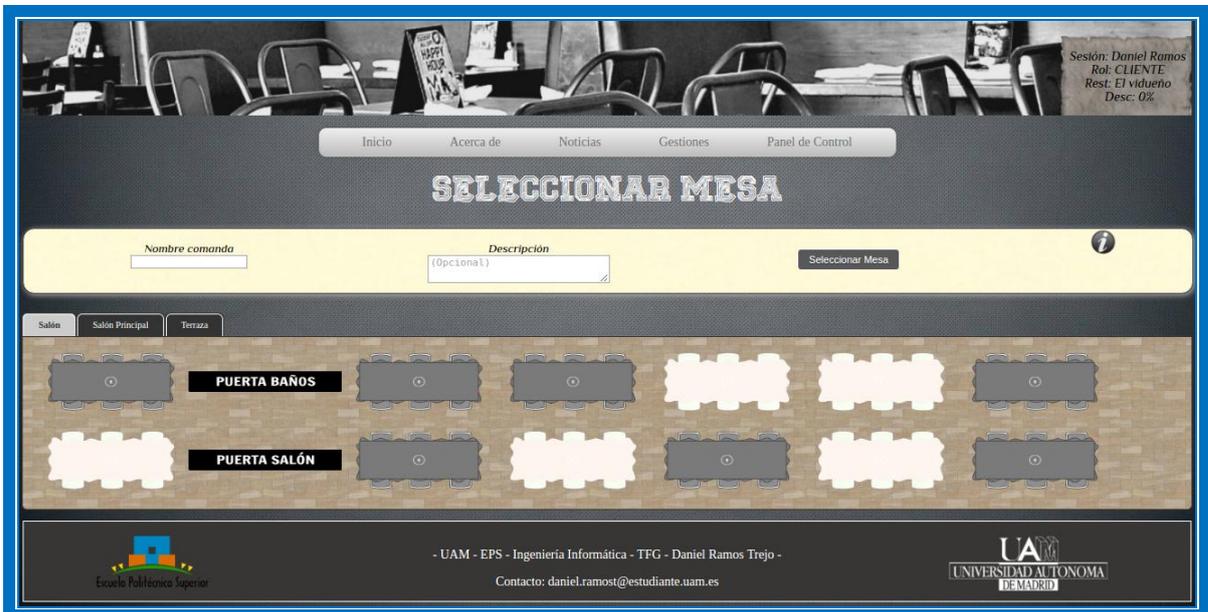
### Pantalla de inicio del cliente logueado.

En esta pantalla se mostrará los restaurantes a los que está dado de alta el usuario. Podrá darse de baja, asociarse a uno nuevo o seleccionar un restaurante de la lista para acceder a pedir la comanda del mismo.



**Figura 0-18: Ventana de inicio del cliente**

A continuación, se selecciona un restaurante donde vamos a realizar las gestiones del manual. Por ejemplo, pulsamos el botón ‘*Seleccionar*’ y elegimos el restaurante *El Vidueño*. A continuación, nos aparece la pantalla para la selección de la mesa (donde las mesas grises son las que están ocupadas, las blancas las disponibles y la roja la seleccionada para nuestra comanda).



**Figura 0-19: Ventana de selección de la mesa**

### ***Barra de tareas para el cliente.***

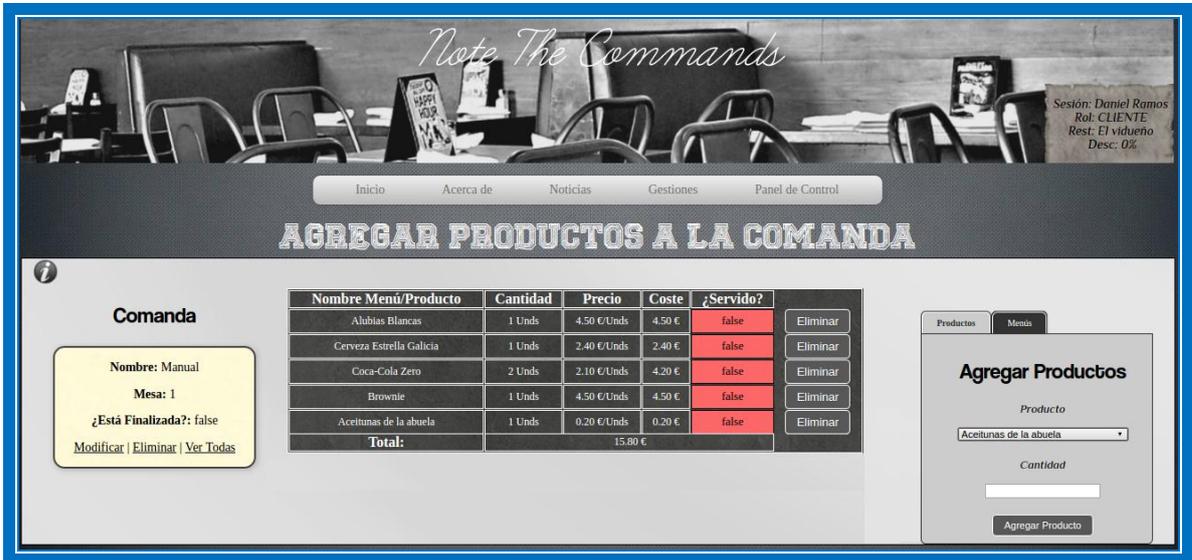
A continuación, mostraremos las distintas funcionalidades que presenta la barra de tareas del cliente en sistema.



**Figura 0-20: Barra de tareas del cliente**

### *Gestión de nuestra comanda.*

En esta pantalla nos aparecerán los productos que vamos añadiendo a nuestra comanda que hemos creado anteriormente. Podremos añadir tanto productos como menús ya creados que podremos visualizar para ver su contenido. Una vez seleccionados los productos, el sistema se lo notificará al camarero (en caso de bebidas como en la Figura 0-66) y al cocinero (en caso de comidas como en la Figura 0-68) para que estos lo gestionen y lo sirvan a la mesa.



**Figura 0-21: Ventana para añadir productos o menús a nuestra comanda**

En esta ventana se puede ver el desglose de los productos y las cantidades de cada uno de ellos. También podremos ver si los productos se han servido a la mesa o no y por supuesto el total de la comanda. Cabe destacar que hasta que no se hayan servido todos los productos a la mesa el cliente no podrá finalizar la comanda y pedir la cuenta.

Una vez se haya realizado el proceso de gestión de la comanda, es decir, que el cocinero haya preparado la comida y el camarero haya servido todos los productos, el

cliente podrá pedir la cuanta de la comanda pulsando el botón ‘Pedir Comanda’. Una vez pulsado se enviará un aviso al camarero para que este le traiga la cuenta. A continuación, se muestra la ventana de solicitud de la cuenta.



Figura 0-22: Ventana para pedir la cuenta de la comanda

### Tareas del cliente.

Si pulsamos en la barra de tareas en ‘Inicio → Tareas’ se visualizarán las tareas que el cliente puede realizar sobre la aplicación. Estas tareas son las mismas que las que aparecen en la barra de tareas de la Figura 0-20. Cada una de estas tareas funcionan como un botón, por lo que si pulsamos cualquiera de ellas nos direcciona a la ventana correspondiente.

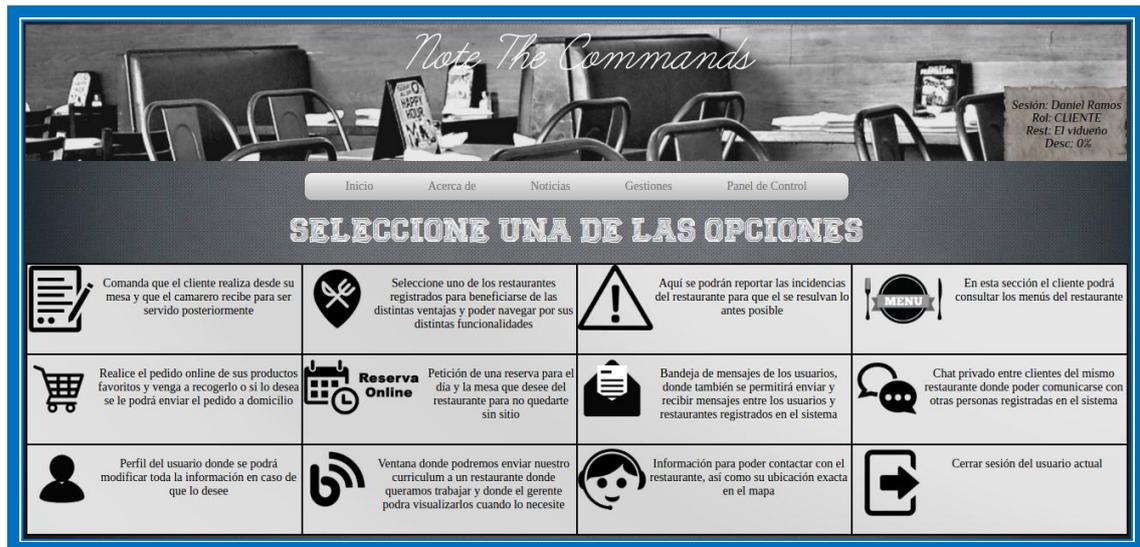


Figura 0-23: Ventana de tareas del cliente

## Comandas del cliente.

Si pulsamos en la barra de tareas en 'Inicio → Mis Comandas' podremos visualizar todas las comandas del cliente ordenadas por si están o no finalizadas. En ella se podrá visualizar el nombre de la comanda, la fecha y el coste de la comanda. Cabe destacar que si la comanda está abierta, aun no mostrará el coste final ya que el cliente puede seguir añadiendo productos o menús y el coste puede variar, por tanto, hasta que no finalice no se mostrará. También podremos visualizar el botón 'editar' donde podremos modificar datos de la comanda como el nombre y también tenemos un botón 'ver comanda' para ver la comanda que tenemos abierta y poder seguir añadiendo productos o finalizarla si procede.

Comanda	Restaurante	Usuario	Mesa	Coste Comanda	Fecha Creación	¿Servido?	¿Está Finalizada?		
Manual	El vidueño	Daniel Ramos	1	0.00 €	11 May 2017 -- 08:26:59	false	false	Editar	Ver Comanda
Cumple mama	El vidueño	Daniel Ramos	1	179.95 €	15 February 2017 -- 13:29:42	true	true		
Buenas noticias	El vidueño	Daniel Ramos	8	59.30 €	08 January 2017 -- 13:29:42	true	true		
Comida entre amigos	El vidueño	Daniel Ramos	10	82.20 €	17 February 2017 -- 13:29:42	true	true		

Figura 0-24: Ventana de las comandas del cliente

## Localización del restaurante.

Si pulsamos en la barra de tareas en 'Acerca de → Donde estamos' podremos visualizar la localización del restaurante, en el que estamos actualmente logueados, mediante el uso de una gema en RoR que se llama *gmaps4rails*. Esta gema genera un *google maps* dentro de la propia aplicación mostrando la ubicación exacta del restaurante, utilizando las coordenadas previamente almacenadas en el registro.

**Localización Establecimiento**

**Información del Restaurante**

Nombre: El vidueño  
Teléfono: 916519761  
Dirección: Avenida de Andalucía, 2  
Ciudad: San Sebastian de los Reyes, Madrid  
País: España  
Código Postal: 28702  
Email: elviduenio@elviduenio.com  
Horario de Apertura: L-D: 07:00-00:30  
Latitud: 40.5538  
Longitud: -3.62945

Figura 0-25: Ventana de localización del restaurante

## Trabajar en el restaurante.

Si pulsamos en la barra de tareas en 'Acerca de → ¿Quieres trabajar con nosotros?' podremos subir nuestro curriculum al restaurante en el que estamos actualmente.

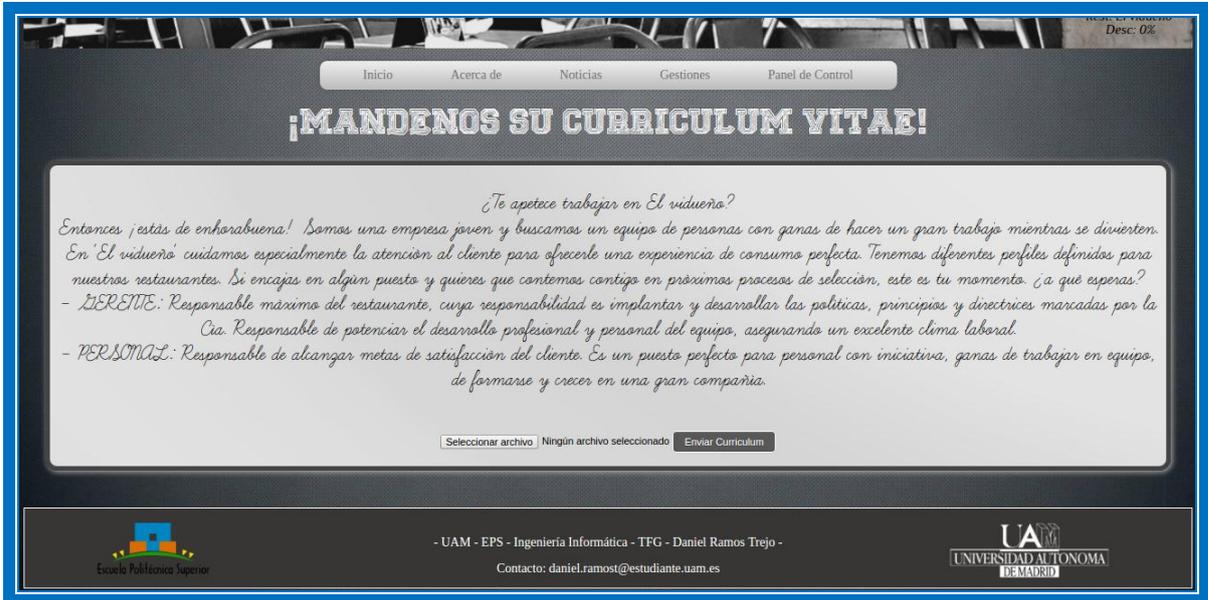


Figura 0-26: Ventana para mandar el curriculum

## Blog del restaurante.

Si pulsamos en la barra de tareas en 'Noticias → Blog' podremos visualizar todas las noticias del blog del restaurante que se hayan seleccionado previamente. En esta ventana se podrán ver todas las noticias del restaurante y comentar cada una de ellas, así como compartir opiniones entre los usuarios.

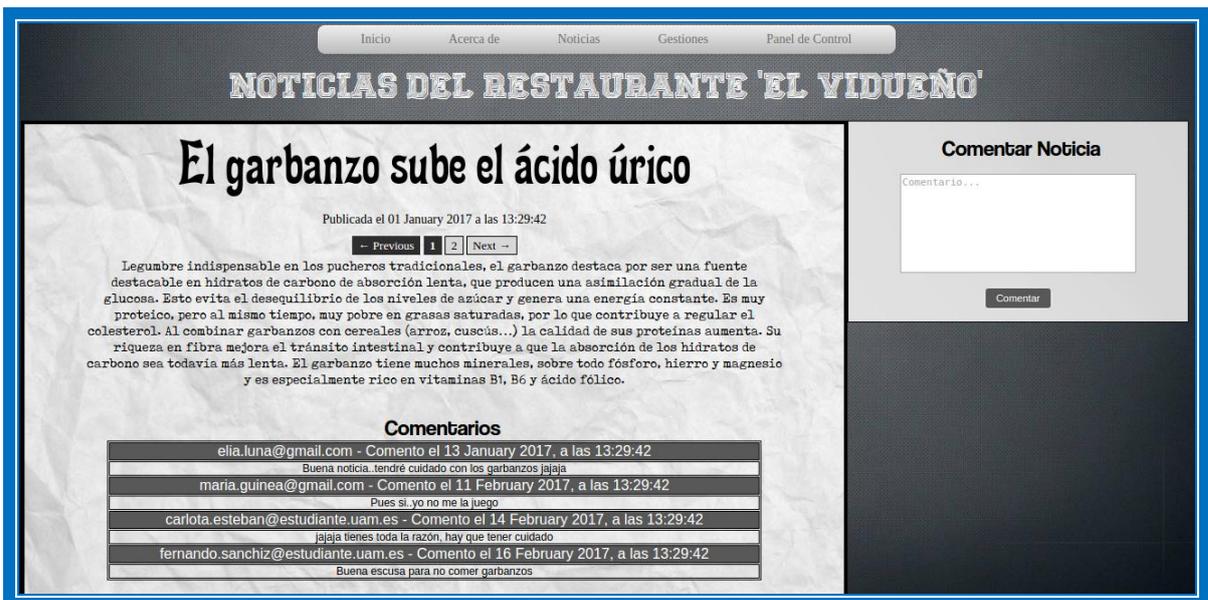


Figura 0-27: Ventana de los blogs del restaurante

### Realizar una reserva en el restaurante por el cliente.

Si pulsamos en la barra de tareas en 'Gestiones → Realizar Reserva' podremos visualizar la ventana de reservas que el cliente deberá rellenar como en la Figura 0-28.

Nombre reserva  
Daniel Chavari

Teléfono de contacto  
633525652

Número personas  
12

Día reserva  
01/07/2017

Hora reserva  
20:00

Menú  
Ninguno

Realizar Reserva

Figura 0-28: Ventana para realizar una reserva

### Pedido online.

Si pulsamos en la barra de tareas en 'Gestiones → Pedido Online' podremos visualizar los pedidos online que el cliente tiene en el restaurante desde que se dio de alta en el sistema. En caso de que la comanda no esté finalizada podremos editar la comanda o agregar productos. También podremos crear un nuevo pulsando el botón 'Crear Pedido'.

Sesión: Daniel Ramos  
Rol: CLIENTE  
Rest: El vidueño  
Desc: 0%

Crear nuevo pedido    Crear Pedido

Nombre del Pedido	Teléfono	Día Recogida	Hora Recogida	Restaurante	Coste Pedido	¿Está Finalizada?	¿Está Entregado?
Ana Garcia	632595987	28 July 2017	13:30:00	El vidueño	192.7 €	false	false
Pepe Alonso	699696969	02 December 2016	13:30:00	El vidueño	100.3 €	true	true
Olga Iborra	650254896	26 December 2016	21:30:00	El vidueño	168.4 €	true	true
Borja Cuesta	758986525	31 December 2016	20:30:00	El vidueño	99.5 €	true	true
Almudena Alonso	698545478	02 January 2017	13:30:00	El vidueño	55.7 €	true	true

Editar    Agregar Productos

Figura 0-29: Ventana de pedidos online

## Menú del restaurante.

Si pulsamos en la barra de tareas en ‘Gestiones → Menú’ podremos visualizar los menús del restaurante y cuales están o no disponibles:

Nombre	Tipo	Precio	Descripción	Restaurante	Creado por:	¿Disponible?
Gran Menú	ESPECIAL	€30.00	Menú muy elaborado	El vidueño	Daniel Ramos	true
Menú Comuniones	ESPECIAL	€30.00	Menú comuniones	El vidueño	Daniel Ramos	true
Menú Especial	ESPECIAL	€23.50	Menú de la casa	El vidueño	Daniel Ramos	true
Menú Messi	ESPECIAL	€100.00	Menú cumpleaños Messi	El vidueño	Daniel Ramos	false
Menú Navidad nº1	ESPECIAL	€35.00	Menú de Navidad 2017	El vidueño	Daniel Ramos	false
Menú Navidad nº2	ESPECIAL	€35.00	Menú de Navidad 2017	El vidueño	Daniel Ramos	false

Figura 0-30: Ventana de los menús del restaurante

Si pulsamos el botón ‘Ver Menú’ podremos visualizar los productos del menú y ver si nos gusta o no.

**Menú**

Nombre: Gran Menú  
Tipo Menú: ESPECIAL  
Precio: 30.0 €  
Descripción: Menú muy elaborado

**Gran Menú**  
Entrante  
*Dambas blancas a la plancha*  
(2 gambas blancas, Ajop, Haró y Sal)

Primer Plato  
*Callos a la Madrileña*  
(1/2 Kg. de callos, 1 Kg. de morro de vaca 100 y de ternera 1, Hueso de puerro 2, morrillas 2, chorizos caseros 1, caldero de agua 2, cebolla, Cebolla de verde Puyol, Laurel, Pasero, Pimentón, Pimentón blanco molido, Sal)

Segundo Plato  
*Cordero lechal asado*  
(2 Cuartos de cordero lechal, Manteca de cerdo, Ajop, Sal, Mostaza con-3 de ajop, vino blanco de val grande y vino carbonado de verano)

Postre  
*Brauño*  
(250 gramos de mantiguillo sin sal - 125 gramos de harina de repostería - 225 gramos de azúcar - 150 gramos de azúcar para forar - 4 Huevos - 5 gramos de canela molida - 5 gramos de especia aromática o un poco de esencia de vainilla)

Bebida  
*Vno tinto de la Rioja*  
(1 cop. local)

Precio: 30,0 €

¿Te gustan los Menús?  
Menús elaborados por los mejores cocineros, con los productos mas exclusivos del mercado que hará que siempre tengas ganas de volver aunque recuerdas. El placer de los banquetes debe medirse no por la abundancia de los manjares, sino por la reunión de los amigos y por su conversación.

Figura 0-31: Ventana con el desglose del menú seleccionado

## Incidencias del restaurante.

Si pulsamos en la barra de tareas en ‘Gestiones → Crear incidencia’ podremos crear una nueva incidencia para el restaurante en el que nos encontramos. Si creamos una incidencia con el asunto comida, esta incidencia irá directamente al camarero para que la

resuelva lo antes posible, el resto irán al gerente del restaurante para que las solucione cuando pueda.

Inicio    Acerca de    Noticias    Gestiones    Panel de Control

## NUEVA INCIDENCIA

**Categoría**  
COMIDA

**Prioridad**  
MUY BAJA

**Asunto**  
Hay un pelo en el plato de la mesa 2

**Descripción**  
He encontrado un pelo en el plato de la mesa 2 y me da mucho asco, queiro que me traigan un nuevo plato

Crear Incidencia

Figura 0-32: Ventana de creación de una incidencia en el restaurante

### Estadísticas del restaurante.

Si pulsamos en la barra de tareas en 'Gestiones → Estadísticas' podremos visualizar las distintas estadísticas disponibles para el cliente.

Inicio    Acerca de    Noticias    Gestiones    Panel de Control

## SELECCIONE UNA DE LAS SIGUIENTES ESTADÍSTICAS

En esta opción el usuario del restaurante al que está registrado podrá visualizar los productos más consumidos en un rango de fechas. Dicha operación se calculará sumando todos los productos del restaurante y mostrará el porcentaje de cada producto sobre el total. Todos los datos serán explícitos al detalle.

Sesión: Daniel Ramos  
Rol: CLIENTE  
Rest: El vidueño  
Desc: 0%

Escuela Politécnica Superior    - UAM - EPS - Ingeniería Informática - TFG - Daniel Ramos Trejo -    UNIVERSIDAD AUTÓNOMA DE MADRID

Contacto: daniel.ramos@estudiante.uam.es

Figura 0-33: Ventana de estadísticas del restaurante

En este caso el cliente solo podrá ver las estadísticas de los productos más vendidos del restaurante para hacerse una idea de que es lo más pedido. La selección la podrá realizar en un rango de fechas y seleccionar una categoría de producto (ración, entrante,

bebida, pescado, etc.). A la derecha tenemos una tabla con los precios de los productos para que el cliente se haga una idea de lo que cuestan.



Figura 0-34: Ventana de estadísticas de todas las categorías

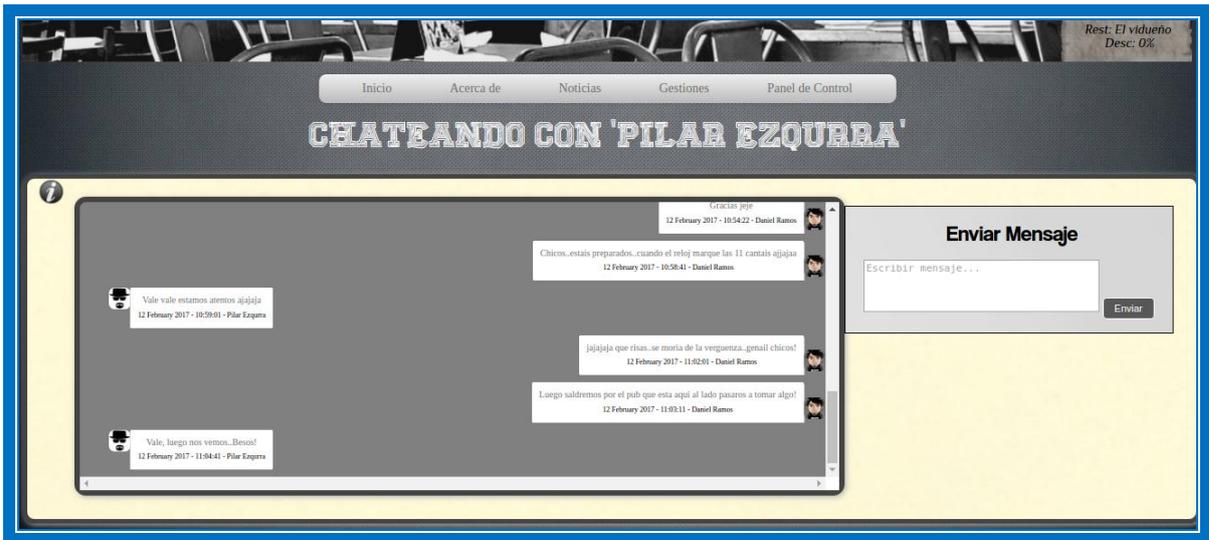
### Chat usuario.

Si pulsamos en la barra de tareas en ‘Panel de Control → Chat’ podremos visualizar la pantalla de chat del sistema. En ella se muestra las conversaciones abiertas y los chats disponibles con los que poder contactar.



Figura 0-35: Ventana del chat del usuario en el sistema

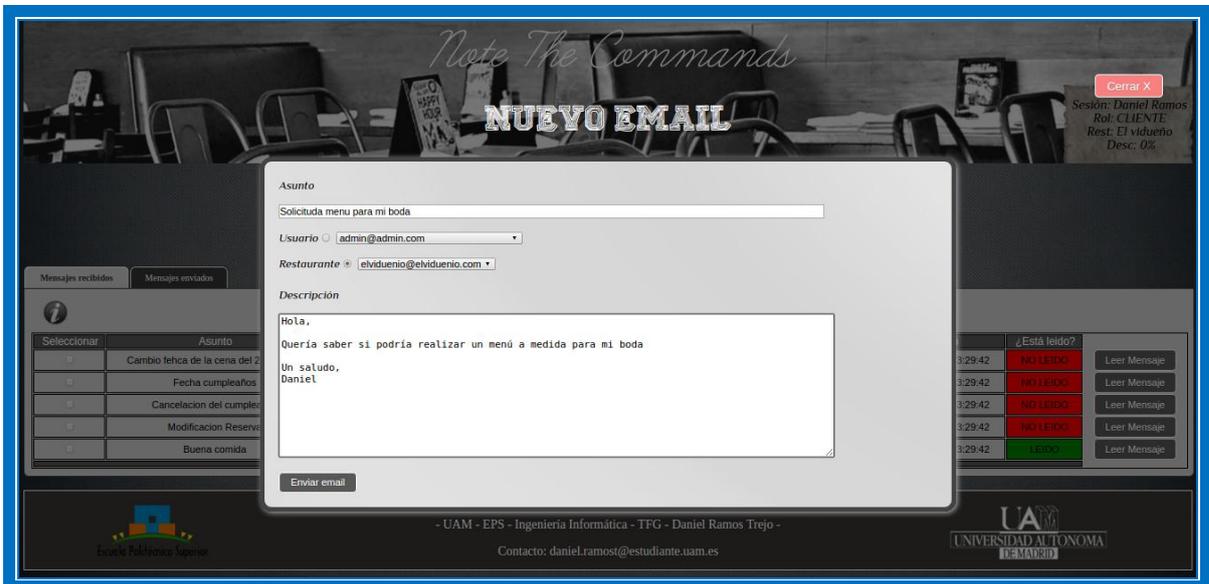
Si seleccionamos una conversación abierta se nos mostrará una pantalla con el historial de la conversación tal y como la dejamos.



**Figura 0-36: Ventana de una conversación del usuario**

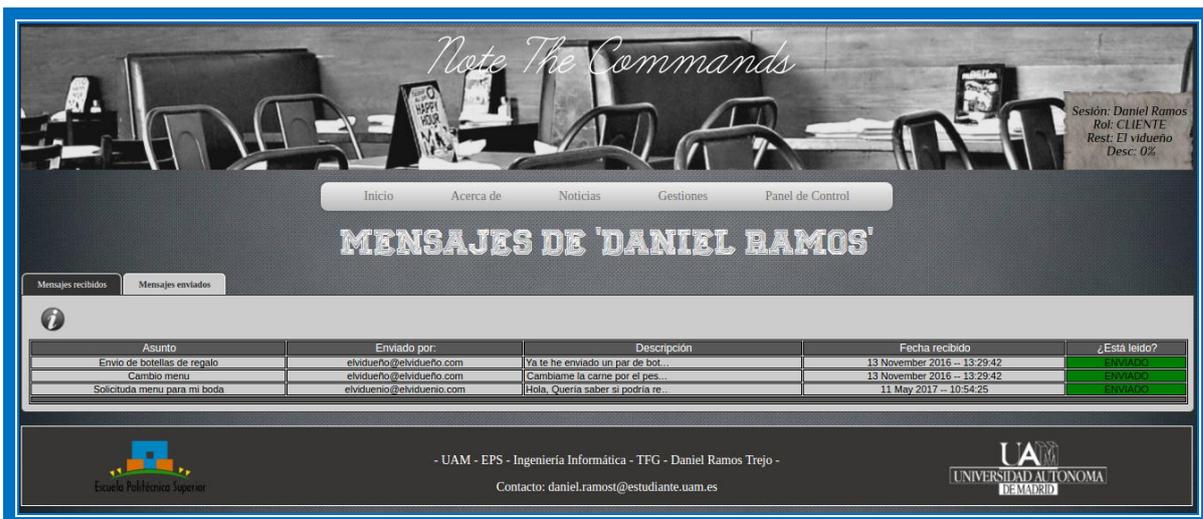
### **Correo del usuario.**

Si pulsamos en la barra de tareas en ‘Panel de Control → Bandeja Mensajes’ podremos visualizar los correos del usuario y redactar nuevos, o bien a otro usuario, o bien al restaurante en el que esta logueado. Los correos que se envíen a un restaurante estarán sincronizados con el correo del gerente de dicho restaurante.



**Figura 0-37: Ventana para redactar un nuevo correo**

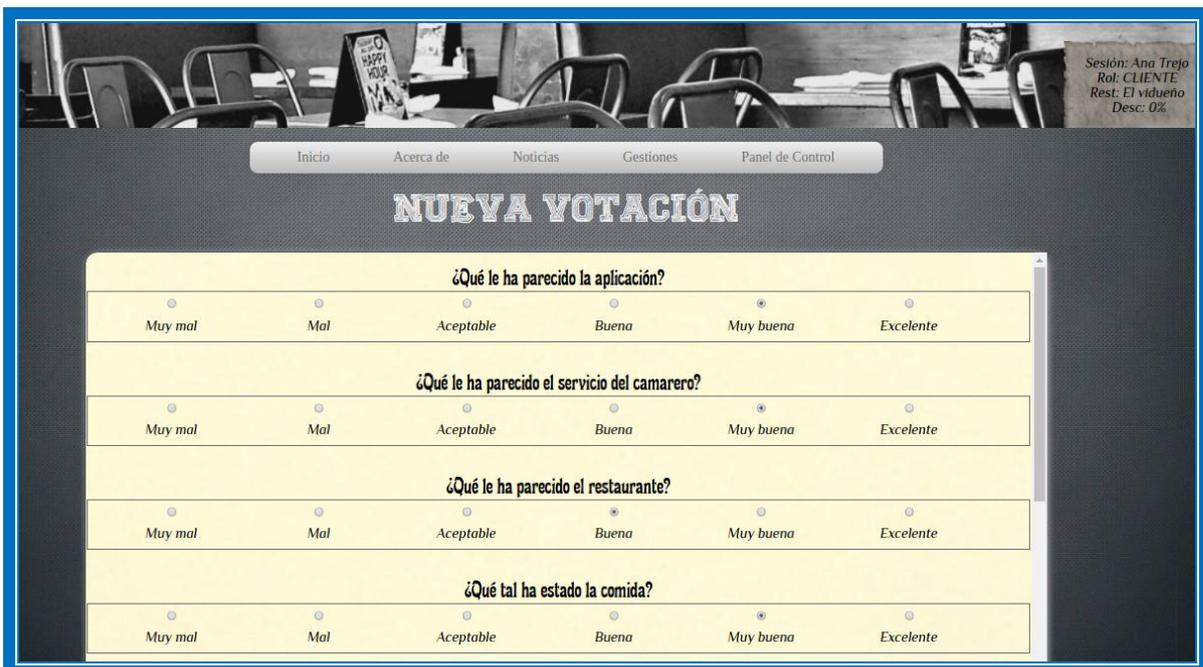
También se puede ver en la otra pestaña los correos enviados por el usuario a otros destinatarios.



**Figura 0-38: Ventana de los correos enviados**

### *Votaciones del usuario.*

Cuando el usuario finalice su comanda se le solicitará que realice un breve cuestionario mientras que el camarero le trae la cuenta. Los cuestionarios varían según el restaurante. A continuación, mostraremos un ejemplo:



**Figura 0-39: Ventana con cuestionarios del restaurante**

### *Perfil del cliente.*

Si pulsamos en la barra de tareas en 'Panel de Control → Mi Perfil' podremos visualizar el perfil del usuario que esta logueado, en este caso el del cliente, pudiendo modificarlo si es necesario. La ventana será similar a la de la Figura 0-9.

### ***Restaurantes del cliente.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Restaurantes’ podremos visualizar los restaurantes a los que está asociado el cliente como en la Figura 0-18.

### ***Cerrar sesión del cliente.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Cerrar Sesión’ podremos salir de la sesión del cliente. Esta funcionalidad estará disponible para todos los usuarios del sistema. Cabe destacar que si un cliente tiene comandas abiertas no podrá cerrar sesión.

## **GERENTE**

### ***Pantalla de inicio del gerente logueado.***

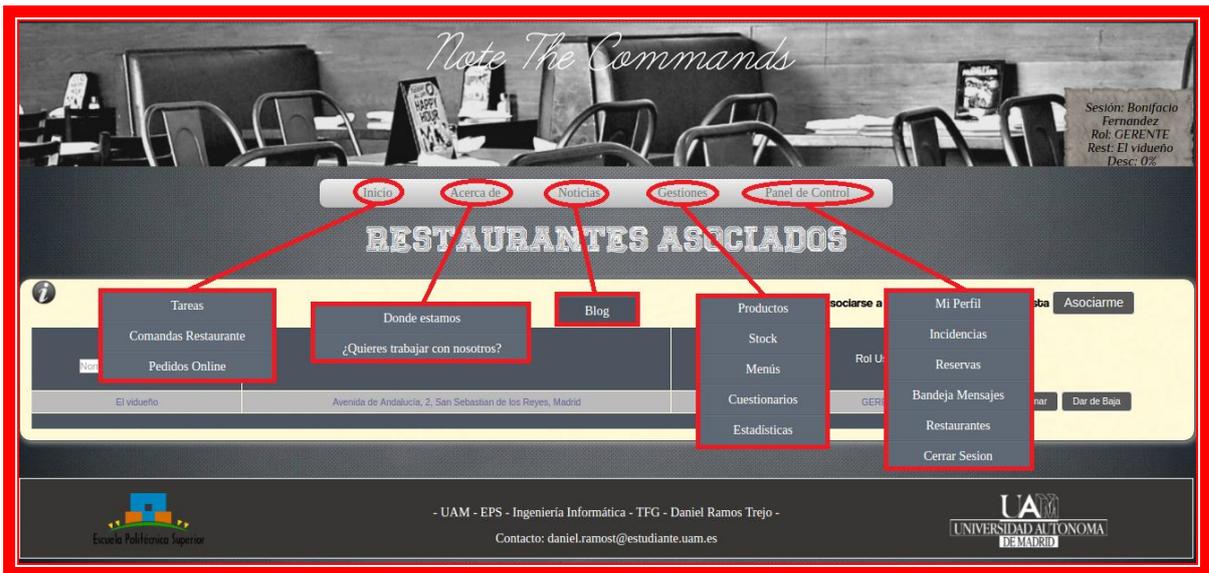
En esta pantalla se mostrarán los restaurantes a los que está dado de alta el gerente. Podrá darse de baja, darse de alta, seleccionar un restaurante de la lista o filtrar los restaurantes por su nombre.



**Figura 0-40: Ventana de inicio del gerente**

### ***Barra de tareas del gerente.***

A continuación, mostraremos las distintas funcionalidades que presenta la barra de tareas del gerente del sistema.



**Figura 0-41: Barra de tareas del gerente**

### *Tareas del gerente.*

Si pulsamos en la barra de tareas en 'Inicio → Tareas' se visualizan las tareas que el gerente puede realizar sobre la aplicación. Estas tareas son las mismas que las que aparecen en la barra de tareas de la Figura 0-41. Cada una de estas tareas funcionan como un botón, por lo que si pulsamos cualquiera de ellas nos direccionará a la ventana correspondiente.



**Figura 0-42: Ventana de tareas del gerente**

### *Comandas del restaurante para el gerente.*

Si pulsamos en la barra de tareas en 'Inicio → Comandas del Restaurante' podremos visualizar todas las comandas del restaurante ordenadas por si están o no

finalizadas. En ella se podrá visualizar el nombre de la comanda, la fecha y el coste de la comanda. Cabe destacar que si la comanda está abierta, aun no mostrará el coste final ya que el cliente puede seguir añadiendo productos o menús y el coste puede variar, por tanto, hasta que no se finalice no mostrará el precio final. Esto sucedía también en el cliente como hemos explicado anteriormente. También podremos visualizar el botón ‘*ver comanda*’ para ver el contenido de la comanda.

Session: Bonifacio Fernandez  
Rol: GERENTE  
Rest: El vidueño  
Desc: 0%

Inicio Acerca de Noticias Gestiones Panel de Control

### COMANDAS DEL RESTAURANTE

Comanda	Restaurante	Usuario	Mesa	Fecha Creación	Ganancia por Comanda	¿Servido?	¿Está Finalizada?	
Manual	El vidueño	Daniel Ramos	1	11 May 2017 – 12:19:17	0.00 €	false	false	Ver Comanda
Día de relax	El vidueño	Miguel Ruiz	9	17 February 2017 – 13:29:42	133.00 €	true	true	Ver Comanda
Comida entre amigos	El vidueño	Daniel Ramos	10	17 February 2017 – 13:29:42	82.20 €	true	true	Ver Comanda
Celebración fiesta del caballo	El vidueño	Daniel Trejo	12	16 February 2017 – 13:29:42	42.75 €	true	true	Ver Comanda
Cumplimientos de mi padre	El vidueño	Carolina Llamas	6	16 February 2017 – 13:29:42	135.95 €	true	true	Ver Comanda
Partido Real Madrid	El vidueño	Jose Fuertes	3	15 February 2017 – 13:29:42	605.50 €	true	true	Ver Comanda
Tarde de terrazo	El vidueño	Maria Guinea	2	15 February 2017 – 13:29:42	166.70 €	true	true	Ver Comanda
Cumple mañá	El vidueño	Daniel Ramos	1	15 February 2017 – 13:29:42	179.95 €	true	true	Ver Comanda

Previous 1 2 Next

**Figura 0-43: Ventana de comandas del restaurante**

***Pedidos online del restaurante para el gerente.***

Si pulsamos en la barra de tareas en ‘*Inicio → Pedidos Online*’ podremos visualizar todos los pedidos online de todos los clientes del restaurante.

Session: Bonifacio Fernandez  
Rol: GERENTE  
Rest: El vidueño  
Desc: 0%

Inicio Acerca de Noticias Gestiones Panel de Control

### PEDIDOS AL RESTAURANTE 'EL VIDUEÑO'

Nombre del Pedido	Teléfono	Día Recogida	Hora Recogida	Restaurante	Ganancia Pedido	¿Está Finalizada?	¿Está Entregado?	
Ana Garcia	632595987	28 July 2017	13:30:00	El vidueño	192.7 €	false	false	Ver Pedido
Pepe Alonso	699696969	02 December 2016	13:30:00	El vidueño	100.3 €	true	true	Ver Pedido
Olga Iborra	650254896	26 December 2016	21:30:00	El vidueño	168.4 €	true	true	Ver Pedido
Borja Cuesta	75896525	31 December 2016	20:30:00	El vidueño	99.5 €	true	true	Ver Pedido
Almudena Alonso	698545478	02 January 2017	13:30:00	El vidueño	55.7 €	true	true	Ver Pedido

**Figura 0-44: Ventana de pedidos online del restaurante para el gerente**

Si pulsamos en el botón ‘*Ver Pedido*’ nos mostrará los detalles del pedido como se aparece en la Figura 0-45.

Sesión: Bonifacio Fernandez  
Rol: GERENTE  
Rest: El vidueño  
Desc: 0%

Inicio   Acerca de   Noticias   Gestiones   Panel de Control

## PEDIDO ONLINE NÚMERO: 5

**Pedido**

Nombre Pedido: Ana Garcia  
Dia de recogida: 07/28/17  
Hora de recogida: 13:30:00  
Resaurante: El vidueño  
Descuento adicional: 0%

Nombre Producto	Cantidad	Precio	Coste
Dorada a la espalda	3 Unds	9.90 €/Unds	29.70 €
Merluza a la plancha	6 Unds	8.50 €/Unds	51.00 €
Cordero lechal asado	8 Unds	8.50 €/Unds	68.00 €
Coca-Cola normal	10 Unds	2.10 €/Unds	21.00 €
Cerveza Estrella Galicia	20 Unds	2.40 €/Unds	48.00 €
<b>Total:</b>			<b>217.70 €</b>

- UAM - EPS - Ingeniería Informática - TFG - Daniel Ramos Trejo -  
Contacto: daniel.ramost@estudiante.uam.es

Escuela Politécnica Superior   UAM   UNIVERSIDAD AUTÓNOMA DE MADRID

**Figura 0-45: Ventana de detalles del pedido online para el gerente**

### *Localización del restaurante.*

Si pulsamos en la barra de tareas en ‘Acerca de → Donde estamos’ podremos ver la localización del restaurante que hayamos seleccionado previamente en *Google Maps* dentro de la propia aplicación. Un ejemplo de ello sería la Figura 0-25.

### *Currículos de los usuarios.*

Si pulsamos en la barra de tareas en ‘Acerca de → ¿Quieres trabajar con nosotros?’ podremos ver la ventana donde el gerente podrá también subir un curriculum.

Sesión: Bonifacio Fernandez  
Rol: GERENTE  
Rest: El vidueño  
Desc: 0%

Inicio   Acerca de   Noticias   Gestiones   Panel de Control

## ¡MANDENOS SU CURRÍCULUM VITAE!

*Te apetece trabajar en El vidueño?*

Entonces ¡estás de enhorabuena! Somos una empresa joven y buscamos un equipo de personas con ganas de hacer un gran trabajo mientras se divierten. En 'El vidueño' cuidamos especialmente la atención al cliente para ofrecerle una experiencia de consumo perfecta. Tenemos diferentes perfiles definidos para nuestros restaurantes. Si encajas en algún puesto y quieres que contemos contigo en próximos procesos de selección, este es tu momento. ¿a qué esperas?

- **GERENTE:** Responsable máximo del restaurante, cuya responsabilidad es implantar y desarrollar las políticas, principios y directrices marcadas por la Cía. Responsable de potenciar el desarrollo profesional y personal del equipo, asegurando un excelente clima laboral.
- **PERSONAL:** Responsable de alcanzar metas de satisfacción del cliente. Es un puesto perfecto para personal con iniciativa, ganas de trabajar en equipo, de formarse y crecer en una gran compañía.

Seleccionar archivo   Ningún archivo seleccionado   Enviar Curriculum

Ver Currícula

**Figura 0-46: Ventana de envío de curriculum para el gerente**

Al pulsar el botón ‘Ver Currricula’ el gerente podrá visualizar cada uno de los curriculum que los usuarios han ido enviando al restaurante.



Figura 0-47: Ventana *curricula* para el gerente

### *Blogs del restaurante.*

Si pulsamos en la barra de tareas en ‘Noticias → Blogs’ podremos visualizar las noticias del restaurante. Estas noticias fueron creadas por el gerente y por ello tendrá permisos para modificarlas, eliminarlas, comentarlas o también podrá crear nuevas noticias.

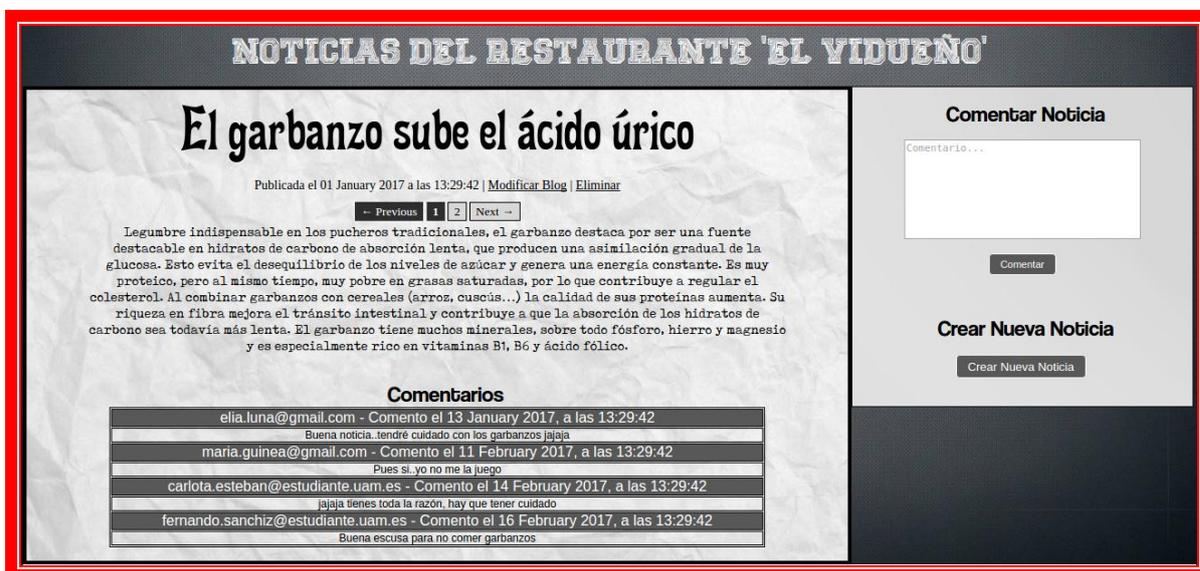


Figura 0-48: Ventana de noticias para el gerente del restaurante

### *Productos del restaurante.*

Si pulsamos en la barra de tareas en ‘Gestiones → Productos’ podremos visualizar todos los productos que están dados de alta en el restaurante. En esta ventana se puede

observar una tabla con los productos del restaurante los cuales podremos filtrarlos por el nombre o también dar de baja un determinado producto del restaurante.



**Figura 0-49: Ventana de productos del restaurante**

En la parte superior derecha de la tabla podemos ver el botón ‘*Crear Producto*’ mediante el cual al pulsarlo podremos dar de alta un nuevo producto en el restaurante.



**Figura 0-50: Ventana para crear un nuevo producto en el restaurante**

### **Stock del restaurante.**

Si pulsamos en la barra de tareas en ‘*Gestiones → Stock*’ podremos visualizar el stock del restaurante, esto es, todos los productos de los que dispone el restaurante en el almacén. En esta ventana se muestra una tabla con el nombre del producto, el tipo del producto, el coste de venta del producto, la cantidad que hay en el almacén del producto, el

precio de venta del producto por unidad y el coste total de la materia prima (multiplicando la cantidad de un producto por el coste de la materia prima del mismo). También podremos filtrar por descripción de un producto y a su vez en caso de querer modificar algún parámetro de un producto en el stock podremos utilizar el botón ‘*Editar Stock*’, donde entre otras cosas se podrá modificar la cantidad de existencias que hay de un producto, ya sea porque se haya realizado un pedido o porque no coincida con las existencias reales que hay en el almacén.

Restaurante	Tipo Producto	Nombre Producto	Descripción	Coste de venta	Cantidad en Stock	Coste Materia Prima Unidad	Coste Total Materia Prima	
El vidueño	RACION	Morcilla de Burgos	Morcilla de Burgos unidad	€5.90	2	€2.10	€4.20	Editar Stock
El vidueño	POSTRE	Cortado	Cafe cortado	€13.00	6	€0.20	€1.20	Editar Stock
El vidueño	RACION	Patatas bravas	Patatas bravas unidad	€4.90	8	€1.20	€9.60	Editar Stock
El vidueño	VERDURAS	Pochas a la Riojana	Pochas a la Riojana	€7.55	9	€1.90	€17.10	Editar Stock
El vidueño	APERITIVO	Patatas fritas	Patatas fritas unidad	€0.20	10	€0.90	€9.00	Editar Stock
El vidueño	CARNES-ROJAS	Secreto ibérico a la parrilla	Secreto ibérico a la parrilla	€5.75	11	€5.20	€57.20	Editar Stock
El vidueño	POSTRE	Cafe con leche	Cafe con leche unidad	€1.20	12	€0.20	€2.40	Editar Stock
El vidueño	RACION	Cazon en adobo	Cazon en adobo unidad	€8.90	12	€3.20	€38.40	Editar Stock
El vidueño	VERDURAS	Parrillada de verduras	Parrillada de verduras	€6.75	12	€1.30	€15.60	Editar Stock
El vidueño	VERDURAS	Revuelto de habitas con morcilla	Revuelto de habitas con morcilla	€7.75	12	€4.20	€50.40	Editar Stock
El vidueño	RACION	Croquetas Caseras	Croquetas unidad	€6.00	15	€0.20	€3.00	Editar Stock
El vidueño	RACION	Migas Extremeñas	Migas Extremeñas unidad	€6.50	15	€0.60	€9.00	Editar Stock

Figura 0-51: Ventana del stock del restaurante

En la parte superior derecha de la tabla podemos ver el botón ‘*Añadir Producto al Stock*’ mediante el cual si lo pulsamos podremos añadir un nuevo producto al stock.

Restaurante	Tipo Producto	Nombre Producto	Descripción	Coste de venta	Cantidad en Stock	Coste Materia Prima Unidad	Coste Total Materia Prima	
El vidueño	RACION	Morcilla de Burgos	Morcilla de Burgos unidad	€5.90	2	€2.10	€4.20	Editar Stock
El vidueño	POSTRE	Cortado	Cafe cortado	€13.00	6	€0.20	€1.20	Editar Stock
El vidueño	RACION	Patatas bravas	Patatas bravas unidad	€4.90	8	€1.20	€9.60	Editar Stock
El vidueño	VERDURAS	Pochas a la Riojana	Pochas a la Riojana	€7.55	9	€1.90	€17.10	Editar Stock
El vidueño	APERITIVO	Patatas fritas	Patatas fritas unidad	€0.20	10	€0.90	€9.00	Editar Stock
El vidueño	CARNES-ROJAS	Secreto ibérico a la parrilla	Secreto ibérico a la parrilla	€5.75	11	€5.20	€57.20	Editar Stock
El vidueño	POSTRE	Cafe con leche	Cafe con leche unidad	€1.20	12	€0.20	€2.40	Editar Stock
El vidueño	RACION	Cazon en adobo	Cazon en adobo unidad	€8.90	12	€3.20	€38.40	Editar Stock
El vidueño	VERDURAS	Parrillada de verduras	Parrillada de verduras	€6.75	12	€1.30	€15.60	Editar Stock
El vidueño	VERDURAS	Revuelto de habitas con morcilla	Revuelto de habitas con morcilla	€7.75	12	€4.20	€50.40	Editar Stock
El vidueño	RACION	Croquetas Caseras	Croquetas unidad	€6.00	15	€0.20	€3.00	Editar Stock
El vidueño	RACION	Migas Extremeñas	Migas Extremeñas unidad	€6.50	15	€0.60	€9.00	Editar Stock

Figura 0-52: Ventana para registrar un producto en el stock de un restaurante





**Figura 0-55: Ventana para crear un nuevo menú para el restaurante**

### *Cuestionarios del restaurante.*

Si pulsamos en la barra de tareas en 'Gestiones → Cuestionarios' podremos visualizar los cuestionarios que el gerente ha generado para su restaurante. El gerente podrá crear nuevos cuestionarios al restaurante mediante el botón 'Crear Cuestionario' situado en la parte superior derecha de la tabla.



**Figura 0-56: Ventana de cuestionarios creados por el gerente**

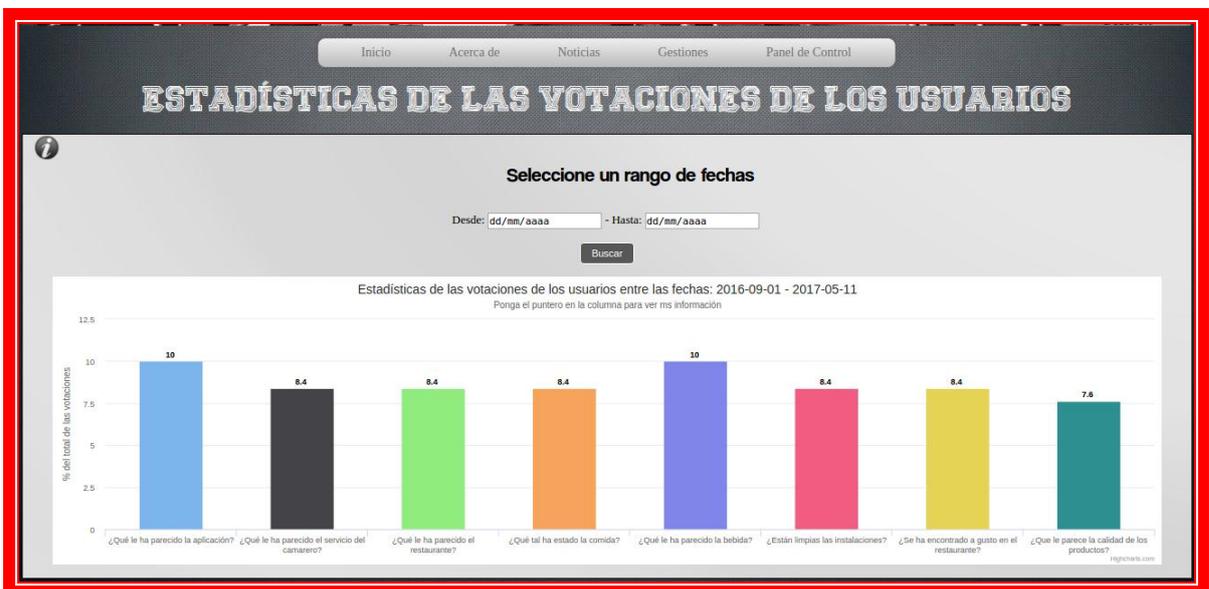
### *Estadísticas del restaurante para el gerente.*

Si pulsamos en la barra de tareas en 'Gestiones → Estadísticas' podremos visualizar las estadísticas del restaurante que están disponibles para el gerente.



**Figura 0-57: Ventana de estadísticas de las que dispone el gerente**

En este caso el gerente dispondrá de una ventana de estadísticas para los productos más consumidos en un rango de tiempo como vimos en la Figura 0-34. También podrá visualizar las estadísticas de las votaciones que los clientes de forma anónima han ido realizando mientras el camarero le suministraba la cuenta de la comanda como vimos en la Figura 0-39. En la siguiente imagen hemos filtrado la búsqueda de la estadística entre dos fechas y las puntuaciones obtenidas son las siguientes:



**Figura 0-58: Ventana de estadísticas de las votaciones que dispone el gerente**

### ***Incidencias del restaurante.***

Si pulsamos en la barra de tareas en '*Panel de Control* → *Incidencias*' podremos visualizar las incidencias del restaurante. El gerente podrá tener constancia de todas las categorías de las incidencias que se produzcan en el restaurante. También dispondrá de un botón '*Nueva Incidencia*' para crear una nueva incidencia que haya visto y tener registro un registro de la misma.

Sesión: Bonifacio  
Fernandez  
Rol: GERENTE  
Rest: El vidueño  
Desc: 0%

Inicio   Acerca de   Noticias   Gestiones   Panel de Control

## INCIDENCIAS DEL RESTAURANTE

Pendientes   Terminadas   Totales

Crear nueva Incidencia   Nueva Incidencia

Título	Categoría	Prioridad	Descripción	Fecha creación	Usuario	¿Esta resuelto?	
Hay una luz fundia en la cocina	COCINA	MEDIA	Hay una bombilla que no deja ...	13 November 2016 -- 13:29:42	admin	false	Edit Incidencia
Luz fundida en el leterero publicitario del Restaurante	ELECTRICIDAD	MEDIA	El leterero del Restaurante he v...	13 November 2016 -- 13:29:42	admin	false	Edit Incidencia

- UAM - EPS - Ingeniería Informática - TFG - Daniel Ramos Trejo -  
Contacto: daniel.ramos@estudiante.uam.es

UNIVERSIDAD AUTÓNOMA DE MADRID

**Figura 0-59: Ventana de incidencias para el gerente del restaurante**

### *Reservas del restaurante.*

Si pulsamos en la barra de tareas en 'Panel de Control → Reservas' podremos visualizar las reservas con fecha de hoy, las futuras, las finalizadas y las totales. En la Figura 0-60 vemos en una tabla las reservas totales del restaurante *El Vidueño*, donde podremos ver el nombre de la reserva, el número de personas, si tiene o no menú la reserva, el día de la reserva, la hora de la reserva, quien ha realizado la reserva y si está o no despachada la reserva, es decir, si el camarero le ha dado ya la mesa y la comanda o bien ha comenzado o bien ha finalizado.

Fernandez  
Rol: GERENTE  
Rest: El vidueño  
Desc: 0%

Inicio   Acerca de   Noticias   Gestiones   Panel de Control

## RESERVAS DEL RESTAURANTE

Hoy   Futuras   Finalizadas   Totales

Nombre reserva	Número de personas	Menú	Día reserva	Hora reserva	Restaurante	Teléfono contacto	Reserva realizada por:	¿Despachado?
Alvaro Arnedo	9		10 August 2017	21:30:00	El vidueño	699654775	Daniel Ramos	false
Daniel Chavani	12		01 July 2017	20:00:00	El vidueño	633525652	Daniel Ramos	false
Pedro Ferrer	200		28 June 2017	20:00:00	El vidueño	698742255	Daniel Ramos	false
Ella Luna	23		07 May 2017	14:00:00	El vidueño	698854445	Daniel Ramos	false
Daniel Vega	20		12 March 2017	21:30:00	El vidueño	325698587	Daniel Ramos	false
Vicente Teruel	2		14 February 2017	21:00:00	El vidueño	644235698	Daniel Ramos	true
Silvia Angaita	7		17 January 2017	20:00:00	El vidueño	699674585	Daniel Ramos	true
Luis Perez	3		02 January 2017	14:15:00	El vidueño	655789954	Daniel Ramos	true
Pilar Ezcurra	8		26 December 2016	21:30:00	El vidueño	655411202	Daniel Ramos	true
Daniel Ramos	23		20 December 2016	21:00:00	El vidueño	658899601	Daniel Ramos	true

**Figura 0-60: Ventana de reservas del restaurante**

Por ejemplo, si vemos la pestaña de las reservas de hoy podemos ver que ha llegado una reserva con nombre *ReservaManual*. A continuación, el camarero o el gerente podrá

despachar la reserva cuando la mesa esté lista. También se permitirá mediante el botón ‘Realizar Reserva’, crear una nueva reserva por parte del gerente o del camarero.

Nombre reserva	Número de personas	Menú	Día reserva	Hora reserva	Restaurante	Teléfono contacto	Reserva realizada por:	¿Despachado?
ReservaManual	2		14 May 2017	20:20:00	El vidueño	1231231	Daniel Ramos	false

**Figura 0-61: Ventana de reservas de hoy para el restaurante**

### ***Perfil del cliente.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Mi Perfil’ podremos visualizar el perfil del usuario que esta logueado, en este caso el del cliente, pudiendo modificarlo si es necesario. La ventana será similar a la de la Figura 0-9.

### ***Restaurantes del gerente.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Restaurantes’ podremos visualizar los restaurantes a los que está asociado el gerente, independientemente del restaurante donde es empleado, el gerente podrá asociarse a los restaurantes que quiera como cliente, aprovechando el registro de usuario que ya tiene.

### ***Correo del gerente.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Bandeja Mensajes’ se visualiza la ventana de envío y recepción de mensajes será similar a las del cliente que hemos visto anteriormente en la Figura 0-37 y la Figura 0-38. Cabe destacar que el gerente recibirá tanto los correos que le manden a él como los correos que los usuarios del sistema manden al restaurante en el que él es gerente.

### ***Cerrar sesión del usuario gerente.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Cerrar Sesión’ podremos salir de la sesión del usuario. Esta funcionalidad estará disponible para todos los usuarios del sistema.

## CAMARERO

### *Pantalla de inicio del camarero logueado.*

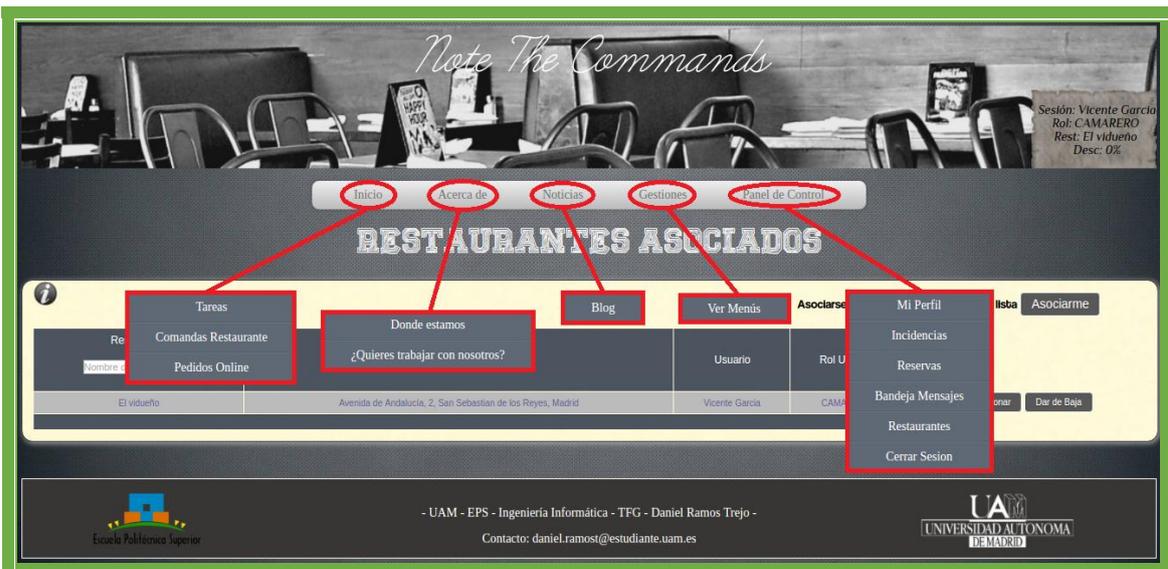
En esta pantalla se mostrará los restaurantes a los que está dado de alta el camarero. Podrá darse de baja, darse de alta, seleccionar un restaurante de la lista o filtrar un restaurante por su nombre.



**Figura 0-62: Ventana de inicio del camarero**

### *Barra de tareas del camarero.*

A continuación, mostraremos las distintas funcionalidades que presenta la barra de tareas del camarero del sistema.



**Figura 0-63: Barra de tareas del camarero**

### Tareas del camarero.

Si pulsamos en la barra de tareas en ‘Inicio → Tareas’ se visualizan las tareas que el camarero puede realizar sobre la aplicación. Estas tareas son las mismas que las que aparecen en la barra de tareas de la Figura 0-63. Cada una de estas tareas funcionan como un botón, por lo que si pulsamos cualquiera de ellas nos direccionará a la ventana correspondiente.



Figura 0-64: Ventana de tareas del camarero

### Comandas del restaurante para el camarero.

Si pulsamos en la barra de tareas en ‘Inicio → Comandas del Restaurante’ podremos visualizar todas las comandas del restaurante siendo las primeras las que aún no están finalizadas. En la tabla se podrá visualizar el nombre de la comanda, la fecha y el coste de la misma. Cabe destacar que si la comanda está abierta, aun no mostrará el coste final ya que el cliente puede seguir añadiendo productos o menús y el coste puede variar, por tanto, hasta que no se finalice no mostrará el precio final. Esto sucedía también en el cliente como hemos explicado anteriormente. También podemos visualizar el botón ‘ver comanda’ el cual nos permitirá ver el contenido de la comanda.



**Figura 0-65: Ventana de comandas para el camarero**

Si pulsamos en el botón ‘ver comanda’ podremos visualizar el desglose de la comanda y ver qué productos están listos para servir y cuáles no. A continuación, podremos ver la ventana con la comanda que habíamos creado anteriormente en la Figura 0-21 y Figura 0-22:



**Figura 0-66: Ventana con el desglose de la comanda para el camarero**

Si en la Figura 0-65 pulsamos en el botón ‘servir’ se servirán todos los productos que sean de bebidas, y también los de comida siempre y cuando el cocinero lo haya cocinado previamente. A continuación, vemos como aparecerían servidos los productos de bebida mientras que los de comida como aún no están cocinados se encuentran a la espera de ser servidos:



**Figura 0-67: Productos servidos por el camarero**

### ***Pedidos online del restaurante para el camarero.***

Si pulsamos en la barra de tareas en ‘Inicio → Pedidos Online’ podremos visualizar todos los pedidos online que los clientes han realizado para el restaurante en concreto. Esta ventana es igual que la que le aparece al gerente del restaurante que hemos visto en la Figura 0-29.

### ***Localización del restaurante para el camarero.***

Si pulsamos en la barra de tareas en ‘Acerca de → Donde estamos’ podremos ver la localización del restaurante que hayamos seleccionado previamente en google maps dentro de la propia aplicación. Esta ventana es similar a la Figura 0-25.

### ***Currícula para el camarero.***

Si pulsamos en la barra de tareas en ‘Acerca de → ¿Quieres trabajar con nosotros?’ podremos ver la ventana similar a la Figura 0-26, donde el camarero podrá también subir un currículum como hace el cliente, pero no tendrá los privilegios para poder ver los currícula como tiene el gerente.

### ***Blog del restaurante para el camarero.***

Si pulsamos en la barra de tareas en ‘Noticias → Blog’ podremos visualizar todas las noticias del blog del restaurante. En esta ventana podremos ver y comentar cada una de ellas, pudiendo dar nuestra opinión o contestar a las opiniones de los demás. Esta ventana será igual que la de la Figura 0-24.

### ***Menú del restaurante para el camarero.***

Si pulsamos en la barra de tareas en ‘Gestiones → Ver Menú’ podremos visualizar los menús del restaurante, tanto los que están como los que no están disponibles. Esta funcionalidad es similar a las del cliente como en la Figura 0-30 y la Figura 0-31.

### ***Perfil del usuario para el camarero.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Mi Perfil’ podremos visualizar el perfil del usuario que esta logueado, en este caso el del camarero, pudiendo modificarlo si es necesario. La ventana será similar a la de la Figura 0-9.

### ***Incidencias del restaurante para el camarero.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Incidencias’ podremos visualizar las incidencias del restaurante. El camarero podrá tener constancia solo de las categorías relacionadas con la comida. También dispondrá de un botón para crear una nueva incidencia que haya visto y tener registro un registro de la misma. La ventana será similar a la de la Figura 0-32.

### ***Reservas del restaurante para el camarero.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Reservas’ podremos visualizar las reservas con fecha de hoy y las futuras. En la Figura 0-68 vemos en una tabla las reservas de hoy del restaurante *El Vidueño*, donde podremos ver el nombre de la reserva, el número de personas, si tiene o no menú asignado, el día y la hora de la reserva, quien ha realizado la reserva y si está o no despachada, es decir, si el camarero le ha despachado ya la mesa, y la comanda o bien ha comenzado o bien ha finalizado.

Note The Commands

Sesión: Vicente García  
Rol: CAMARERO  
Rest: El vidueño  
Desc: 0%

Inicio Acerca de Noticias Gestiones Panel de Control

## RESERVAS DEL RESTAURANTE

Hoy Futuras

Crear nueva reserva Realizar Reserva

Nombre reserva	Número de personas	Menú	Día reserva	Hora reserva	Restaurante	Teléfono contacto	Reserva realizada por:	¿Despachado?	Despachar
ReservaManual	2		14 May 2017	20:20:00	El vidueño	1231231	Daniel Ramos	Si	Despachar

- UAM - EPS - Ingeniería Informática - TFG - Daniel Ramos Trejo -  
Contacto: daniel.ramost@estudiante.uam.es

Escuela Politécnica Superior UNIVERSIDAD AUTÓNOMA DE MADRID

**Figura 0-68: Ventana reservas del restaurante para el camarero**

### ***Correo del usuario camarero.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Bandeja de Mensajes’ podremos visualizar tanto la bandeja de entrada del camarero. La ventana de envío y recepción de mensajes será similar a las del cliente que hemos visto anteriormente en la Figura 0-37 y la Figura 0-38.

## ***Restaurantes del camarero.***

Si pulsamos en la barra de tareas en ‘Panel de Control → Restaurantes’ podremos visualizar los restaurantes a los que está asociado el camarero, independientemente del restaurante donde es empleado, el camarero podrá asociarse a los restaurantes que quiera como cliente, aprovechando el registro de usuario que ya tiene.

## **COCINERO**

### ***Pantalla de inicio del cocinero logueado.***

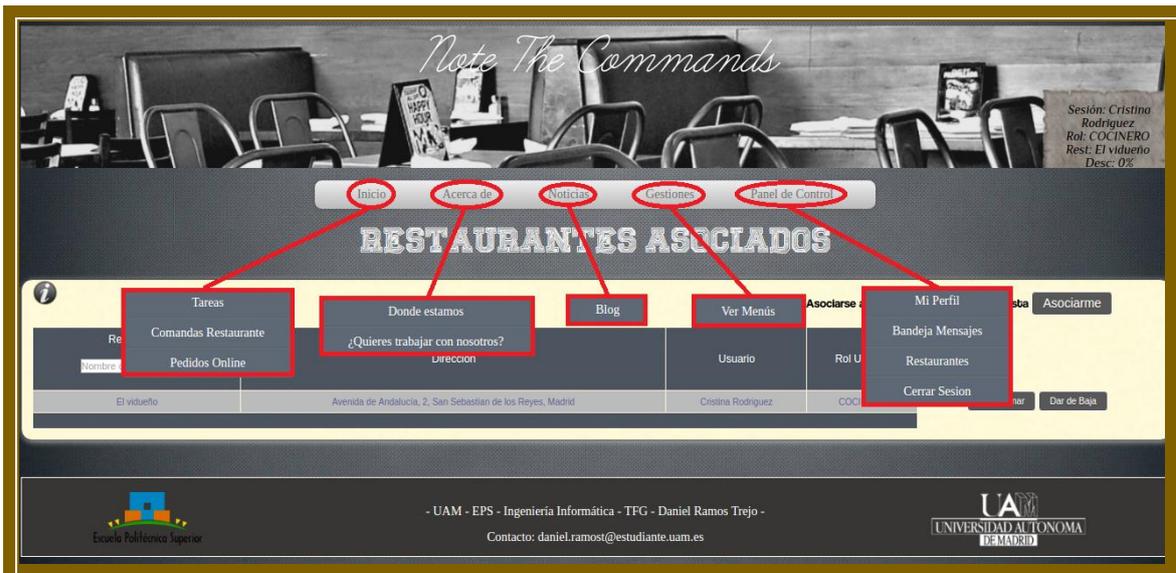
En esta pantalla se mostrará los restaurantes a los que está dado de alta el camarero. Podrá darse de baja, darse de alta o seleccionar un restaurante de la lista.



**Figura 0-69: Ventana de inicio del cocinero**

### ***Barra de tareas del cocinero.***

A continuación, mostraremos las distintas funcionalidades que presenta la barra de tareas del camarero del sistema.



**Figura 0-70: Barra de tareas del cocinero**

**Tareas del cocinero.**

Si pulsamos en la barra de tareas en 'Inicio → Tareas' se visualizan las tareas que el camarero puede realizar sobre la aplicación. Estas tareas son las mismas que las que aparecen en la barra de tareas de la Figura 0-70. Cada una de estas tareas funcionan como un botón, por lo que si pulsamos cualquiera de ellas nos direccionará a la ventana correspondiente.



**Figura 0-71: Ventana de tareas del cocinero**

**Comandas del restaurante para el cocinero.**

Si pulsamos en la barra de tareas en 'Inicio → Comandas del Restaurante' podremos visualizar los productos de comida que hay que cocinar para que el cocinero las pueda servir. Cuando el cocinero los tenga empatados y listos para servir le dará a cada

uno de ellos al botón cocinar y este le enviara un aviso al camarero de que el producto esta cocinado y listo para servir.



**Figura 0-72: Ventana de productos a cocinar**

### ***Pedidos online del restaurante para el cocinero.***

Si pulsamos en la barra de tareas en 'Inicio → Pedidos Online' podremos visualizar todos los pedidos online que los clientes han realizado para el restaurante en concreto. Esta ventana es igual que la que le aparece al gerente del restaurante que hemos visto en la Figura 0-29.

### ***Localización del restaurante para el cocinero.***

Si pulsamos en la barra de tareas en 'Acerca de → Donde estamos' podremos ver la localización del restaurante que hayamos seleccionado previamente en google maps dentro de la propia aplicación. Esta ventana es similar a la Figura 0-25.

### ***Currícula para el cocinero.***

Si pulsamos en la barra de tareas en 'Acerca de → ¿Quieres trabajar con nosotros?' podremos ver la ventana similar a la Figura 0-26, donde el cocinero podrá también subir un curriculum como hace el cliente, pero no tendrá los privilegios para poder ver los currícula como tiene el gerente.

### ***Blog del restaurante para el cocinero.***

Si pulsamos en la barra de tareas en 'Noticias → Blog' podremos visualizar todas las noticias del blog del restaurante. En esta ventana podremos ver y comentar cada una de ellas, pudiendo dar nuestra opinión o contestar a las opiniones de los demás. Esta ventana será igual que la de la Figura 0-24.

### ***Menú del restaurante para el cocinero.***

Si pulsamos en la barra de tareas en ‘*Gestiones → Ver Menú*’ podremos visualizar los menús del restaurante, tanto los que están como los que no están disponibles. Esta funcionalidad es similar a las del cliente como en la Figura 0-30 y la Figura 0-31.

### ***Perfil del usuario para el cocinero.***

Si pulsamos en la barra de tareas en ‘*Panel de Control → Mi Perfil*’ podremos visualizar el perfil del usuario que esta logueado, en este caso el del camarero, pudiendo modificarlo si es necesario. La ventana será similar a la de la Figura 0-9.

### ***Correo de mensajes del cocinero.***

Si pulsamos en la barra de tareas en ‘*Panel de Control → Bandeja de Mensajes*’ podremos visualizar tanto la bandeja de entrada del cocinero. La ventana de envío y recepción de mensajes será similar a las del cliente que hemos visto anteriormente en la Figura 0-37 y la Figura 0-38.

### ***Restaurantes del camarero.***

Si pulsamos en la barra de tareas en ‘*Panel de Control → Restaurantes*’ podremos visualizar los restaurantes a los que está asociado el cocinero, independientemente del restaurante donde es empleado, el cocinero podrá asociarse a los restaurantes que quiera como cliente, aprovechando el registro de usuario que ya tiene.

### ***Cerrar sesión del usuario cocinero.***

Si pulsamos en la barra de tareas en ‘*Panel de Control → Cerrar Sesión*’ podremos salir de la sesión del usuario. Esta funcionalidad estará disponible para todos los usuarios del sistema. Cabe destacar que si un cliente tiene comandas abiertas no podrá cerrar sesión.