



Universidad de Jaén

Escuela Politécnica Superior de Jaén

MENÚ ELECTRÓNICO CON SEGUIMIENTO EN TIEMPO REAL

Alumno: Javier Pinel Santafosta

Tutor: Manuel García Vega

Dpto: Informática



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Manuel Garcia Vega , tutor del Proyecto Fin de Carrera titulado: Menú Electrónico con seguimiento en tiempo real, que presenta Javier Pinel Santafosta, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Febrero de 2013

El alumno:

Javier Pinel Santafosta

Los tutores:

Manuel Garcia Vega

Agradecimientos:

Quisiera dar las gracias a mis familiares. A mis padres que por sus esfuerzos me han permitido tener una educación de calidad, por su cariño incondicional y por su ánimo cuando las cosas no van bien. A mi hermano Carlos que ha sido no solo un hermano, ha sido un gran amigo y compañero.

A antiguos compañeros de piso: Daniel, José, Luis Javier y Juan. Por hacer mi día a día más llevadero, por los ánimos de los que me dotaron y por la complicidad que acabé teniendo con ellos.

A Álvaro, David, Alba y Atria que han sido un apoyo moral inmejorable. Por proporcionar horas y horas de entretenimiento ayudándome a desconectar de las tareas más duras para poder afrontarlas con ganas.

A mis compañeros y compañeras que me acompañaron en mi Erasmus. Que me enseñaron que los mayores problemas no son más que pequeños baches superables como cualquier otra cosa en la vida.

A mi tutor en este proyecto Manuel por la paciencia que ha demostrado tener conmigo y que ha sabido inspirarme para que no perdiese el ritmo

Además de una mención a Lucas y Cristina, que me han acogido en su piso cuando el ambiente a mi alrededor estaba tan espeso que se podía pinchar con tenedor; permitiéndome continuar adelante y cumplir mis metas.

Por último a todos los profesores que han compartido su conocimiento conmigo. Porque también han sabido iluminar mi camino y que, sobre todo, han sabido transmitir su pasión por la informática.

A todos vosotros, muchas Gracias.

Contenido

1.	Introducción.....	7
1.1.	Prefacio.....	7
1.2.	Propósito del proyecto.....	7
1.3.	Objetivos del proyecto.....	7
1.4.	Resultados.....	8
1.5.	Metodología de desarrollo.....	8
1.5.1.	Desarrollo en espiral.....	8
1.5.2.	SCRUM.....	10
2.	Análisis inicial.....	12
2.2.	Análisis de Requisitos.....	12
2.2.1.	Requisitos de la aplicación del servidor:.....	12
2.2.2.	Requisitos para la aplicación del cliente.....	12
2.3.	Casos de uso.....	13
2.4.	Storyboard.....	14
2.5.	Diagrama de flujo de la aplicación de clientes.....	16
3.	Tecnologías.....	18
3.1.	PHP+MySQL.....	19
3.2.	AJAX.....	20
3.3.	Infraestructura.....	21
3.4.	Software.....	22
3.4.1.	CentOS 7 x86 64-Minimal:.....	22
3.4.2.	SSH:.....	22
3.4.3.	Apache:.....	22
3.4.4.	PHP 5.4.16:.....	22
3.4.5.	MySQL:.....	23
3.4.6.	Git 8.0.3:.....	23
3.4.7.	XAMPP:.....	23
4.	Planificación.....	24
5.	Seguimiento de Desarrollo.....	1
5.1.	Esquema de cada iteración.....	1
5.2.	Primer Sprint.....	1
5.2.1.	Planificación de Sprint.....	1

5.2.2.	Implementación	1
5.2.3.	Demostración de requisitos completados	3
5.2.4.	Retrospectiva.....	3
5.2.5.	Refinamiento de lista de requisitos y cambios en el proyecto	3
5.3.	Segundo Sprint	4
5.3.1.	Planificación	4
5.3.2.	Implementación	4
5.3.3.	Demostración de requisitos completados	5
5.3.4.	Retrospectiva.....	6
5.3.5.	Refinamiento de la lista de requisitos y cambios en el proyecto.....	6
5.4.	Tercer Sprint	6
5.4.1.	Planificación	6
5.4.2.	Implementación	6
5.4.3.	Demostración de requisitos completados	9
5.4.4.	Retrospectiva.....	10
5.4.5.	Refinamiento de requisitos y cambios en el proyecto	10
5.5.	Cuarto Sprint	10
5.5.1.	Planificación	10
5.5.2.	Implementación	10
5.5.3.	Demostración de requisitos completados	14
5.5.4.	Retrospectiva.....	14
5.5.5.	Refinamiento de la lista de requisitos y cambios en el proyecto.....	14
5.6.	Cuarto Sprint (extendido)	15
5.6.1.	Planificación	15
5.6.2.	Implementación	15
5.6.3.	Demostración de requisitos completados	18
5.6.4.	Retrospectiva.....	19
5.6.5.	Refinamiento de la lista de requisitos y cambios en el proyecto.....	20
5.7.	Quinto Sprint	20
5.7.1.	Planificación	20
5.7.2.	Implementación	21
5.7.3.	Demostración de requisitos completados	21
5.7.4.	Retrospectiva.....	22
5.7.5.	Refinamiento de la lista de requisitos y cambios en el proyecto.....	22
5.8.	Sexto Sprint	22
5.8.1.	Planificación	22

5.8.2.	Implementación	22
5.8.3.	Demostración de requisitos completados	23
5.8.4.	Retrospectiva.....	23
5.8.5.	Refinamiento de la lista de requisitos y cambios en el proyecto.....	23
5.9.	Septimo Sprint.....	24
5.9.1.	Planificación	24
5.9.2.	Implementación	24
5.9.3.	Demostración de requisitos completados	24
5.9.4.	Retrospectiva.....	25
5.9.5.	Refinamiento de la lista de requisitos y cambios en el proyecto.....	25
5.10.	Octavo Sprint	25
5.10.1.	Planificación	25
5.10.2.	Implementación	25
5.10.3.	Demostración de requisitos completados	26
5.10.4.	Retrospectiva.....	26
5.10.5.	Refinamiento de la lista de los requisitos y cambios en el proyecto	26
5.11.	Noveno Sprint	26
5.11.1.	Planificación	26
5.11.2.	Implementación	27
5.11.3.	Demostración de requisitos completados	28
5.11.4.	Retrospectiva.....	28
5.11.5.	Refinamiento de la lista de requisitos y Cambios en el proyecto	28
5.12.	Décimo Sprint (Fin del proceso de desarrollo).....	28
5.12.1.	Planificación	28
5.12.2.	Implementación	29
5.12.3.	Demostración de los requisitos completados	35
5.12.4.	Retrospectiva.....	35
5.12.5.	Refinamiento de la lista de requisitos y cambios en el proyecto.....	35
6.	Resumen	36
Anexo 1: Manual Técnico de instalación de servidor		39
Capítulo 1 Preparación del servidor.....		39
Capítulo 2 Instalación de un servicio web:.....		39
Anexo 2: Guía de usuario para administrador de la cocina		45
Interface principal		45
Editar platos		46
Editor de menús:.....		47

Editor de opciones	48
Cocina	49
Camareros.....	50
Anexo 3: Manual de cliente:.....	51
Página principal del cliente y elementos.....	51
Página de mesa	52
Página de Platos	53
Página de Cuenta.	54
Referencias:	55

1. Introducción

1.1. Prefacio

Ultimamente se puede apreciar como una gran cantidad de restaurantes de comida rápida están confiando en sistemas automatizados para la toma de pedidos en sus restaurantes.

En nuestra provincia no se ha hecho notar demasiado siendo McDonal's el ejemplo claro de esta tendencia. Pudiese parecer que al aparecer tantas aplicaciones destinadas a esto, abrirse paso con una aplicación más, no tiene muchas expectativas de hacerse hueco en el mercado.

La mayor parte de estas aplicaciones responden a un esquema básico en donde el usuario hace cola en una maquina dedicada, hace su pedido y espera a ver en un monitor que su número de pedido está listo para ser recogido.

Este método no deja muy atrás el modelo clásico de restaurante de comida rápida. Usa maquinas grandes, costosas y no soluciona el problema de las colas; incluso se acentúan cuando algún cliente no se encuentra cómodo o no sabe usar correctamente la máquina.

El problema de interacción con la maquina no tiene una fácil solución. No solo es un problema que incumba a quien esté usando el aparato en el momento, también es un problema de quien está esperando su turno. Pues la frustración de la lidiacon los menús se contagia a quien está mirando y solo puede esperar educado y paciente.

El problema no va a desaparecer fácilmente ante cualquier usuario con un perfil bajo en el uso de dispositivos electrónicos. No obstante se puede reducir el efecto colectivo que el problema de uso supone. En lugar de usar grandes maquinas en puntos específicos del recinto, colocar dispositivos en las propias mesas de los comensales.

1.2. Propósito del proyecto

Diseñar y desarrollar una aplicación que tome pedidos y permita su seguimiento a lo largo del proceso de preparación.

1.3. Objetivos del proyecto

- Realizar análisis orientado al desarrollo de software
- Documentar adecuadamente todo el proceso de desarrollo de la aplicación
- Formar una memoria donde se justifiquen los puntos anteriores
- Desarrollar la aplicación
- Redacción de un manual de usuario y un manual de asistencia técnica mínima.

1.4. Resultados

Como resultado se estima que el desarrollo del proyecto ha concluido cuando se cumplan los siguientes puntos:

- La aplicación cumpla todos los requerimientos del proyecto
- Una memoria que recopila todo el proceso de desarrollo
- Los diferentes manuales de usuario y asistencia

1.5. Metodología de desarrollo

La metodología de trabajo es un pilar fundamental en el proceso de ingeniería del software. Proporciona herramientas y un esquema de trabajo que mejora la productividad y la eficiencia en el proceso de desarrollo. Si además se documenta de forma adecuada, obtendremos una completa guía del proyecto que nos permitirá realizar cambios en el futuro y comprender las decisiones de diseño que ha llevado a la actual forma de la aplicación.

Comprendiendo la importancia de este punto en proceso de ingeniería de software Es fácil tener en cuenta el modelo de desarrollo en espiral.

1.5.1. Desarrollo en espiral.

Boehm, ideó y promulgó un modelo desde un enfoque distinto al tradicional en Cascada: El Modelo Evolutivo Espiral. Su Modelo de Ciclo de Vida en Espiral tiene en cuenta fuertemente el riesgo que aparece a la hora de desarrollar software. Para ello, se comienza mirando las posibles alternativas de desarrollo, se opta por la de riesgo más asumible y se hace un ciclo de la espiral. Tener las etapas tan bien diferenciadas me ayudaría a crear una documentación al día y bien estructurada. En cada ciclo de esta metodología distingo una fase de “Objetivos”

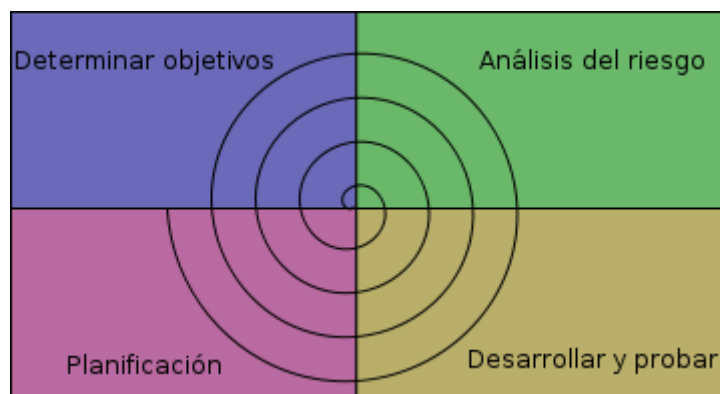


Ilustración 1: Modelo en Espiral

- Determinar objetivos
- Análisis de riesgo
- Desarrollo y prueba
- Planificación

Este modelo exige empezar con un prototipo. Esto se considera muy importante en el desarrollo de software pues ayuda mucho a tener una idea clara que ayude a no perder los objetivos finales. También es de gran utilidad para tener algo que enseñar al cliente. En resumen una gran fuente de retroalimentación (feedback) para el desarrollo de aplicaciones.

No obstante, una vez iniciado el proyecto, según este modelo, a pesar de sus ventajas, la falta de experiencia mella mucho. Ha sido sobre todo en la fase de "Análisis de riesgos" donde resulta muy difícil intentar prever los riesgos que implican las decisiones de diseño. Esto ligado a que el desarrollo de **cada iteración es demasiado largo** hace plantearse cambiar de modelo de desarrollo. Cambiar de modelo en una etapa de desarrollo tan temprana supone una reestructuración menor. Además de haber producido un storyboard.

Investigando mínimamente es fácil encontrar que uno de los modelos más populares que se demandan en las paginas web de busqueda de empleo, es la metodología SCRUM. Llama la atención la cantidad de empresas que no tiene como uno de sus requisitos estar familiarizado con ninguna metodología de trabajo. Excepto, que porque sí que algunas pocas piden algo es por la metodología SCRUM. Despues de barajar sus pros y contras, finalmente se ha decidido cambiar la forma de trabajo a SCRUM.

1.5.2.SCRUM

Scrum es una metodología ágil para planificar el desarrollo de software. Su principal característica es maximizar la **retroalimentación**. En este esquema, primero, se construye la funcionalidad de mayor valor y en cada iteración se añaden otras funcionalidades. En cada una de las iteraciones se termina una de las funcionalidades.

En un entorno real de trabajo, con la metodología Scrum, el cliente se puede comprometer más con el proyecto; lo ve crecer funcionalidad a funcionalidad. Esto permite rápidamente adaptar, añadir o eliminar características del programa para que se ajuste al máximo con el producto final. Scrum promueve la innovación, motivación y compromiso con el equipo que forma el proyecto.

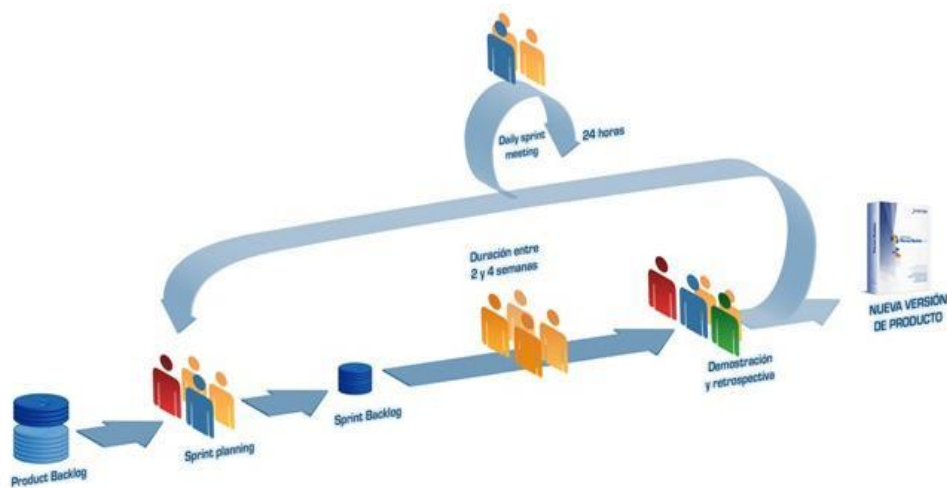


Ilustración 2: Proceso en Scrum

Son diversos los beneficios que proporciona Scrum. Siendo notorios la flexibilidad a cambios, reducción del "Time to Market" en donde el cliente podrá empezar a usar la aplicación antes de que esté terminada, mayor facilidad para predecir tiempos y por supuesto la reducción de riesgos. Al final, el cliente, sabrá cual va a ser el producto final.

Podemos distinguir tres fases en Scrum:

Planificación de iteración: Se realiza el primer día y se trata

- Selección de requisitos: Se extraen los requisitos del producto. Hay que priorizar esta lista. Un buen método es preguntar al propio cliente qué aspecto de la aplicación quiere primero. Esta fase no debe durar más de unas 4 horas.
- Planificación de la iteración: El equipo abstrae las tareas y reparte el trabajo sobre sus miembros. Esta tarea tampoco llevara más de 4 horas.

Ejecución de la iteración: Cada día el equipo hace una **reunión de sincronización** de unos 15 minutos. En esta reunión cada miembro

revisa el trabajo de sus compañeros con los que tenga una dependencia para poder abordar posibles problemas o adaptaciones. Cada miembro debería poder responder adecuadamente las siguientes preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante esta etapa el equipo se pone en marcha con el desarrollo. La duración de esta etapa ("Sprint") es variable entre proyectos pero fija en un único proyecto. En principio, para este proyecto, se van a realizar "Sprints" de 2 semanas por iteración.

Inspección y adaptación:

- Demostración (4h): Se presenta al cliente los progresos. De manera objetiva adaptar el proyecto y re planificar la siguiente iteración
- Retrospectiva (4h): El equipo analiza su manera de trabajar y que problemas han impedido que progresasen mejor para mejorar la productividad.

2. Análisis inicial

El problema al que se busca solución es desarrollar una aplicación para restaurantes cuyas funciones principales sean

- Tomar pedidos en las mesas de los clientes
- El cliente pueda saber si su pedido esta siendo procesado
- La información de los pedidos pueda ser visualizada y alterada en la cocina
- La aplicación pueda gestionarse de forma interna

2.2. Análisis de Requisitos

A continuación se van a exponer los requisitos funcionales de los que la aplicación debe ser capaz de cubrir. Estos requisitos han sido seleccionados pensando en que sea lo más completa posible. Al ser una aplicación de naturaleza cliente servidor se van a separar los requerimientos de cada una de las aplicaciones.

2.2.1. Requisitos de la aplicación del servidor:

- Administrable por formularios: La aplicación debe ser autosuficiente. La inserción de nuevas características se realizará por medio de la propia aplicación.
- Formulario para set de platos (menús): Como característica extra, se tiene que poder escoger una conjunto de platos que puedan funcionar de forma diferente. Como puede ser el precio.
- Secciones de:
 - Entrantes
 - Platos
 - Postres
 - Bebidas
- Estado de comandas: Las comandas sufren un seguimiento desde que se piden, hasta que se sirven.

2.2.2. Requisitos para la aplicación del cliente

- Control por número de mesa: Las mesas serán identificables dentro de la aplicación.
- Página de comandas (estado): Menú donde se puede ver lo pedido y en que estado del proceso se encuentra
- Platos divididos por categorías:
 - Entrantes
 - Platos
 - Postres
 - Bebidas
 - Menús
- Sección de "La cuenta": Donde se podrá ver un resumen de los platos pedidos y su importe total a modo de ticket
- Botón para pedir la ayuda de un camarero

2.3. Casos de uso

A continuación se presentan los casos de uso planteados para cada una de los subsistemas. Cabe destacar que los casos de uso que no se incluyeron en la fase de construcción sólo figuran en estado de propuestos.

Caso de uso	Pedir
Actor primario	Cliente
Sistema	Aplicación menú
Participantes	Cliente, Camarero
Nivel	Peticion de plato, petición de ayuda
Condición previa	Acceso a la aplicación
Operaciones basicas	
1	Selección de operación
2	Ver estado de mesa
3	Seleccionar menú
4	Seleccionar plato
5	Seleccionar bebida
6	Seleccionar postre
7	Pedir cuenta
8	Pedir ayuda
Alternativas	
1.A	Selecciona menú mesa
1.A.1	Pedir ayuda ir a 8
1.A.2	Ir a 2
1.B	Selecciona menú
1.B.1	Ir a 3
1.C	Seleccionar plato
1.C.1	Ir a 4
1.D	Selecciona bebida
1.D.1	Ir a 5
1.E	Seleccionar postre
1.E.1	Ir a 6
1.F	Pedir Cuenta
1.F.1	Ir a 7

Caso de uso	Gestionar cocina
Actor primario	Camarero/Encargado
Sistema	Aplicación Cocina
Participantes	Camarero/Cocinero
Nivel	Gestion de las comandas
Condición previa	Autorizado para gestionar
Operaciones basicas	
1	Subir prioridad de pedido
2	Cambiar estado de pedido
3	Atender camarero

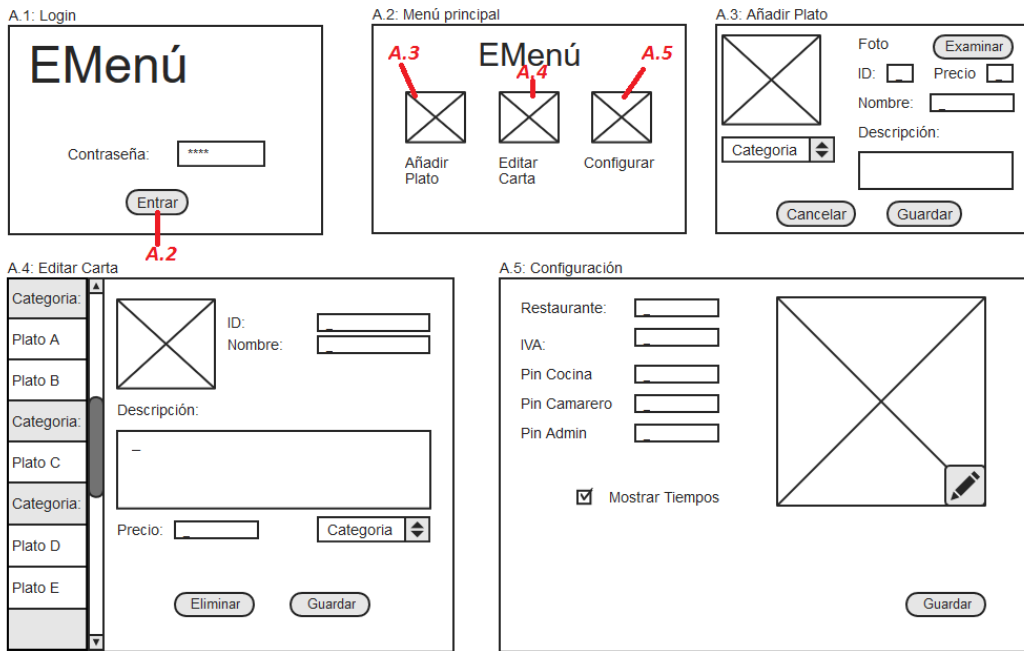
Alternativas	
1.A	Cambio de prioridad
1.A.1	Subir prioridad
1.A.2	Bajar prioridad
2.	Cambio de estado
2.A.1	En espera (por defecto)
2.A.2	En proceso
2.A.3	Entregado
2.B	Eliminar pedido
3.	Marcar mesa como atendida

Caso de uso	Gestionar Menú
Actor primario	Encargado
Sistema	Aplicación Administración
Participantes	Encargado
Nivel	Gestion de menú
Condición previa	Acceso a la aplicación
Operaciones basicas	
1	Selección de operación
2	Editar carta
3	Añadir Plato
4	Cambiar configuración
Alternativas	
1.A	Selección de operación
1.A.1	Si editar ir a 2
1.A.2	Si añadir ir a 3
1.A.3	Si configurar ir a 4

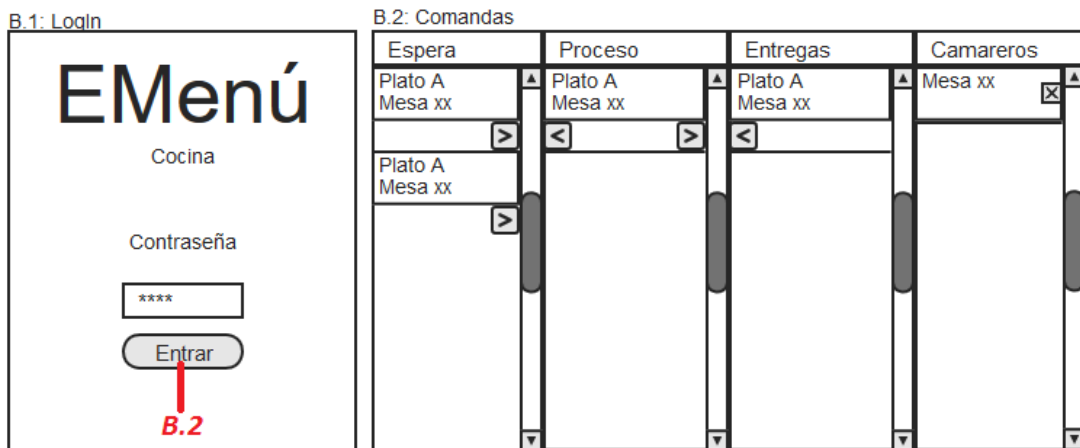
2.4. StoryBoard

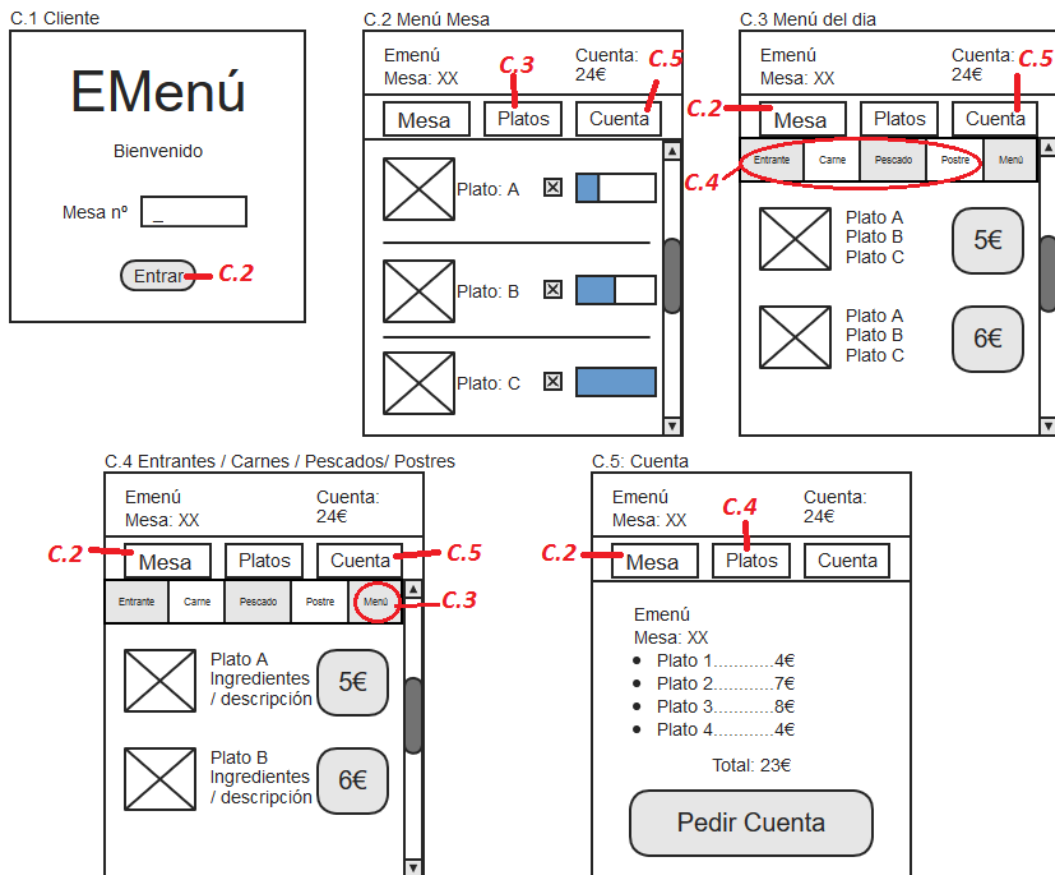
Ahora que están definidos los casos de usos, la visión global de la aplicación es mucho más sólida. Con el fin de consolidar de forma más clara el alcance de la aplicación, se diseña un story board que recoge los casos de uso y se le da profundidad a los diferentes menús.

Storyboard: Administración



Storyboard: Cocina





- 2.5. Diagrama de flujo de la aplicación de clientes
- Mientras que en las aplicaciones de administración y cocina, la navegación, es inmediata, la aplicación de clientes sí que tiene secciones y subsecciones.

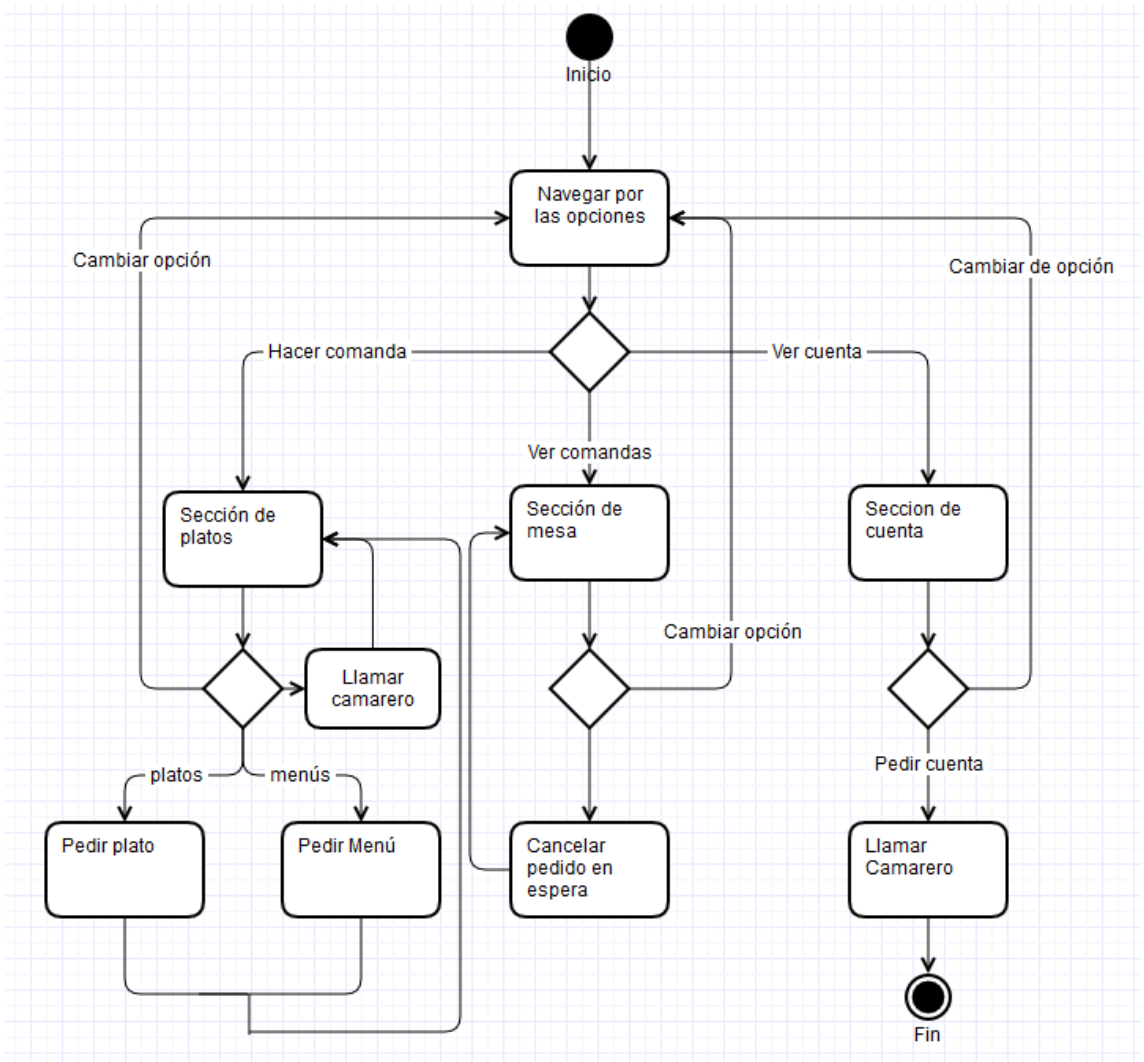


Ilustración 3: Diagrama Aplicación de Cliente

3. Tecnologías

Si nos centramos en España como principal público objetivo de la aplicación, nos encontramos con el dilema de qué plataforma escoger para iniciar el desarrollo.

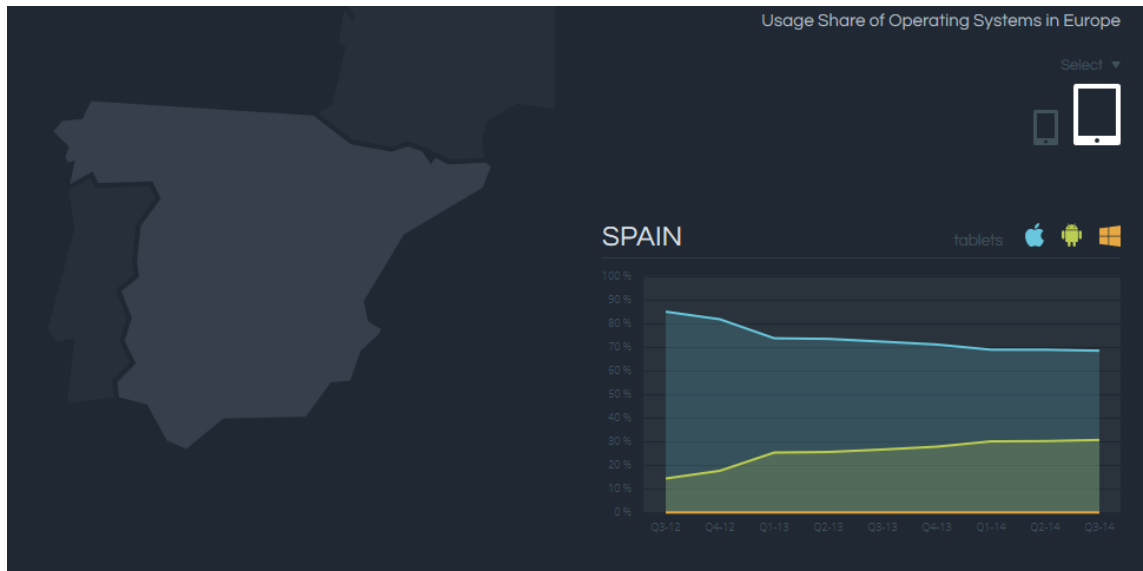


Ilustración 4: Uso de tabletas en España

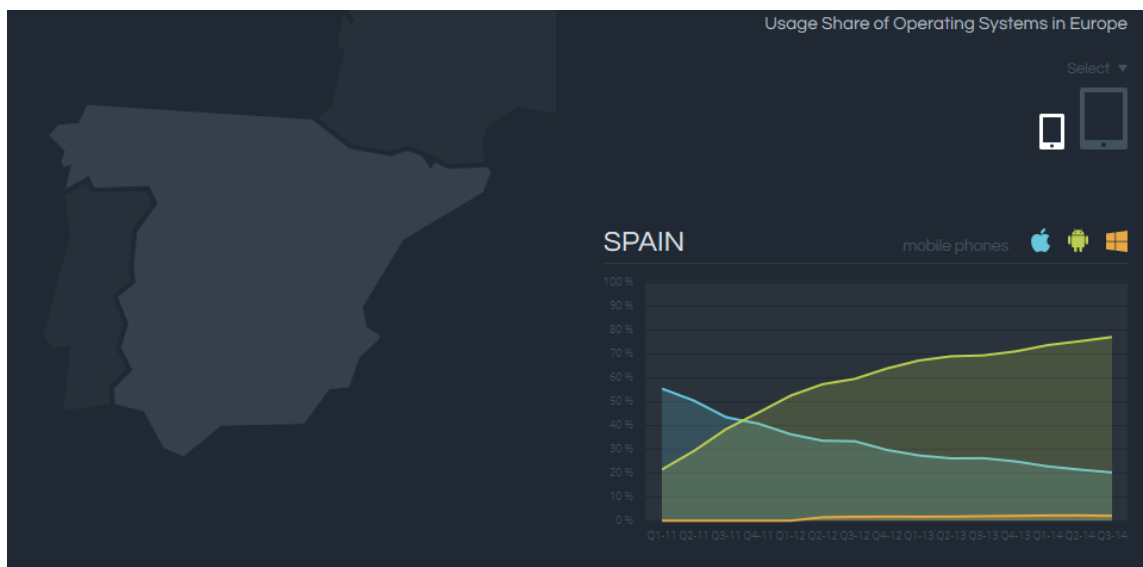


Ilustración 5: Uso de Smartphone en España

Pudiéramos pensar en desarrollar la aplicación de Tablet por su comodidad a la hora de mostrar cosas en una pantalla más grande. Pero en este punto estamos rechazando la posibilidad de que la aplicación se use de formas restringidas al establecimiento. Pensando en el futuro, en donde sea por ejemplo, que el establecimiento disponga del servicio pero no la infraestructura en las mesas para que la aplicación sea viable, el usuario podría venir de su casa con la aplicación cliente en el móvil. También queda la opción de que en vez de que el restaurante use tabletas/móviles use

equipos baratos de bajo rendimiento o una composición simple personalizada de raspberry pi o arduino.

Son muchos los factores a tener en cuenta si se quiere cubrir en caso de querer cubrir el mayor campo posible. La solución más universal es realizar la aplicación usando tecnologías web.

3.1. PHP+MySQL

El desarrollo web es seguramente la mejor opción para realizar la aplicación debido a su universalidad y a la disponibilidad de multitud de tecnologías que pueden ser usadas en la aplicación. Incluso, la aplicación tiene un claro esquema cliente servidor. Hay muy pocas formas de evitar necesitar un servidor debido a la naturaleza de la aplicación. Pudiendo aprovechar y quitamos carga de trabajo de procesamiento a los dispositivos cliente para permitir el uso de dispositivos de características bajas.

Usar una tecnología web parece realmente la más indicada para llegar al mayor número de clientes posibles. Ahora es cuestión de escoger un lenguaje de programación. Son múltiples las opciones, pero una vez más, pretendiendo abarcar todo lo posible una gran opción es usar la combinación de PHP + MySQL.

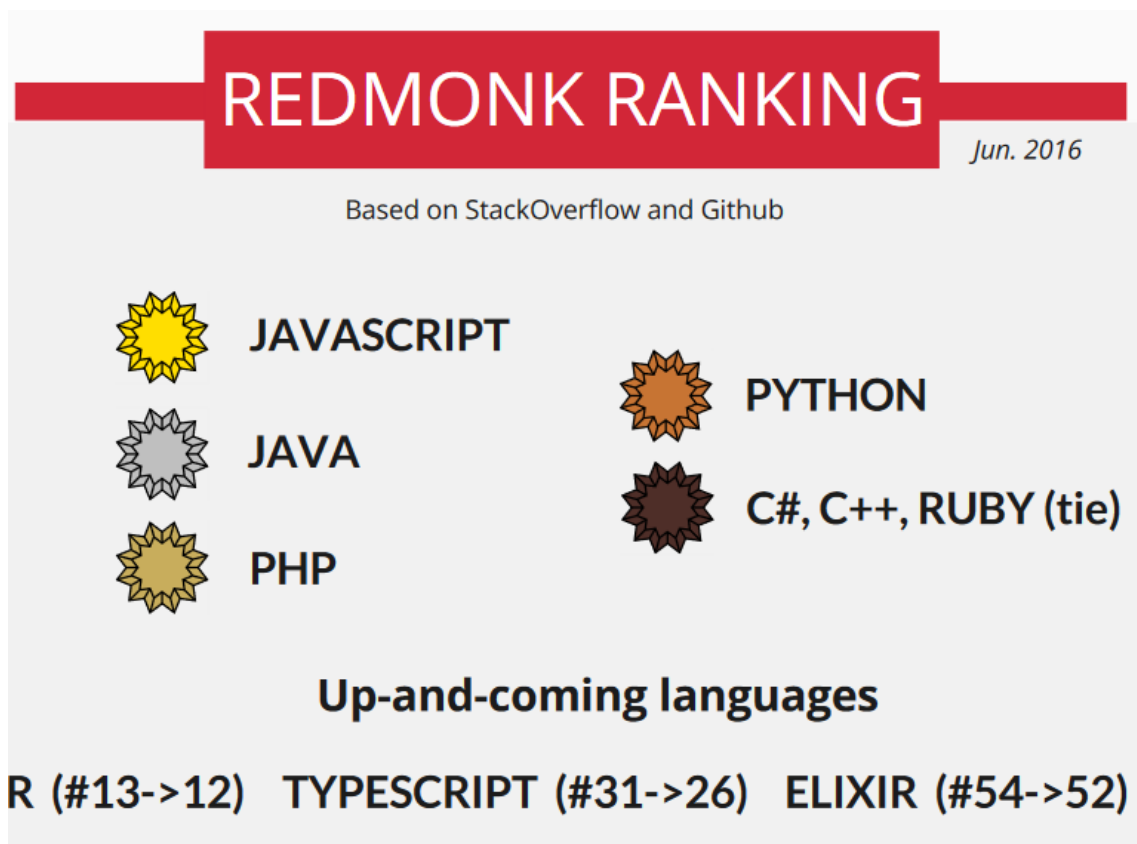


Ilustración 6: Ranking de popularidad de lenguajes por su comunidad

Principalmente por ser un lenguaje libre y abierto. Así se reducen costes de licencias, pudiendo llegar a evitarlas. El segundo motivo es su comunidad. PHP tiene una inmensa comunidad, en caso de necesidad, un cliente que quiera hacer cambios más profundos en el programa por cualquier motivo, podría hacerlo sin mayores problemas para encontrar con una correcta y estructurada implementación. La adaptación del código a un componente de CMS es muy simple. Las APIs de las diferentes versiones de SQL para PHP cambian mínimamente de una versión a otra.

3.2. AJAX

Ajax no es en sí un lenguaje de programación. "Ajax es un conjunto de métodos y técnicas que permiten intercambiar datos con un servidor y actualizar partes de páginas web sin necesidad de recargar la página completamente". Aunque Ajax se pensó inicialmente para transferir datos en un solo formato (XML), actualmente Ajax permite la transmisión de datos en múltiples formatos: XML, JSON, EBML, texto plano, HTML, etc.

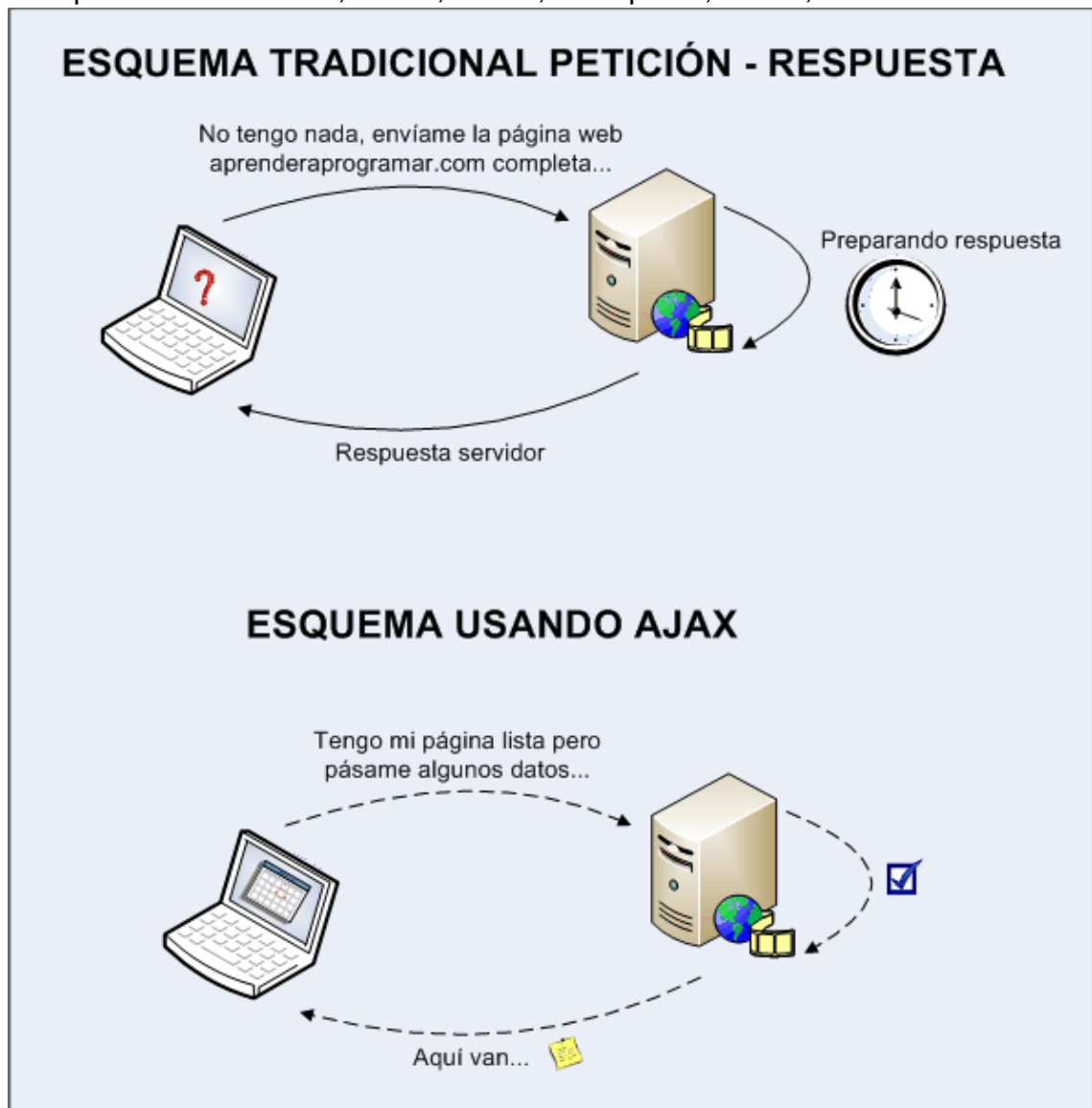


Ilustración 7: Esquema de funcionamiento de AJAX

3.3. Infraestructura

A pesar de que el servidor está pensado para funcionar en un equipo de bajas características. Las características mínimas que un equipo requiere para hacer funcionar el servidor sería de:

- RAM: 512MB-1GB
- HDD: >4GB
- Procesador x64 AMD a 2GHz

Durante el desarrollo de la aplicación se va a seguir un esquema de “Servidor de producción + Servidor de desarrollo” usando un equipo personal como servidor de desarrollo y un portátil como el de producción.



Ilustración 8: Servidor desarrollo + producción

El objetivo de esto es simular durante el desarrollo una instalación de alta disponibilidad. Incluso, con un equipo más disponible se podría montar un servidor dedicado de BBDD.



Ilustración 9: Servidores producción web, BBDD, correo + Desarrollo

Otro de los beneficios de desarrollar la aplicación en web: la fácil implementación de sistemas de escalabilidad y balanceo de carga.

3.4. Software

A continuación se va a listar el software usado en el servidor y su versión.

3.4.1. CentOS 7 x86 64-Minimal:

Como sistema operativo: CentOS 7 por su estabilidad y popularidad en aplicaciones web. Se trata de una versión minimalista sin interface gráfica.



Ilustración 10: Logo CentOS

3.4.2. SSH:

Se va a usar un servidor de desarrollo externo, se hará uso de una herramienta SSH como forma de comunicación entre los dos equipos.

3.4.3. Apache:

Como servidor web para dar servicio a una red local. No se abrirá servicio al exterior como medida cautelar ante ataques y porque la aplicación lo no exige.



Ilustración 11: Logo Apache

3.4.4. PHP 5.4.16:

Versión más reciente de PHP que proporciona Apache. PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML.



Ilustración 12: Logo PHP

3.4.5. MySQL:

MySQL, es un sistema de gestión de base de datos relacional o SGBD. Este gestor de base de datos en multihilo y multiusuario, lo que le permite ser utilizado por varias personas al mismo tiempo, e incluso, realizar varias consultas a la vez, lo que lo hace sumamente versátil.



Ilustración 13: Logo: MySQL

3.4.6. Git 8.0.3:

Para realizar el control de versiones se utilizará Git con interface gráfica SmartGit. Hay muchos host de git en línea como por ejemplo GitHub. No obstante, necesito una cuenta de pago para poder crear un repositorio privado. Buscando un poco más se puede encontrar otro host llamado GitLab. Este sí me permite crear repositorios privados en su servidor. Su única desventaja frente a GitHub es una interface un poco más confusa y menos recursos gráficos. Pero mientras la primera no supone un problema notorio, la otra no tiene mayor importancia.



Ilustración 14: Logo SmartGit

3.4.7. XAMPP:

Para el equipo de desarrollo se va a simplificar el uso de software. XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.



Ilustración 15: Logo XAMPP

4. Planificación

Modelo Constructivo de Costos (COCOMO) es un modelo matemático utilizado para la estimación de costes. Dependiendo del nivel de detalle podemos usar una de las siguiente variaciones del modelo: básico, intermedio y detallado.

Para este caso vamos a usar la variación básica. Estimando el resultado final a unas 2000 líneas de código. También podemos encuadrar este proyecto en el modo orgánico pues los conocimientos a priori sobre las tecnologías a usar no son desconocidas y los requerimientos son flexibles.

Usando una calculadora COCOMO obtenemos los siguientes resultados para nuestro proyecto:

SLOC	Modo	Meses requeridos	Numero de personas
2000	Orgánico	4.5	1
2000	Semi-Orgánico	4.8	1
2000	Empotrado	4.9	2

Como ya se ha comentado, podemos encuadrar este proyecto en un proyecto orgánico. Lo que nos lleva a que el tiempo de desarrollo será de aproximadamente 5 meses.

El diagrama de Gantt es una herramienta muy útil en la planificación de un proyecto. En este diagrama, se puede ver facilmente de un vistazo todas las tareas y su duración

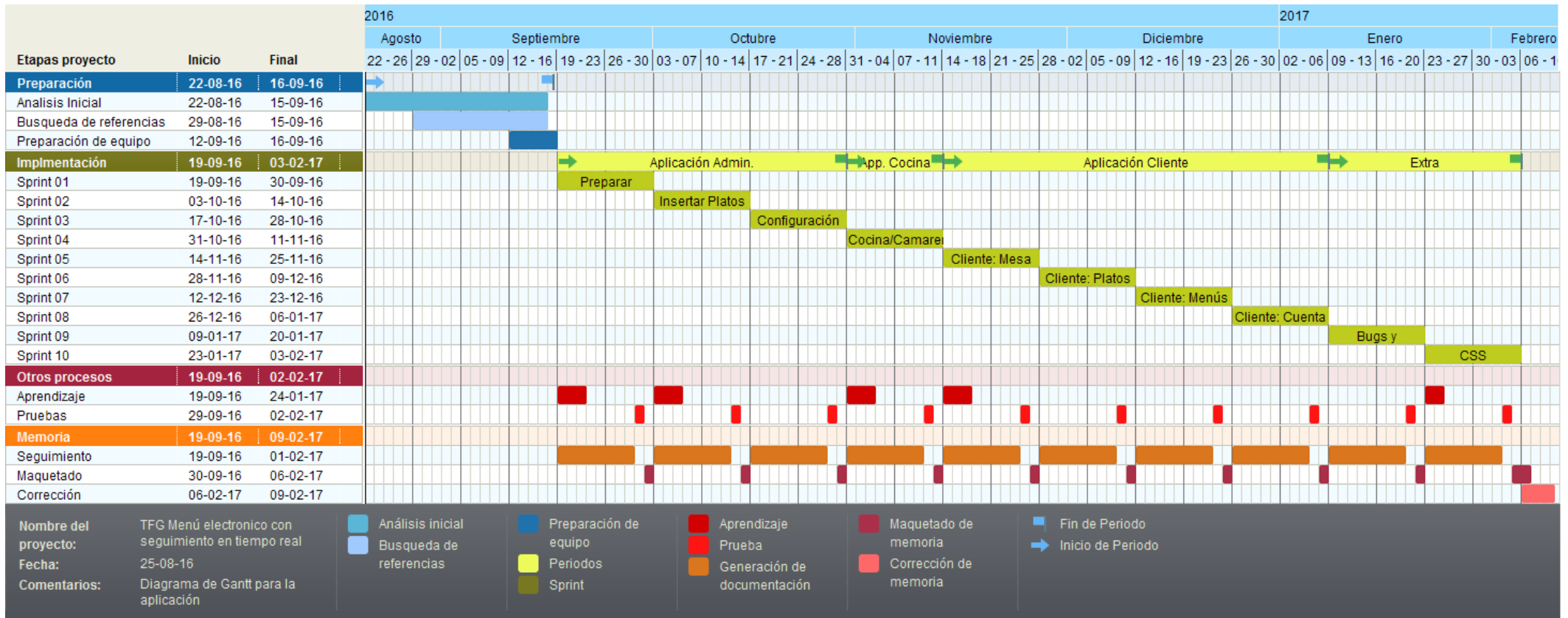


Ilustración 16: Diagrama de Gantt

Con una estimación del tiempo de desarrollo, podemos empezar a hacer un presupuesto estimado donde reflejamos los costes.

Gracias al COCOMO tenemos una idea del tiempo requerido para desarrollar la aplicación. Además en el diagrama de Gantt están más detalladas otras funciones que también requieren de personal extra, como las personas encargadas de realizar pruebas.

Teniendo en cuenta que el sueldo de un graduado medio es de 21.600€ brutos anuales (1800€ mensuales brutos) para un trabajo de 20 horas semanales en medíajornada: toca a 11,25€ la hora. Con estos datos, ya podemos hacer un desglose del presupuesto.

Duración del proyecto: 138 días

Horas por díatrabajado: 4

Total de 6.210€ brutos para pagar al desarrollador.

Suponiendo que nuestro tester sea, también, un graduado en ingeniería informática, este sería su salario:

Díasde pruebas: 10

Horas por díatrabajado 2

Total de 225€ para pagar a un tester para nuestra aplicación

No obstante debido a la naturaleza de nuestra aplicación, las pruebas deben hacerse de forma concurrente. Como mínimo son necesarias 2 personas destinadas a hacer pruebas.

Cocepto	días	horas	Total
Desarrollador	138	4	6.210€
Tester 1	10	2	225€
Tester 2	10	2	225€
		Presupuesto	6.660€

5. Seguimiento de Desarrollo

5.1. Esquema de cada iteración

Ya con una visión global, objetivos y una mecánica de trabajo definida, es el momento de comenzar con el desarrollo de la aplicación. Como ya se ha mencionado en el capítulo 1, se va a usar la metodología de trabajo SCRUM. Por tanto se escogerán una serie de requisitos prioritarios en cada una de las iteraciones, Sprints a partir de ahora, y se implementaran siguiendo el esquema de SCRUM:

- Planificación de Sprint
- Implementación
- Demostración de requisitos completados
- Retrospectiva
- Refinamiento de lista de requisitos y cambios en el proyecto

5.2. Primer Sprint

5.2.1. Planificación de Sprint

- Preparación de servidor:
 - A. Preparar maquina física
 - B. Instalación del SO CentOS
 - C. Adaptar la red
 - D. Instalar apache
 - E. Instalar servidor MySQL
- Base de Datos
 - A. Diseño de BBDD para la cocina
 - B. Preparado de Scrip de tablas
 - C. Preparado de Scrip de datos de prueba
- Servidor Web
 - A. Investigar últimos protocolos HTML5
 - B. Puesta al día con la API de MySQL para PHP
 - C. Crear web de visualización de datos

5.2.2. Implementación

Durante este Sprint, se va a diseñar el menú de editar carta "A.4" del storyboard. Usando herramientas muy prácticas de HTML5 para implementar una aplicación muy similar a lo ya diseñado sin grandes esfuerzos en CSS. Y sobre el CSS: se va a hacer un uso mínimo de este en las primeras iteraciones por resultar de gran esfuerzo y tiempo.

Esta pantalla de edición de carta va a disponer de 3 módulos web:

- "edicionCarta.html" Será la estructura principal su único propósito es contener a los otros dos módulos

- “listaPlatos_editorPlatos.php” será un frame a la izquierda de la página que mostrará por categorías los platos en la BBDD
- “detalle_plato” Es otro frame que se localiza a la derecha. En este formulario se mostrará la información del plato seleccionado y se permitirá la edición de este.

Diseño de la interface:

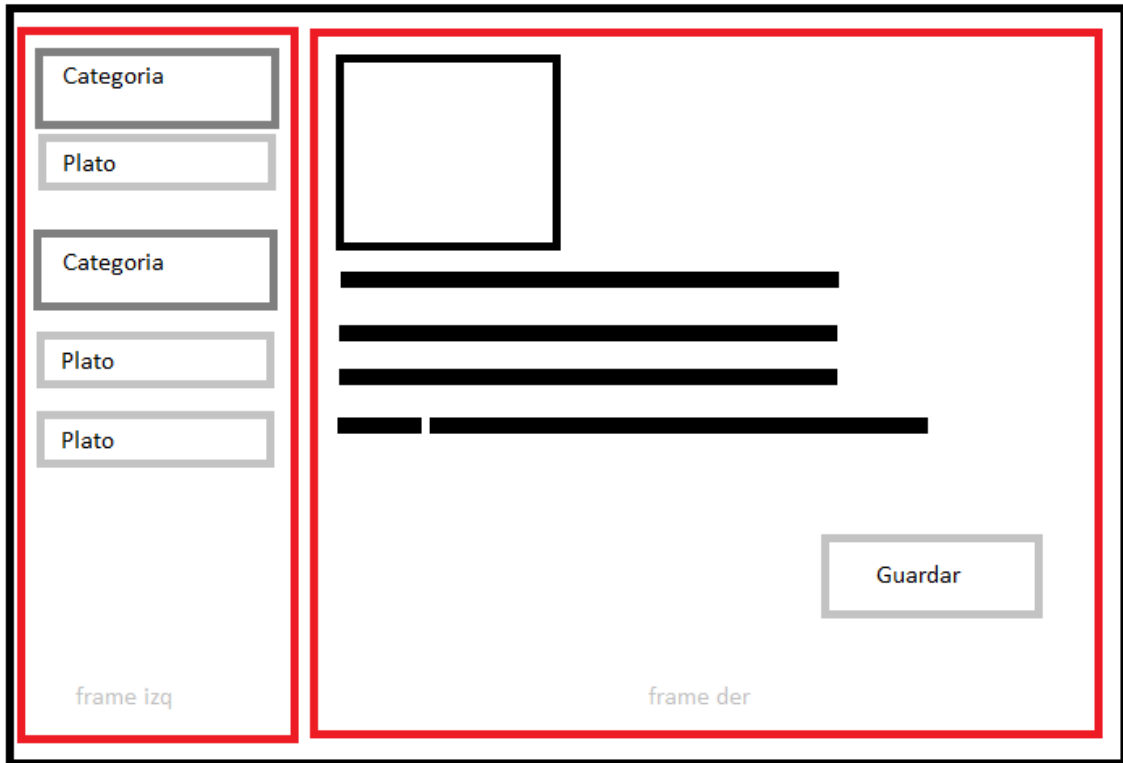


Ilustración 17: Editor de platos

Sobre la BBDD se ha hecho un esquema mínimo sobre los datos necesarios para la implementación de esta primera parte.

Categorías: (id, categoría)

Platos: (id, plato, descripción, foto, precio, categoría, t_preparación)

Detallados de la siguiente forma:

Categorías:

Id	Int 4	Auto primaria
categoría	Varchar (15)	Not null

Platos:

id	Int 4	Auto primaria
Plato	Varchar (50)	Not null
Descripción	Varchar (250)	
Foto	Text	

Precio	Decimal(3,2)	
Categoría	Int(4)	Foránea(categorías)
T_preparacion	Int(3)	

Y un esquema:



Esta primera implementación no ha dado grandes problemas por parte del servidor apache o MySQL. En el anexo sobre el montaje del servidor se da información detallada sobre el proceso de instalación de los diferentes componentes y soluciones a posibles problemas. Este anexo tiene formato de manual, pues se ha pensado que puede ser útil su distribución para hacer al cliente final autosuficiente para la instalación del sistema desde cero con conocimientos mínimos de informática.

En cuanto a la parte web del proyecto, las últimas actualizaciones de las tecnologías php ha cambiado sustancialmente del conocimiento a priori. Esto ha retrasado más de lo esperado el desarrollo de la implementación del primer Sprint.

5.2.3. Demostración de requisitos completados

Se han implementado con éxito los requisitos propuestos para esta implementación. El proceso de aprendizaje ha llevado más tiempo del esperado.

5.2.4. Retrospectiva

Durante el proceso se ha llegado a la conclusión de que la sección de menú, como estaba originalmente propuesta, no debe ser mostrada en la misma sección de edición de carta. El motivo principal es estético: los controles de edición de un menú son drásticamente diferentes al de edición de un plato.

5.2.5. Refinamiento de lista de requisitos y cambios en el proyecto

Después de ser mostrado el final de la iteración, se ha recibido un nuevo requisito no funcional para la aplicación. Para su comodidad, la misma pantalla de edición de plato debe ser capaz de añadir platos nuevos con la adición de un botón adicional. Originalmente esta función se realizaría en una pantalla a parte. Después de ser probado se ha encontrado un bug en la selección de categoría al modificar un plato. Al parecer en algunos casos cambia automáticamente. Resumiendo el resultado de la implementación se ha de cambiar:

- Nuevo requisito:
 - Agregación de platos mediante la ventana de modificación

- Bugs:
-Problemas con la selección de categoría

5.3. Segundo Sprint

5.3.1. Planificación

- Cambios del Sprint anterior:
 - A. Añadir botón añadir plato en la ventana de edición de platos
 - B. Resolver bugs
- Ventana Menú
 - A. Diseñar ventana Edición de menú
 - B. Diseñar la nueva estructura de la base de datos
 - C. Preparar y ejecutar los scrips MySQL
 - D. Implementar la ventana de edición de menú

5.3.2. Implementación

Ahora se ha de realizar un cambio sobre la anterior iteración. La dificultad de realizar este cambio es mínima y se reduce a la inserción de una nueva instrucción en php con su respectivo botón.

En cuanto al bug de categoría: No se ha localizado donde estaba el error que lo producía. Pero al reescribir el código se solventó el problema.

Para la ventana de menús se va a usar el mismo esquema visual y de estructura de archivos que se usó para la ventana de edición de platos. Las diferencias van a ser que en el frame izquierdo, los menús no se van a dividir por categorías. Además, el frame derecho va a estar adaptado a las necesidades del menú.

La base de datos tiene que añadir una nueva tabla (menús) que ya se contempló en la fase anterior. Pero ahora, que la idea esta más pulida y se ha recibido feedback de la iteración anterior, se ha decidido que el menú va a estar compuesto por únicamente 3 platos. Estos platos podrán ser escogidos de las diferentes categorías.

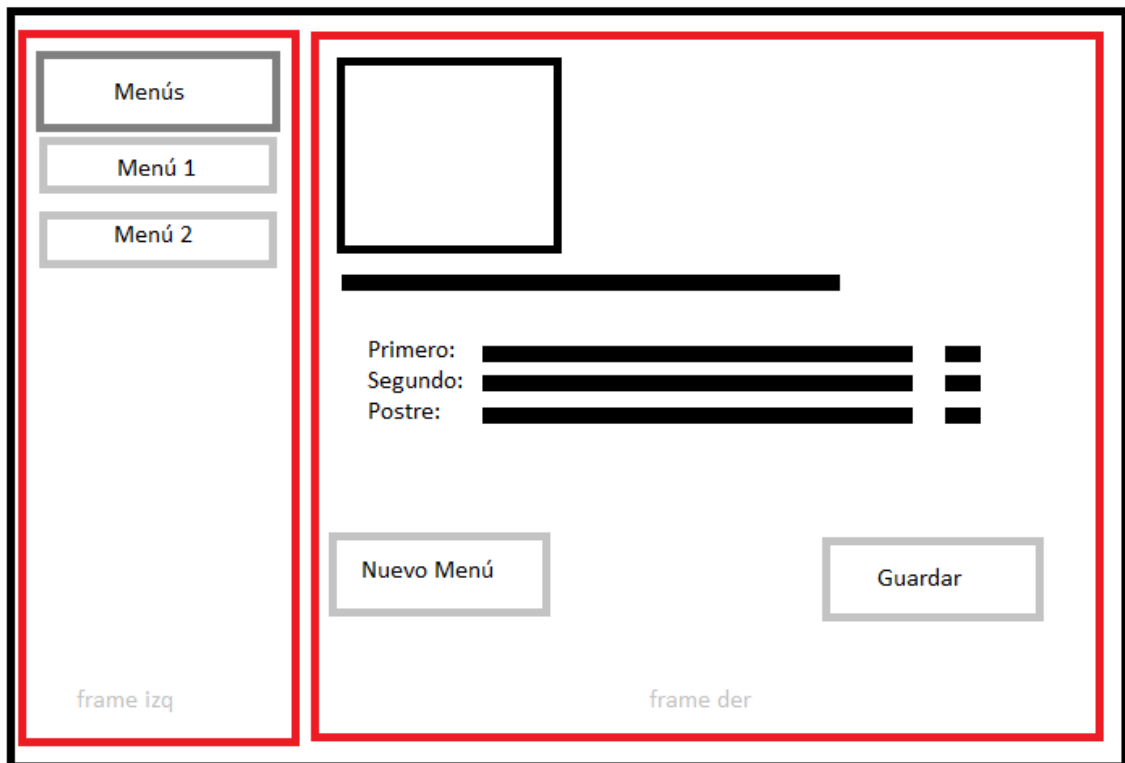


Ilustración 18: Editor Menús



Con base de datos detallada de la siguiente forma:

Menús:

Id	Int (4)	Auto primaria
Descripción	Varchar (250)	
foto	Text	
Plato1	Int (4)	Not null, foránea (platos)
Plato2	Int (4)	Not null, foránea (platos)
Plato3	Int (4)	Not null, foránea (platos)
Precio	Decimal(3,2)	
T_preparacion	Int(3)	

5.3.3. Demostración de requisitos completados

En esta nueva iteración se han cumplido con todos los puntos propuestos en la sección 5.3.1 de planificación. Además, por no necesitar un periodo de aprendizaje, se han realizado antes de lo esperado.

- Solucionado bug de categorías

- Añadido botón de añadir plato
- Implementada ventana de menús tanto para añadir como para modificar nuevos menús

5.3.4. Retrospectiva

La anterior iteración ha servido de gran ayuda para consolidar el producto final que se desea entregar y está prácticamente listo en su versión final a pesar de que aún queda la aplicación de las hojas de estilo CSS. También se ha recibido información acerca de las funcionalidades de los parámetros: se quiere que los platos tengan un parámetro booleano que indique si el plato está disponible o no. Esta nueva funcionalidad se va a extender a los menús a partir de la próxima iteración.

5.3.5. Refinamiento de la lista de requisitos y cambios en el proyecto

Se va a añadir un nuevo parámetro en la bbdd en las tablas platos y menús. Este parámetro es “disponibilidad”

5.4. Tercer Sprint

5.4.1. Planificación

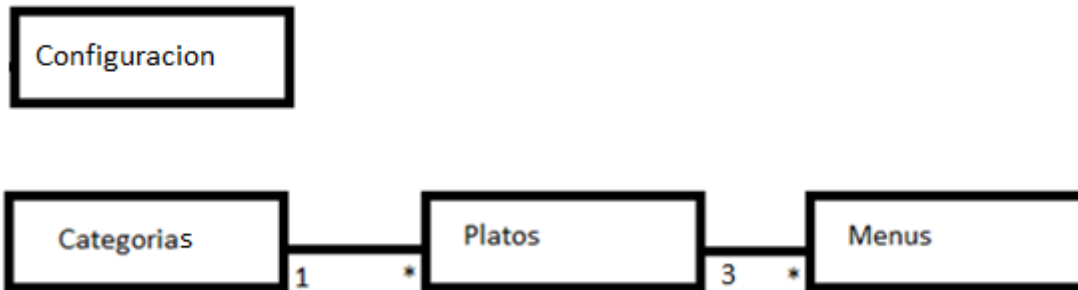
Una vez más se va a editar la BBDD para añadir un parámetro más. Este parámetro es el de disponibilidad de platos/menús. En esta iteración se planea, además, de terminar con la parte del sitio dedicado a la administración interna diseñando e implementando la ventana de configuración global.

Resumiendo, los objetivos serán:

- BBDD:
 - Añadir parámetro “disponible” a las tablas menús y platos
 - Estructurar la nueva tabla de Configuración.
- Web:
 - Diseño ventana de configuración
 - Implementación de la ventana de configuración

5.4.2. Implementación

Comparada con las 2 anteriores implementaciones, esta, no resulta ningún problema de complejidad. La nueva tabla que se va a añadir no está relacionada con el resto. Quizás, en un futuro, lo esté con otra tabla de idiomas disponibles. Pero, para nuestra primera versión final, no será necesario debido al reducido número de opciones. A sí mismo la tabla de configuración pudiese ser un archivo independiente. Pero con objeto de tener todo centralizado, y ya que se dispone del recurso, se implementará en la base de datos



Las tablas quedarán con la siguiente estructura:

platos:

id	Int 4	Auto primaria
Plato	Varchar (50)	Not null
Descripción	Varchar (250)	
Foto	Text	
Precio	Decimal(3,2)	
Categoría	Int(4)	Foránea(categorías)
T_preparacion	Int(3)	
Disponible	Boolean	

menús:

Id	Int (4)	Auto primaria
Descripción	Varchar (250)	
foto	Text	
Plato1	Int (4)	Not null, foránea (platos)
Plato2	Int (4)	Not null, foránea (platos)
Plato3	Int (4)	Not null, foránea (platos)
Precio	Decimal(3,2)	
Disponible	Boolean	

configuracion:

nombre	Varchar(250)	Not null
logo	text	
mostrar_tiempo	Boolean	
pin_cocina	Int(4)	
pin_camarero	Int(4)	
pin_admin	Int(4)	

Véase que hay 3 parámetros “pin”. Estos se usaran para dar acceso a las diferentes secciones en caso de que por motivos de seguridad así se desee.

The image shows a wireframe of a configuration page. It is enclosed in a rectangular border. At the top left, there is a small box labeled "Logo". Below this, on the left side, is a larger square box labeled "Logo". To the right of this box are two smaller boxes: a solid black rectangle and an empty rectangular box. The main content area on the right contains a list of settings:

- Restaurante: followed by a horizontal input field.
- Idioma por defecto: followed by a horizontal input field with a small grey square on the right side.
- Mostrar tiempos: followed by a small square checkbox.
- Pin Cocina: followed by a horizontal input field.
- Pin Camarero: followed by a horizontal input field.
- Pin Administracion: followed by a horizontal input field.

At the bottom right of the main content area is a button labeled "Guardar".

Ilustración 19: Diseño Configuración

Debido a su simpleza, se decide construir esta página con una hoja de estilos CSS que respeta la filosofía relacionada con la interacción persona ordenador. Esta forma de actuar se va a implementar en todas las páginas y abarca la distribución de los elementos en pantalla, márgenes, código adaptado a la accesibilidad (en iteraciones anteriores se mostró como cumplía con las normas del Consorcio WWW) y usando la paleta de colores web safe

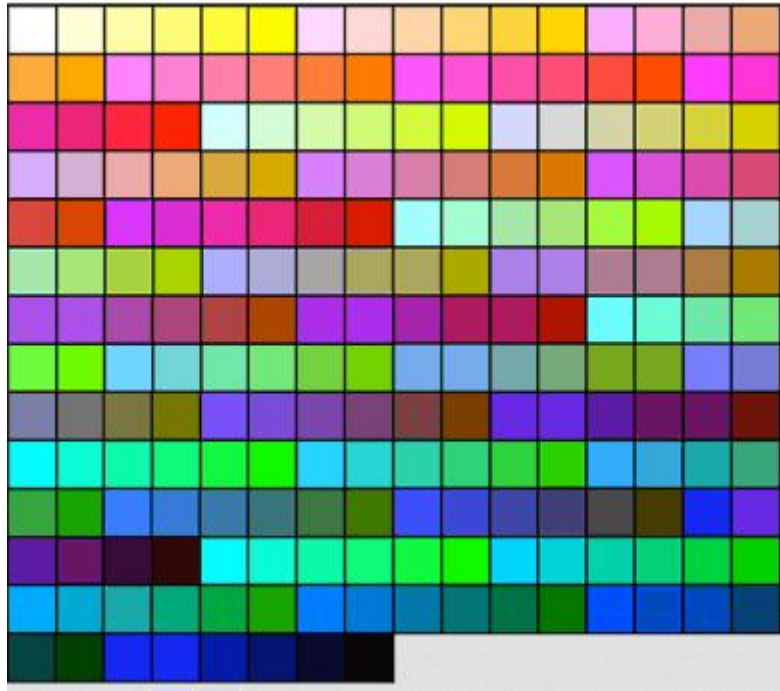


Ilustración 20: Paleta websafe

5.4.3. Demostración de requisitos completados

Como era de esperar se han cumplido todos los puntos de la iteración con éxito.

- Actualizadas bases de datos
- Diseñado e implementación de la nueva web dedicada a la configuración
- Añadida hoja de estilos al diseño web

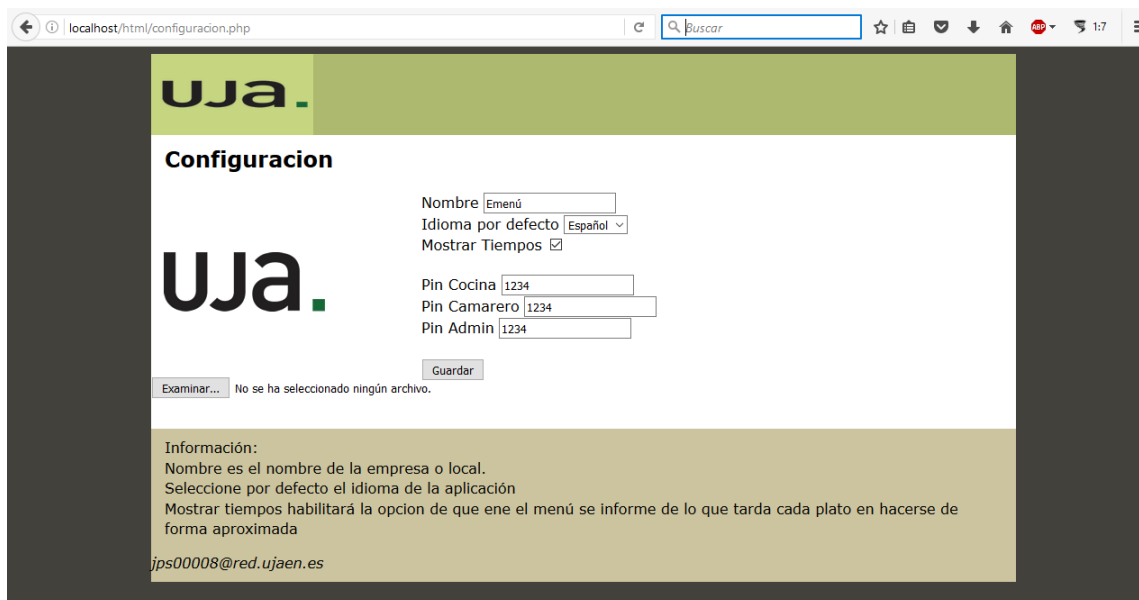


Ilustración 21: Captura configuración

5.4.4. Retrospectiva

Se ha dado un feedback positivo con respecto a las anteriores versiones sin comentarios sobre cambios. El proyecto sigue a delante sin más inconvenientes y siguiendo el método de trabajo actual.

5.4.5. Refinamiento de requisitos y cambios en el proyecto

En esta iteración no han surgido nuevos cambios o imprevistos que sugieran realizar cambios sobre el trabajo ya realizado o sobre futuras implementaciones.

5.5. Cuarto Sprint

5.5.1. Planificación

Se va a abordar la aplicación que se usará en la cocina. Esta aplicación también va a sufrir un cambio importante desde su concepción en el storyboard original. Después de ser hablado, se ha llegado a la conclusión de que hay que separar esta aplicación en dos; una para la cocina y otra para los camareros.

Así definidos, los objetivos de esta iteración serán:

- BBDD:
 - Añadir tabla de comandas
- Web:
 - Diseño ventana de cocina
 - Implementar un sistema de drag & drop
 - Implementar una conexión asíncrona con la BBDD y la web de la cocina.
 - Crear una hoja de estilos CSS mínima que asegure el funcionamiento de la web

5.5.2. Implementación

En la aplicación de la cocina, se hará gestión de los platos que están pendientes de ser preparados, los que están siendo preparados y los que se han terminado. En la aplicación para los camareros se mostraran avisos de las mesas que han solicitado un camarero. Originalmente las dos funciones se iban a mostrar en la misma aplicación de la cocina.

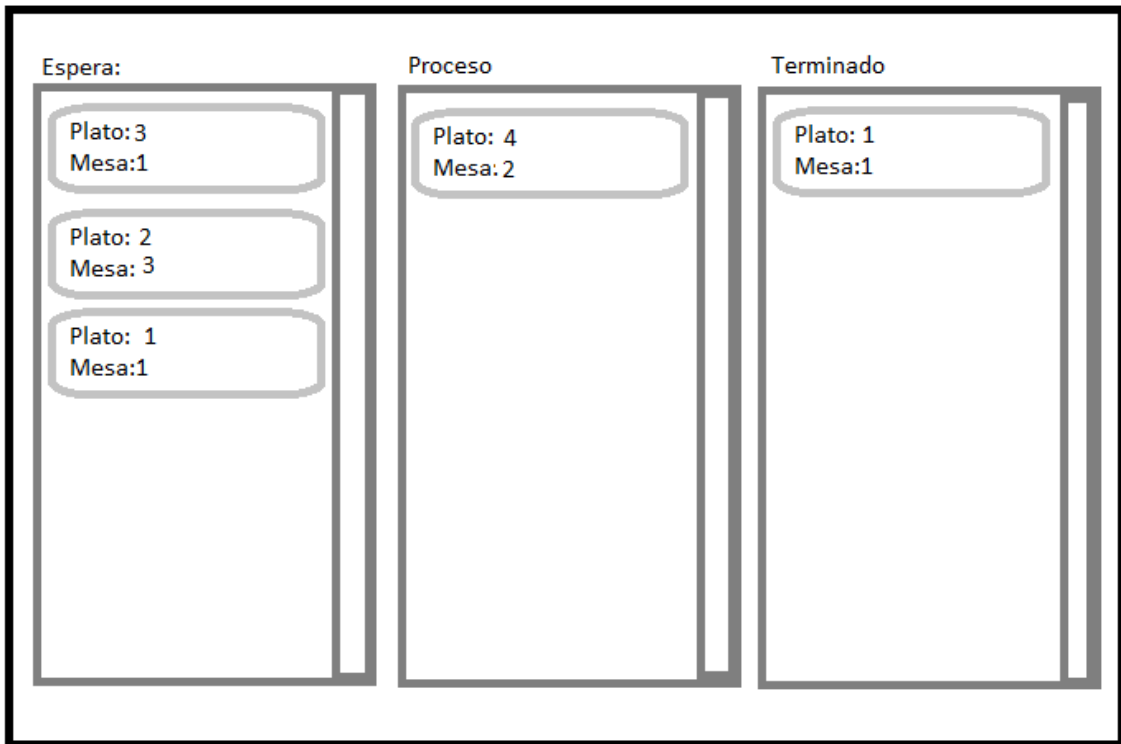


Ilustración 22: Esquema cocina

Para dar comodidad al diseño se va a implementar un sistema de draga & drop para los elementos.

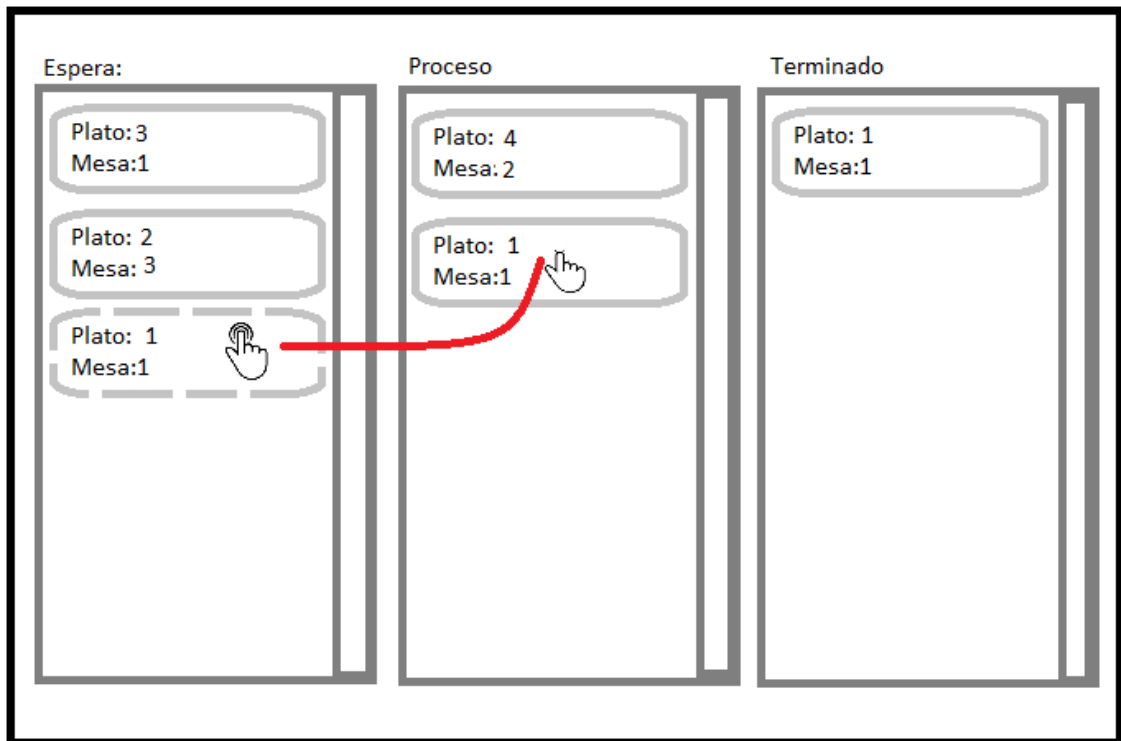


Ilustración 23: Funcionamiento Cocina

Haciendo la actualización en la base de datos automática. En esta web los procesos son más complicados, por lo que una estructura lineal para la

realización de operaciones dará problemas en el futuro si se quieren aplicar cambios. Se quiere tener constancia del contenido de la base de datos en todo momento y sin necesidad de recargar la página. Para conseguir esto, se va a utilizar AJAX como puente de conexión con la base de datos.

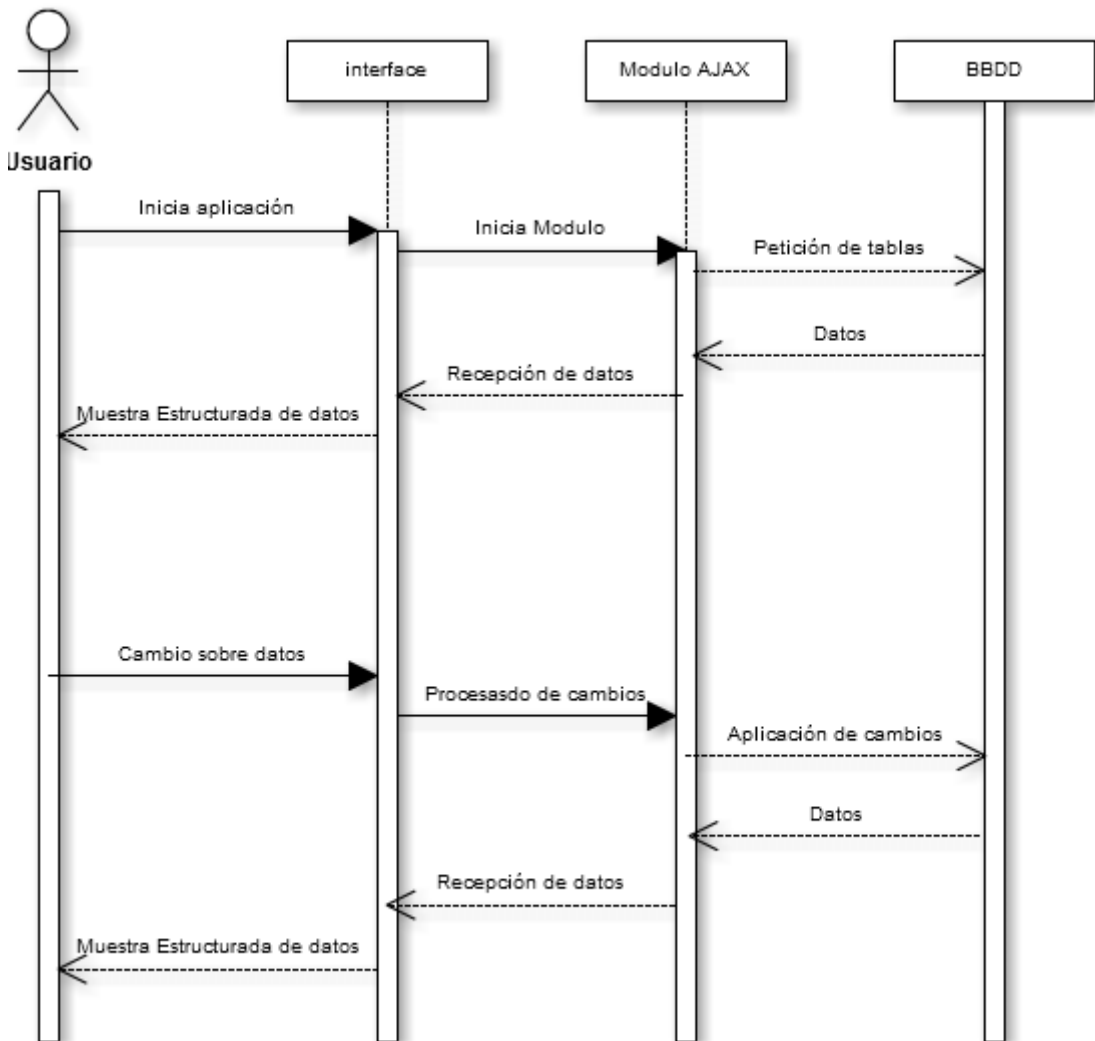
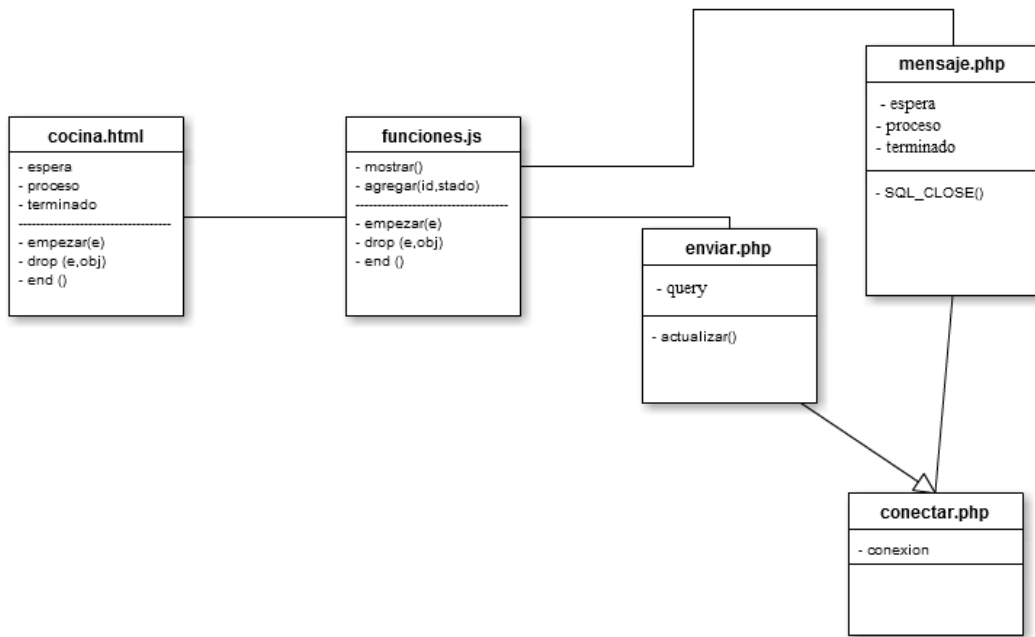
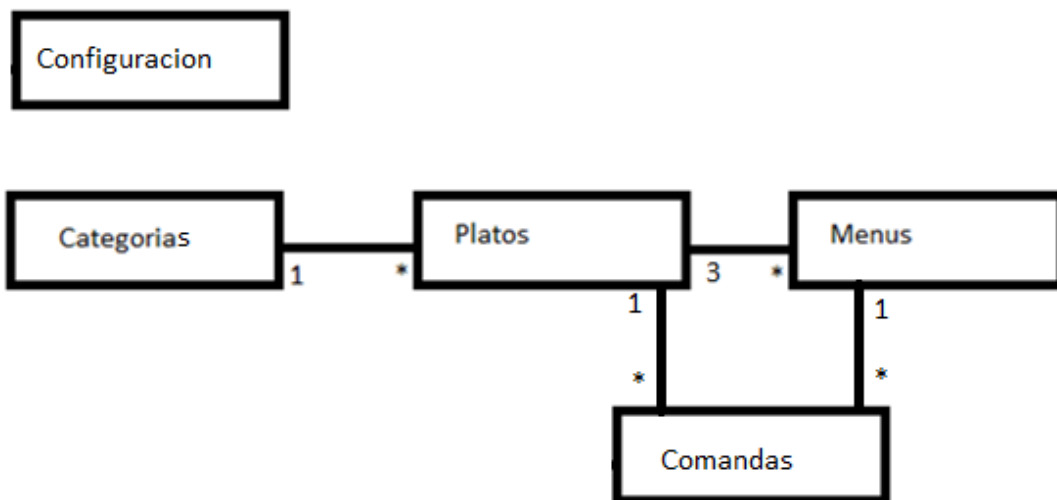


Ilustración 24: ciclo de vida de la conexión

Así mismo la interface (que es el archivo web) dispone en JavaScript de funciones que se van a utilizar en el futuro.



También hay que añadir una nueva tabla para la base de datos con la información necesaria sobre las comandas.



comandas:

mesa	Int(4)	Primaria
Plato	Int(4)	Primaria, foránea(platos)
Fecha	Timestamp	Primaria, dafault(actual)
Estado	Enum(espera, proceso,	Default(espera)

	terminado	
Menú	Int(4)	Foránea(menus)

5.5.3. Demostración de requisitos completados

Esta ha sido la iteración más dura desde que comenzó el proyecto. No obstante se han alcanzado de forma exitosa todas las metas propuestas para la iteración. En algún momento se perdió el estado de una de las comandas pero es un error que no se ha podido reproducir de nuevo.

5.5.4. Retrospectiva

Durante esta iteración ha habido un cambio importante: se ha visto lo poco práctico que resultaba la clave primaria de la tabla de comandas y se ha actuado en consecuencia. Más detalles en el próximo apartado.

5.5.5. Refinamiento de la lista de requisitos y cambios en el proyecto

Se ha añadido un nuevo requisito a la lista:

Los camareros tendrán su propia página para atender las peticiones de las mesas.

Por motivos de claridad en el código de la web, se ha decidido hacer un cambio significativo en la estructura de la tabla de comandas en la base de datos. Era muy engorroso obtener tantos parámetros como claves tenía la tabla y se ha añadido un identificador autoincremental a las nuevas comandas. Este parámetro id se considera ahora clave única y la estructura actual de la base de datos tiene, ahora, este aspecto:

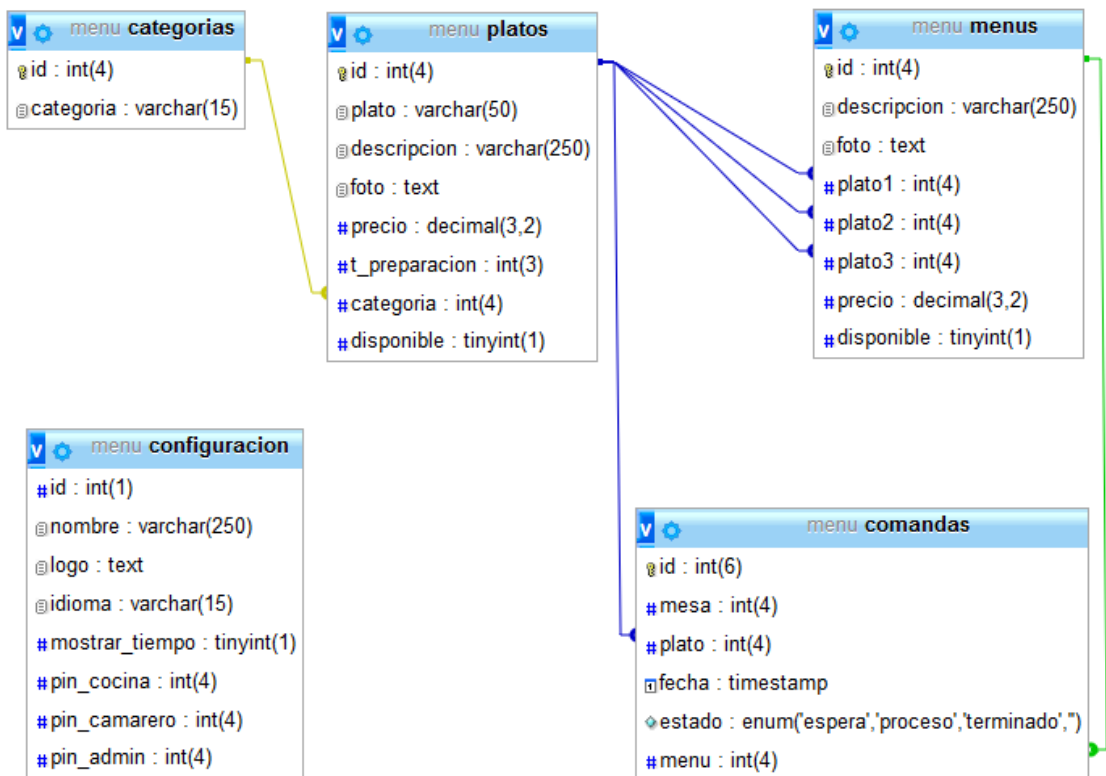


Ilustración 25: Diagrama Base de datos

5.6. Cuarto Sprint (extendido)

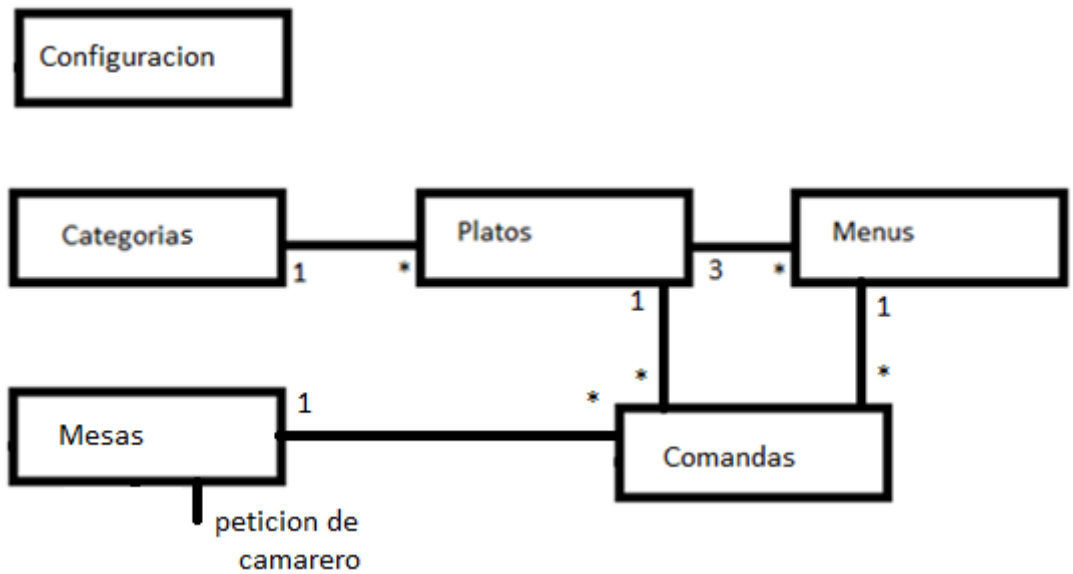
5.6.1. Planificación

En esta iteración se van a realizar trabajos en los que se va a ajustar el ritmo de desarrollo. Ha habido cambios a lo largo del proceso que han dejado “cojo” a la visión global de la aplicación. Esta no es una iteración provista y el labor que se va a hacer aquí se pudiera haber repartido entre las iteraciones venideras. Pero debido a el tiempo sobrante del resto de iteraciones, se va a utilizar este espacio de tiempo dedicado a:

- BBDD:
 - Adaptar la nueva tabla de mesas
 - Estructurar una nueva tabla para atender las peticiones de camareros
- Web:
 - Diseño ventana de camareros
 - Diseñar e implementar una Web de navegación entre los diferentes menús de administración

5.6.2. Implementación

Ahora, las mesas cobran algo más de importancia en la aplicación. Hasta el momento del diseño en las comandas, bastaba con numerarlas, pero ahora hace falta tener un registro de estas para conocer cuales están pidiendo un camarero.



Dejando una estructura así:

Mesas:

id	Int(4)	Auto Primaria
nombre	Varchar(50)	
camarero	Boolean	Default(false)

Comandas:

id	Int(4)	Auto Primaria
mesa	Int(4)	Foránea(mesa)
Plato	Int(4)	Foránea(platos)
Fecha	Timestamp	Dafault(actual)
Estado	Enum(espera, proceso, terminado)	Default(espera)
Menú	Int(4)	Foránea(menus)

Para la aplicación del camarero, se ha pensado en ella como exclusivamente móvil aunque pudiera ser ejecutada en una pantalla de mayor tamaño dispuesta de forma vertical. Se va a reutilizar en mayor medida la web de comandas pues su lógica es la misma. Su mayor diferencia que la aplicación de camarero es meramente informativa y no requiere de los metodos de drag & drop. En su lugar se pondrá un botón cuya función es de eliminar la alerta.

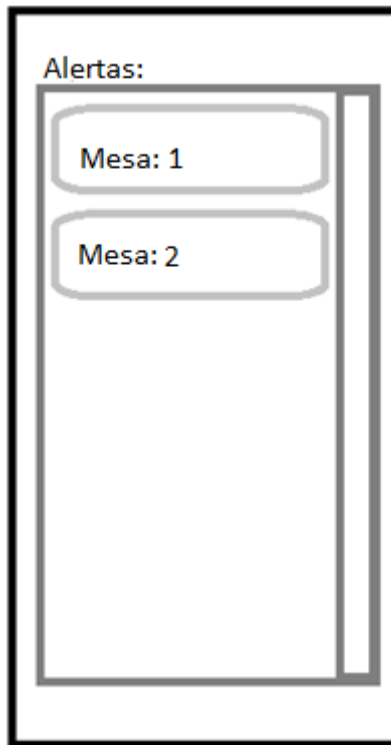


Ilustración 26: Esquema alertas camareros

También queda pendiente una web-menú de navegación para agrupar las diferentes webs de administración. Pues va a ser una web plana se va a llevar a su versión final en el momento. También se quiere recordar que las decisiones de diseño que no se explican están basadas en recomendaciones de los libros de la Asociación Interacción Persona-Ordenador (**AIPO**) y con una estructura que valida el w3c.

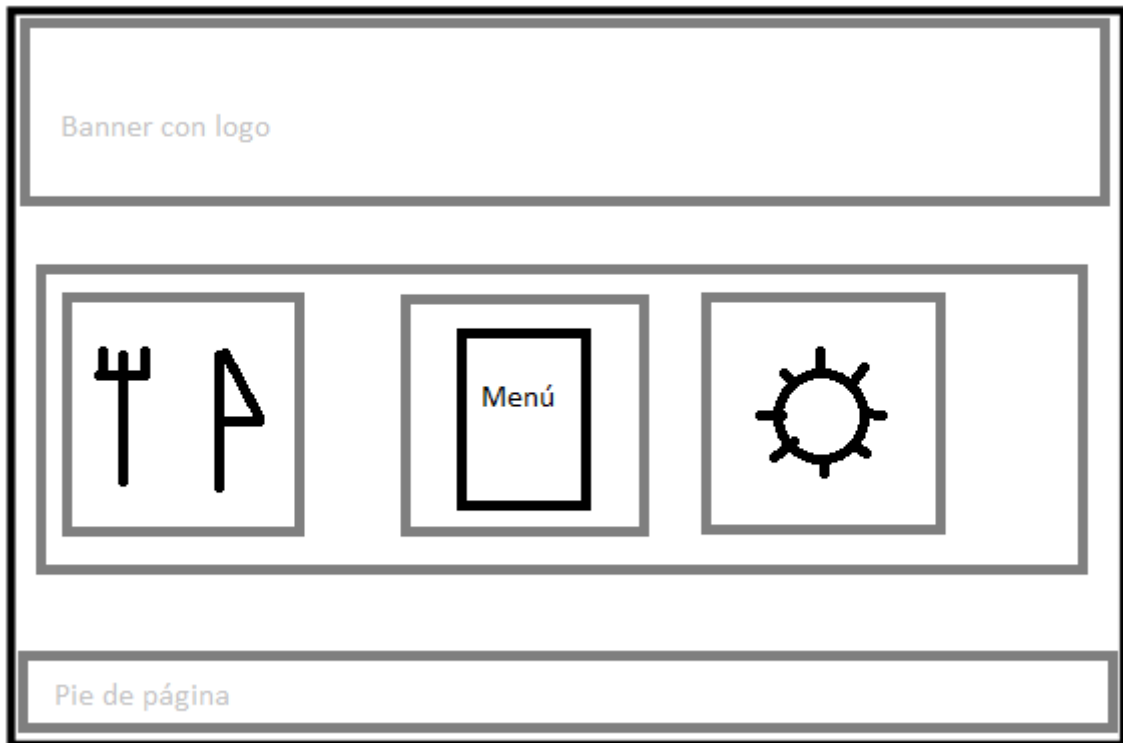


Ilustración 27: Esquema menú administración

5.6.3. Demostración de requisitos completados

Los cambios en las tablas ha sido un asunto trivial, en las tablas antiguas no ha hecho falta realizar ningún cambio mas que una nueva dependencia de tablas. La web de camareros.html ha sido reutilizada mayormente y apenas ha habido cambios sobre los archivos de funciones y, aunque falta hacer un diseño acorde de css el resultado se parece mucho al producto final:

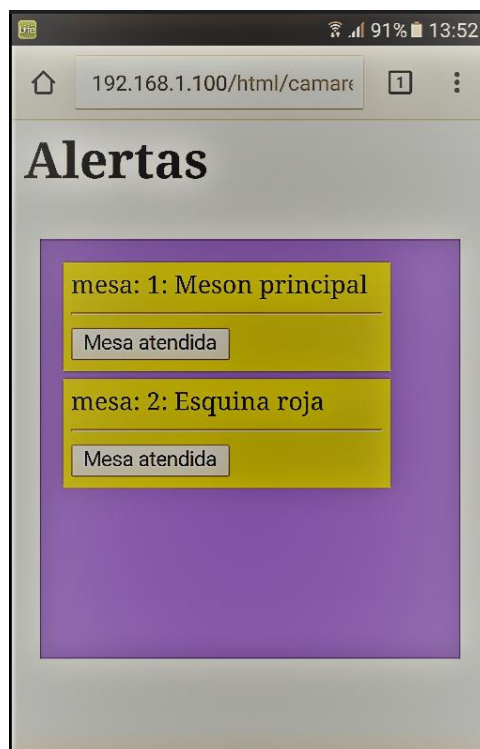


Ilustración 28: Captura de alertas

Y no hay mucho que mencionar sobre la pantalla web de unión entre las paginas de platos, menús y configuración. También comparte el mismo documento de estilo css que la web de configuración para darle cohesión. Que el usuario comprenda que está usando la misma aplicación a pesar de estar en otra pantalla.



Ilustración 29: Captura menú de administración

5.6.4. Retrospectiva

Adicionalmente se ha añadido una nueva característica a la aplicación durante esta iteración, se ha añadido un atributo

nombre a la tabla de mesas. Esto es así porque a algunos restaurantes, se les ponen sobrenombres a las mesas como forma de llamar a los comensales. Unos ejemplos de esta practica son la cadena 100 montaditos, Starbucks o Machitos

5.6.5. Refinamiento de la lista de requisitos y cambios en el proyecto

Se ha añadido el nuevo termino “nombre” a la tabla de mesas además de haberse separado finalmente las webs de cocina y de camareros. No hay más cambios y el proyecto sigue como estaba provisto. En la próxima iteración se empezara a hacer las webs relacionadas con el cliente, el comensal.

5.7. Quinto Sprint

5.7.1. Planificación

Se esta llegando a la recta final del proceso de desarrollo de la aplicación y se inicia el diseño de las webs cliente. Despues de esto, solo quedará la implementación del sistema de login y creación de hojas de estilo CSS.

En esta implementación se va a:

- Diseñar y crear una página esqueleto que soporta los menús y sub menús de la aplicación cliente
 - Esta pagina cuenta con un baner de estado
 - El banner de estado esta siempre visible
 - Una sección con los 3 posibles estados de la aplicación (Comandas, platos y factura)
- Sección con los 3 posibles estados de la aplicación
 - Los estados son enlaces a las paginas que referencia a las comandas (platos pedidos), Platos (donde se realizan los pedidos) y factura (donde se detalla la factura)
- Diseño e implementación de la página de comandas para el cliente

Esta iteración parece más cargada de lo normal, en cierto modo, así es. No obstante se van a reutilizar archivos de otras paginas ya implementadas. De hecho, la única pagina que no se ha modularizado ha sido la de edición de menú. En futuras iteraciones, más orientadas al pulimento y optimización del código, puede que esto cambie y se adapte también el modulo de conexión a la base de datos.

5.7.2. Implementación

Para esta parte de la aplicación se va a construir una web recipiente de frames. Originariamente, en el storyboard, se pensó en esta aplicación como una aplicación móvil. Pero, en caso de que el establecimiento cuente con su propia infraestructura, es más cómodo que se usen tabletas. Es por esto por lo que se opta por un diseño más panorámico.



Ilustración 30: Concepto diseño de aplicación de cliente

Los elementos que constituyen el contenedor van a ser un banner siempre visible desde cualquier menú. Este banner va a tener el logo del restaurante, el nombre de la mesa y/o numero de esta y por ultimo un calculo del importe total de los platos pedidos. Más abajo se encuentra la lista de las secciones. Este menú de navegación, debido a su simpleza y poca personalización, se mantiene como una pagina estática.

Por último el fuerte de esta iteración: una web donde se mostrarán todas las comandas. Muestra el plato pedido, una foto, el estado en la comanda y un botón de eliminar comanda. Este botón solo está disponible en caso de que el pedido esté en estado de “espera”. Esta web también tiene un botón dedicado a hacer una petición a un camarero.

5.7.3. Demostración de requisitos completados

Se han completado mayormente todos los requisitos. Falta en la web de “mesa.php” terminar la estructura del listado de platos tal como se ha mostrado en los diseños previos. Esto no se ha terminado de completar por falta de tiempo al finalizar la implementación en conjunto con la falta de recursos gráficos. Además, se ha decidido retrasar este aspecto por la necesidad de usar un estilo css que de más cohesión a la aplicación. Añadir la estructura y el parámetro foto ahora, sin el orden que proporciona la hoja de estilos, hace más pesado y antiestético el uso de la aplicación para sus pruebas de funcionamiento que si sigue siendo la prioridad a estas alturas de el desarrollo.

5.7.4. Retrospectiva

La retroalimentación con los encargados de probar la aplicación sigue siendo satisfactoria. Pero desde el punto de vista del desarrollo empieza a haber problemas con el uso de los “frameset”, más concretamente con la comunicación entre estos.

5.7.5. Refinamiento de la lista de requisitos y cambios en el proyecto

Con el fin de mejorar la claridad del código y solventar problemas de comunicación en las paginas, se va a adelantar el uso de las variables sesión. En vez de esperar a empezar a crear pantallas de login, se van a empezar a usar ahora como sustituto a pasar las variables mediante “POST & GET”

5.8. Sexto Sprint

5.8.1. Planificación

Durante esta iteración se va a implementar la pagina para hacer pedidos. De momento se han usado inyecciones sql para hacer pruebas con la aplicación. Ya empezaba a urgir un método más sencillo de editar la tabla de comandas. Así mismo Esta aplicación se va a separar en 2 partes debido a que funcionan de forma diferente. Una para añadir platos normales, otra para añadir set de platos (menús).

- Diseñar y crear una página esqueleto cliente que permita añadir platos a una comanda
- Permitir la adición de comandas
- Añadir un sub menú de la pagina cliente que separe las categorías.

5.8.2. Implementación

Como ya se dijo en la iteración anterior, se va a empezar a usar variables de sesión para la aplicación de cliente. De momento solo ha sido un problema el conocimiento de la mesa desde la que se están haciendo las operaciones, haciendo el código bastante sucio. Por esto se va a empezar a usar a partir de ahora, y en el futuro, sobre paginas anteriores, una nueva variable de sesion llamada “_mesa”

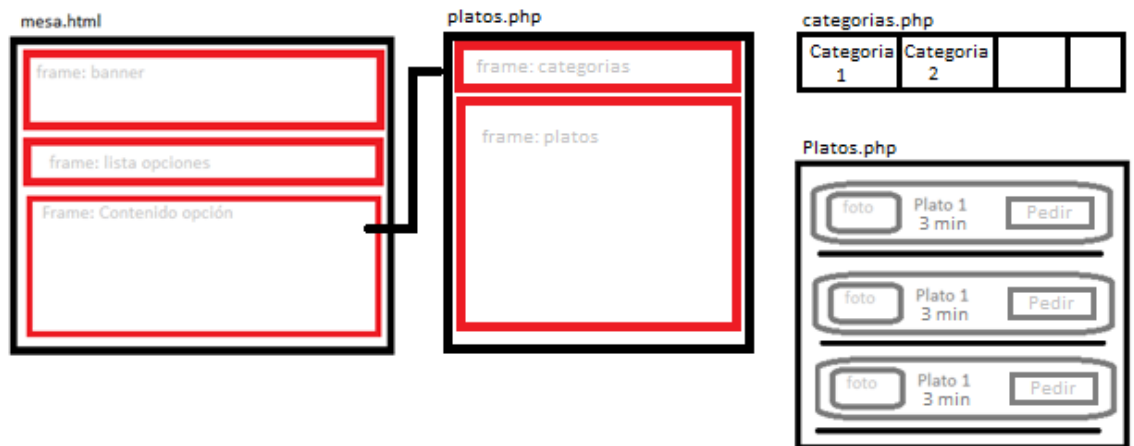


Ilustración 31: Uso de frameset en aplicación cliente

La estructura de esta web va a ser similar a la de cliente. Un menú vertical con las diferentes categorías seguido de la lista de platos que corresponda a la categoría seleccionada. Cada una de esta parte montada en un frameset. En las categorías aparecerá en futuras implementaciones una opción más que será la de menús.

5.8.3. Demostración de requisitos completados

Una vez más se han cumplido todos los requisitos propuestos al comienzo de la iteración. Además, por falta de su correcta previsión se ha añadido una pagina más que inicializa Las variables de sesión de las que se habló, pero quedaron no reflejadas en el diseño. Esta pagina es simple, de carácter únicamente provisional y consta solo de un formulario que pide la introducción del número de la mesa.

5.8.4. Retrospectiva

En esta iteración se ha ampliado la plantilla de betatesters. Los resultados ha sido inmediatos con una gran cantidad de sugerencias e ideas para implementar en futuras implementaciones.

5.8.5. Refinamiento de la lista de requisitos y cambios en el proyecto

Por error se llamó en el diseño “platos.php” tanto al contenedor de frames como a la web encargada de listar los mismos. Esta web que se encarga de listar los platos pasa a llamarse ahora “catalogo.php”

Gracias a los nuevos test se ha localizado un par de errores en la aplicación:

- No aparece el precio de los platos en el catalogo.
- El identificador de plato resulta confuso en la visualización de comandas.
- En la cocina pueden desaparecer las comandas si se arrastran al área de otra comanda.

- En la nueva web de introducción a la mesa, falla la primera entrada.
También se ha sugerido que entren nuevas características:
- Botones de pasar de estado a otro en la cocina

5.9. Séptimo Sprint

5.9.1. Planificación

En esta iteración se va a continuar con la pagina de catalogo de platos. Más concretamente se va a terminar de implemetar la función de adición de menús.

Respecto a los cambios planeados en la anterior iteración, se va solventar el problema de la aparición de los precios en los platos.

5.9.2. Implementación

La pantalla de adición de “menús” pertenece al mismo frame que el de platos. Y su acceso se realiza mediante la sección de categorías. El motivo por el que se ha separado del desarrollo normal: su estética. Ciertamente se asemeja mucho al de selección de platos; así debe ser, a fin de cuentas se encuentra en el mismo ámbito de función. Pero así mismo es obvio que es diferente por el número de elementos en pantalla.



Ilustración 32: Diseño de catálogo de menús

Así mismo, también se incorporará un nuevo enlace en la barra de categorías que enlaza con esta página nueva y, por último, se cambiará el enlace por defecto el frame de platos.php al de catalogoM.php.

5.9.3. Demostración de requisitos completados

Terminados todos los requisitos propuestos para la iteración, queda mencionar que esta página cadece de ningún CSS. El CSS

usado de forma provisional queda excesivamente mal. También falta por añadir espacio para la foto. Estas cuestiones se terminarán de agregar cuando se comience a hacer una hoja de estilos CSS para la aplicación

5.9.4. Retrospectiva

Ahora que se ve, por las diferencias estéticas de esta página, va a ser difícil enmarcar todos los estilos en un único CSS. Más tarde se diferenciará el número de estilos total que harán falta. Tampoco hay que olvidar que aunque hagan falta varios archivos CSS para la aplicación, esta, no debe desentonar entre páginas; No hay que perder la homogeneidad que denote que se está usando una misma aplicación en todas sus páginas.

5.9.5. Refinamiento de la lista de requisitos y cambios en el proyecto

Esta iteración no ha dado a pensar en cambios. Queda, aún, terminar de arreglar los problemas de iteraciones anteriores:

- En la cocina pueden desaparecer las comandas si se arrastran al área de otra comanda.
- En la nueva web de introducción a la mesa, falla la primera entrada.
- El mensaje de control al pedir camarero es aún visible

5.10. Octavo Sprint

5.10.1. Planificación

En esta iteración se van a tratar la página de cuenta. La página cuenta.php va a ser una página que no necesita aprovechar las características de la comunicación asíncrona. Además se van a resolver el problema de entrada a la mesa.

5.10.2. Implementación

La dependencia de esta página va a ser únicamente con la clase conect.php. Esta página, además, vendrá con un botón para pedir un camarero por lo que hará uso de las mismas funciones que platos.php. En esta página se busca un diseño austero que deje claro su función. Como un titeck de restaurante normal.

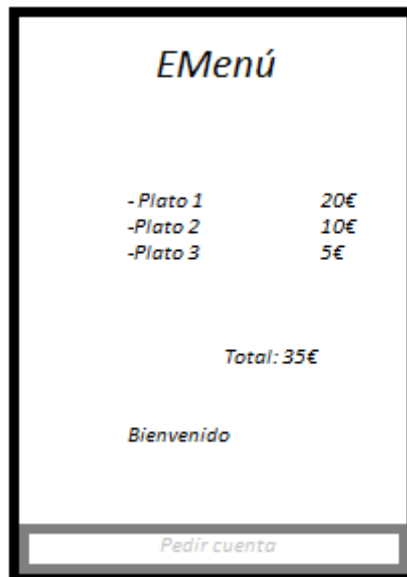


Ilustración 33: Diseño Concepto de cuenta

5.10.3. Demostración de requisitos completados

Llegados a este punto se puede decir que todas las funcionalidades primarias han sido completadas. Se ha arreglado el bug de entrada a la mesa, se ha implementado la página de la cuenta y se le ha añadido un botón de para advertir a un camarero. Además se ha cambiado en color de la fuente en la lista de opciones a negro con la finalidad de que sea más visible.

5.10.4. Retrospectiva

Durante las pruebas con gente nueva del menú de adición de platos y menús, se ha confundido las funciones de los botones guardar y nuevo. Por lo demás la aplicación ha pasado las pruebas necesarias de una forma satisfactoria.

5.10.5. Refinamiento de la lista de los requisitos y cambios en el proyecto

Como se menciona en la retrospectiva, se va a buscar un nuevo nombre para los botones de guardar/nuevo de los platos y los menús. También queda pendiente el resto de los pequeños problemas que no se han solucionado aún.

- Cambio texto botones en el listado de platos/menús
- En la cocina pueden desaparecer las comandas si se arrastran al área de otra comanda.
- La opción por defecto de la carta sea el de menús

5.11. Noveno Sprint

5.11.1. Planificación

Ya están implementadas todas las funciones que resumen el núcleo de la aplicación. Esta iteración se va a centrar en los detalles que quedaron un poco en el aire durante el desarrollo. A continuación se listan los puntos que se van a tratar:

- Cambio texto botones en el listado de platos/menús
- En la cocina pueden desaparecer las comandas si se arrastran al área de otra comanda.
- Mensajes para dar “feedback” al usuario cuando llama a un camarero
- La opción por defecto de la carta sea el de menús
- Buscar forma alternativa de cambiar el estado de las comandas.

5.11.2. Implementación

Empezando por lo fácil, se va a buscar un nuevo nombre para los botones que hay en las páginas de añadir/modificar platos o Menús. Se va a entrevistar en forma de encuesta a las personas que lo han probado para decidir qué set de nombres es más conveniente para comprender mejor su función.

Nombre de botones para adición/modificación de Menús o Platos:

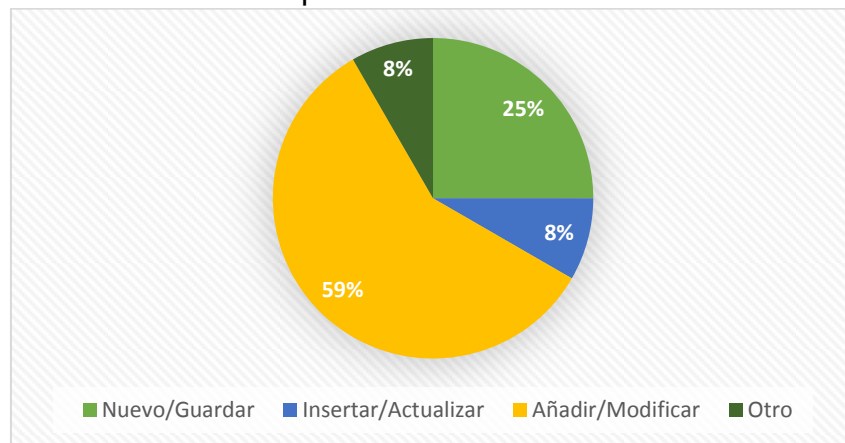


Ilustración 34: Encuesta de cambio en interface

Los resultados de la encuesta sugieren que el texto de los botones sea “Añadir” y “Modificar”. Aunque no parece que un 59% sea motivo suficiente para tomar este dato como válido, el formulario en sí no ha dado a confusión grave mas que en un único caso. Se realizarán los cambios y se realizarán nuevos cambios en caso de que las futuras pruebas empeoren el uso de la aplicación.

La retroalimentación con el usuario cuando pide un camarero se va a conseguir mostrando un mensaje emergente que pregunte al usuario si esta seguro de llamar a un camarero. Muchos testers no han estado seguros de si han llamado al camarero despues de pulsar el botón. Este es el motivo por el que se decide poner este mensaje. También se pondrá el mensaje al pedir la cuenta por los mismos motivos.

El bug de desaparición de comandas cuando se arrastra a lugares bloqueados de la pagina se soluciona fácilmente controlando el identificador de destino.

En el caso de la forma de cambiar el estado de las comandas, se ha pensado en dos opciones nuevas. La primera consiste en colocar un evento onclick para cambiar al siguiente estado. Esta opción se descarta casi de inmediato por presentar dos problemas; El primero referente a que no aparece información a priori sobre que realizar esa acción tenga ese efecto; el segundo a que la comanda tiene que esperar a pasar por todos los estados antes de que vuelva al estado “espera”. Esto nos deja con la otra opción, que se remonta al diseño original: Colocar botones en cada comanda. Un único botón en las comandas en espera que cambie el estado a en proceso. Dos botones en las comandas “en proceso”, uno para pasar a “en espera” y otro para “terminado”. Por último un botón único para las comandas “Terminadas” que cambie el estado a “proceso” . De momento, para evitar una gran cantidad de texto se van a usar los símbolos “←” y “→” como metáfora de su función; aprovechando que, efectivamente, los contenedores donde se agrupan los estados están alineados y son consecutivos.

5.11.3. Demostración de requisitos completados

La mayoría de los objetivos de esta iteración son meramente estéticos. Los problemas de feedback se han solucionado con popups de javascript. La forma alternativa de cambiar el estado de las comandas se realiza con botones y bug de comandas que desaparecen se solucionó fácilmente condicionando las áreas de arrastre.

5.11.4. Retrospectiva

Los cambios realizados ha dado un buenos resultados. Quienes han probado la aplicación han reconocido la mejora en la facilidad de intuir la función de cada control.

5.11.5. Refinamiento de la lista de requisitos y Cambios en el proyecto

Debido al los buenos resultados que han tenido las pruebas hasta el momento, el proyecto, se va a mantener según los últimos cambios. No se ha sugerido la implementación de nuevas características.

5.12. Décimo Sprint (Fin del proceso de desarrollo)

5.12.1. Planificación

Todas las funcionalidades están terminadas, se han arreglado los principales problemas bugs y se ha preparado la estructura de las diferentes paginas siguiendo las normas y consejos de

accesibilidad del w3c. Llegados a este punto, es el momento de preocuparse de la estética de la aplicación. Las decisiones tomadas para la realización de las hojas de estilos están basadas en las recomendaciones de la Asociación Interacción Persona-Ordenador (AIPO); más concretamente en su recomendación en base al “Web Accessibility initiative” (WAI). Anteriormente se habló de la posibilidad de utilizar diferentes hojas de estilo para las diferentes secciones de la aplicación. No obstante, aunque no todas las partes de la aplicación está pensada para el mismo tipo de usuario, se va a utilizar la misma hoja de estilos para todas las secciones de la aplicación con el fin de darle uniformidad a la aplicación como conjunto.

5.12.2. Implementación

Para la realización de la hoja de estilo va a usar como base la “HTML5_thrColFixHdr.css” sugerida por Dreamweaver.

Obviamente, esta hoja de estilos no se ajusta al diseño web que pretendemos conseguir. Por esto, se va a modificar y ajustar al nuestro diseño.

Comenzando con la parte más simple de la web, que desde un inicio ya estaba bastante pulido desde sus primeras versiones, se estructura el CSS para las secciones de administración de la web. Se tratan principalmente de formularios. Para estructurarlos en formato de rejilla, se va a utilizar el siguiente esquema para los elementos de los formularios.

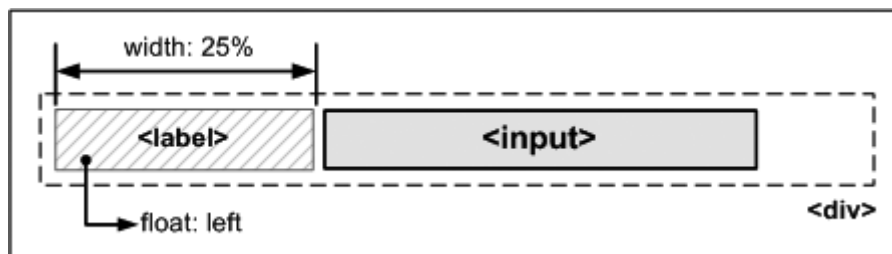


Ilustración 35: Concepto márgenes de alineado en formularios


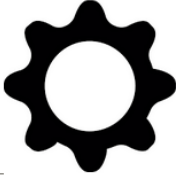
Así mismo, se mantienen las distancias de “padding” y “margin” para evitar problemas futuros en caso de que se adaptase a un sistema multilinguaje.

Menú administrador:

▼ ↻ 🔍 *Buscar*

UJA.

Selección de servicio


Añade o modifica platos en el menú de platos
 Añade un modifica menús en Menús
 Cambia una configuración en Configuración

jps00008@red.ujaen.es

Editor de platos:

← ⓘ localhost/www/editorPlatos.html ↻

- Entrantes
- Panecillos
- Ensalada
- Sopa
- Carnes
- Filete de ternera
- Filetón de cerdo
- Pescados
- Lubina a la plancha
- Pez espada
- Dorada
- Postres
- Flan
- Natillas caseras
- Tarta



No se ha seleccionado ningún archivo.

ID:

Plato:

Disponible:

Descripción:

Categoría:

Precio:

Tiempo preparación:


Acciones:

Para el editor de menús:

localhost/www/editorMenus.html

-Menus

- Sopa y pescado
- Ensalada y carne



Examinar... No se ha seleccionado ningún archivo.

ID: Disponible

Descripción:

Plato 1:

Plato 2:

Plato 3:

Precio:

Acciones:

Administración:

localhost/www/configuracion.php

UJa.

Configuración

Examinar... No se ha seleccionado ningún archivo.

Nombre:

Mostrar Tiempos

Pin Cocina:

Pin Camarero:

Pin Admin:

Indicaciones

Nombre: nombre en todas las páginas

Idioma por defecto: Lenguaje de la interface

Mostrar tiempos: Habilita/deshabilita la aproximación de tiempo en la carta

Cocina: pin para acceder a la cocina

Camarero: pin para acceder a la aplicación de camareros

Admin: Pin para acceder a este menú

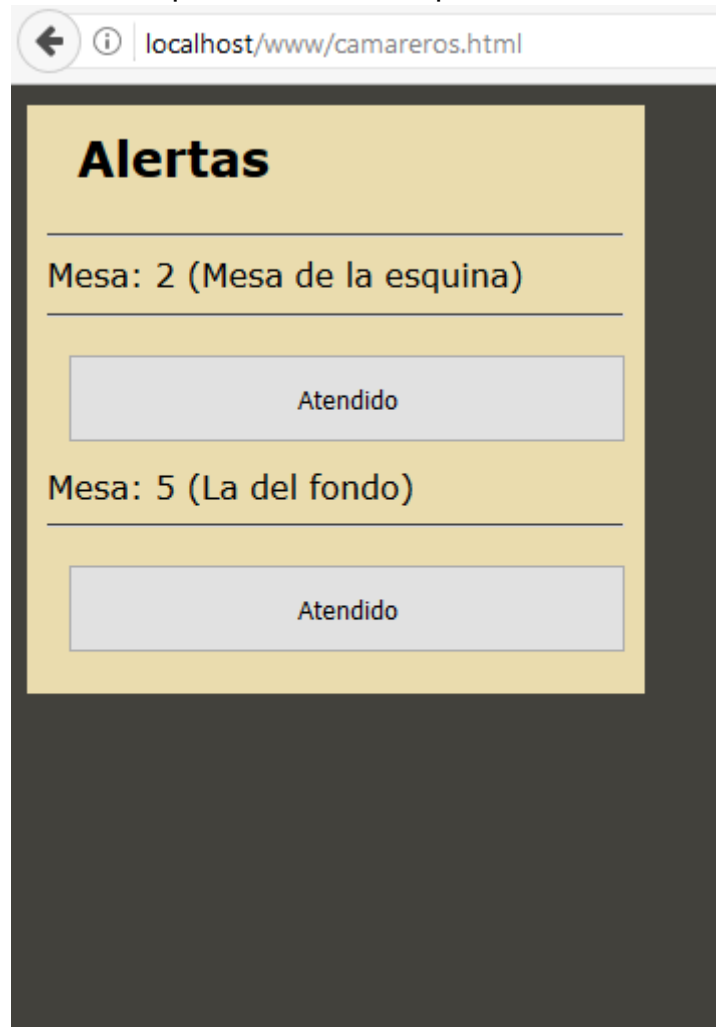
jps00008@red.ujaen.es

Para la aplicación de la cocina, es necesario poder identificar los elementos como elementos interactivos. Por esto se opta por

darle una forma redondeada con un tono de color ligeramente diferente al fondo. Que, en cierto modo, invite a ser arrastrado:



La aplicación de alertas para camareros esta pensada para que se pueda apreciar en formato vertical. De forma de que los camareros puedan tenerla disponible en su teléfono móvil.



Y, por último, las diferentes páginas de la aplicación del cliente:

UJa.

Mesa: Mesa Redonda (1)

Cuenta: 15€



Sopa y pescado

Precio 15.00€

- Panecillos
- Lubina a la plancha
- Flan

Pedir



Ensalada y carne

Precio 11.00€

- Ensalada
- Filetón de cerdo
- Natillas caseras

Pedir

UJa.

Mesa: Mesa Redonda (1)

Cuenta: 15€



Panecillos

terminado

Panecillos al horno con ajo y perejil

Cancelar

Lubina a la plancha

proceso

Filete de lubina con limon y finas hierbas

Cancelar

Flan

espera

Flan casero de huevo

Cancelar

Llamar a un Camarero



Entrantes

Carnes

Pescados

Postres

Menus



Panecillos

Precio 2.50€

Panecillos al horno con ajo y perejil

Pedir



Ensalada

Precio 1.50€

Ensalada de lechuga y maiz

Pedir

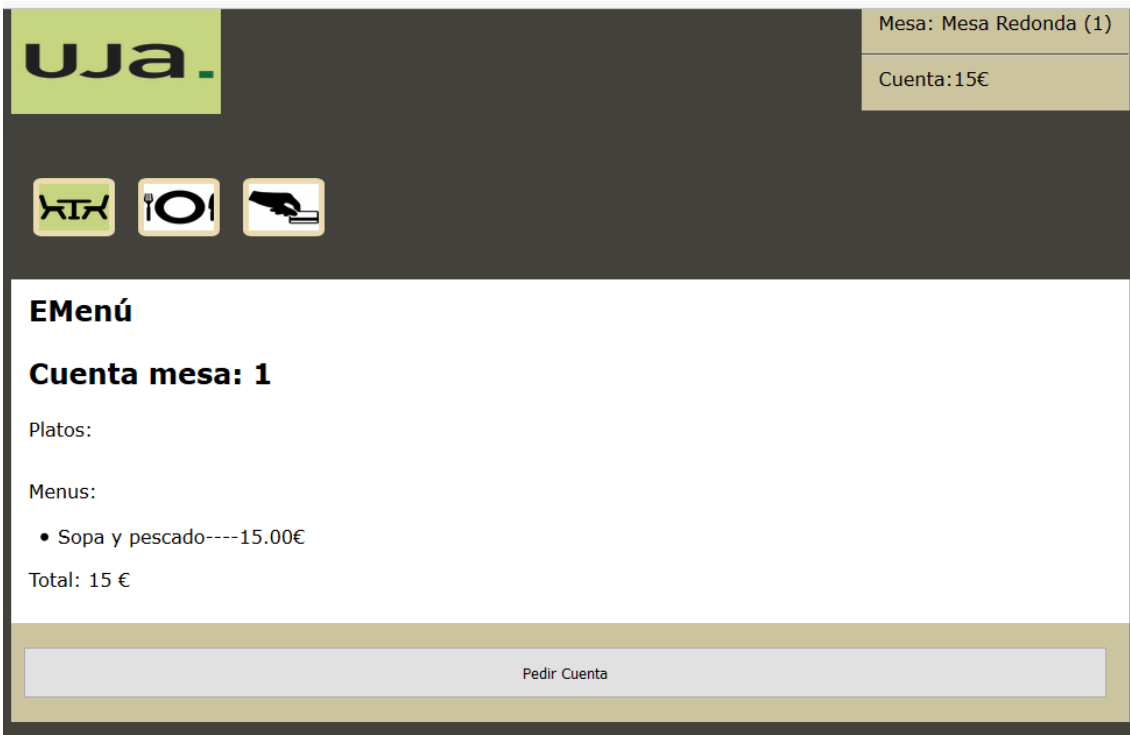


Sopa

Precio 2.00€

Sopa de pollo

Pedir



5.12.3. Demostración de los requisitos completados

Como bien se muestra en el punto 5.12.2., se ha terminado desarrollar una hoja de estilos para todas las secciones de las páginas. Este estilo es uniforme para todas para dar sensación de cohesión y unicidad.

5.12.4. Retrospectiva

Quienes han probado la aplicación han dado un “feedback” positivo. La formas nuevas de las secciones ayudan a la facilidad de uso de estas; siendo ahora más fáciles de comprender.

5.12.5. Refinamiento de la lista de requisitos y cambios en el proyecto

La buena aceptación de la aplicación y sus componentes hacen suficientes los esfuerzos empleados en el desarrollo y podemos concluir que no son necesarias más iteraciones. Por lo que podemos dar por finalizado el proceso de desarrollo.

6. Resumen

Aunque el resultado final es notoriamente similar al de la idea inicial, durante el desarrollo se han realizado cambios significativos debidos a problemas no previstos; como lo fue el uso anticipado de la variables "Session" en el servidor, o por petición de quienes han testado la aplicación entre iteraciones.

6.1. Diagrama Componetes Administración

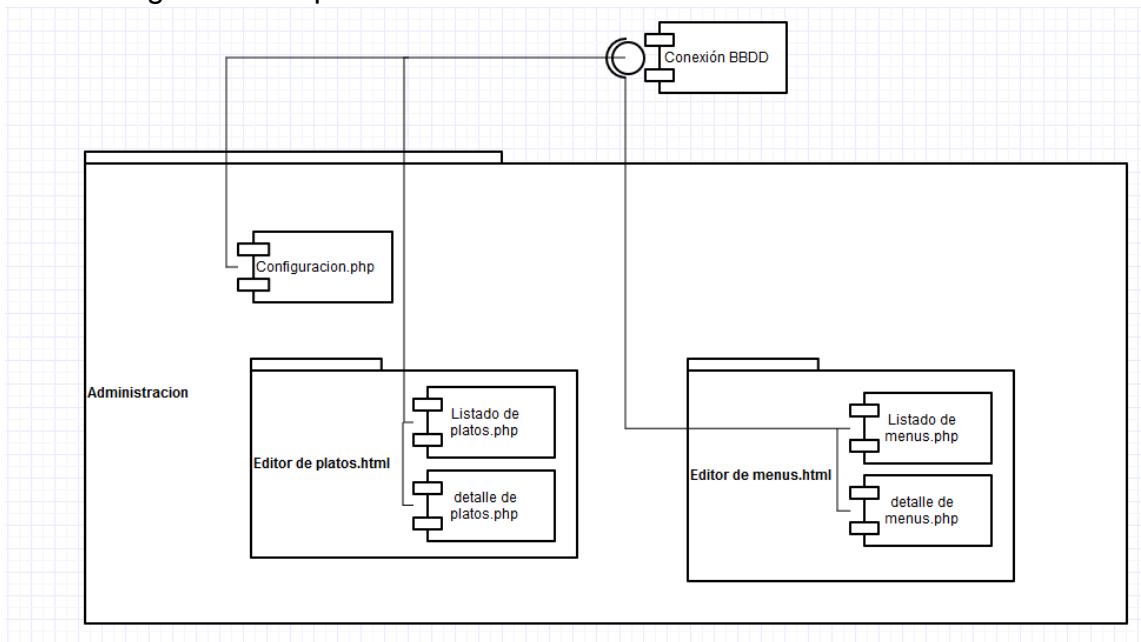


Ilustración 36: Diagrama de Componentes de Administración

6.2. Diagrama administración cocina

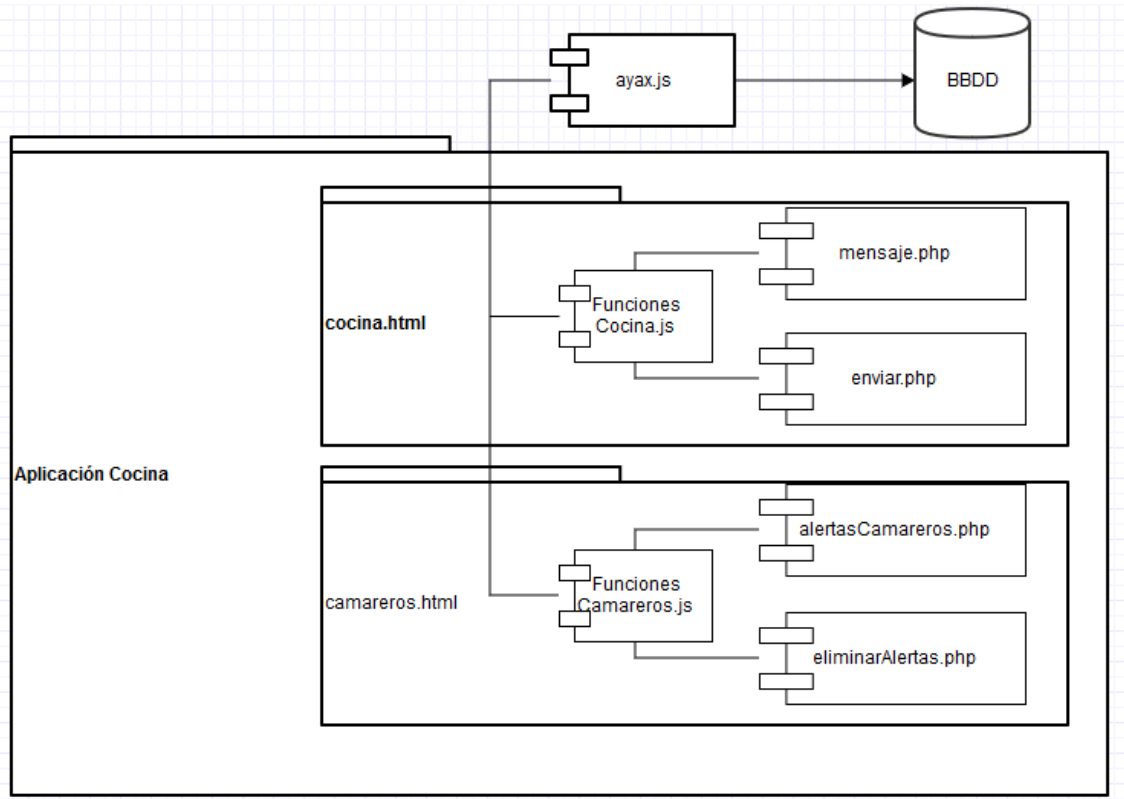


Ilustración 37: Diagrama de componentes de la aplicación de cocina

6.3. Diagrama aplicación cliente

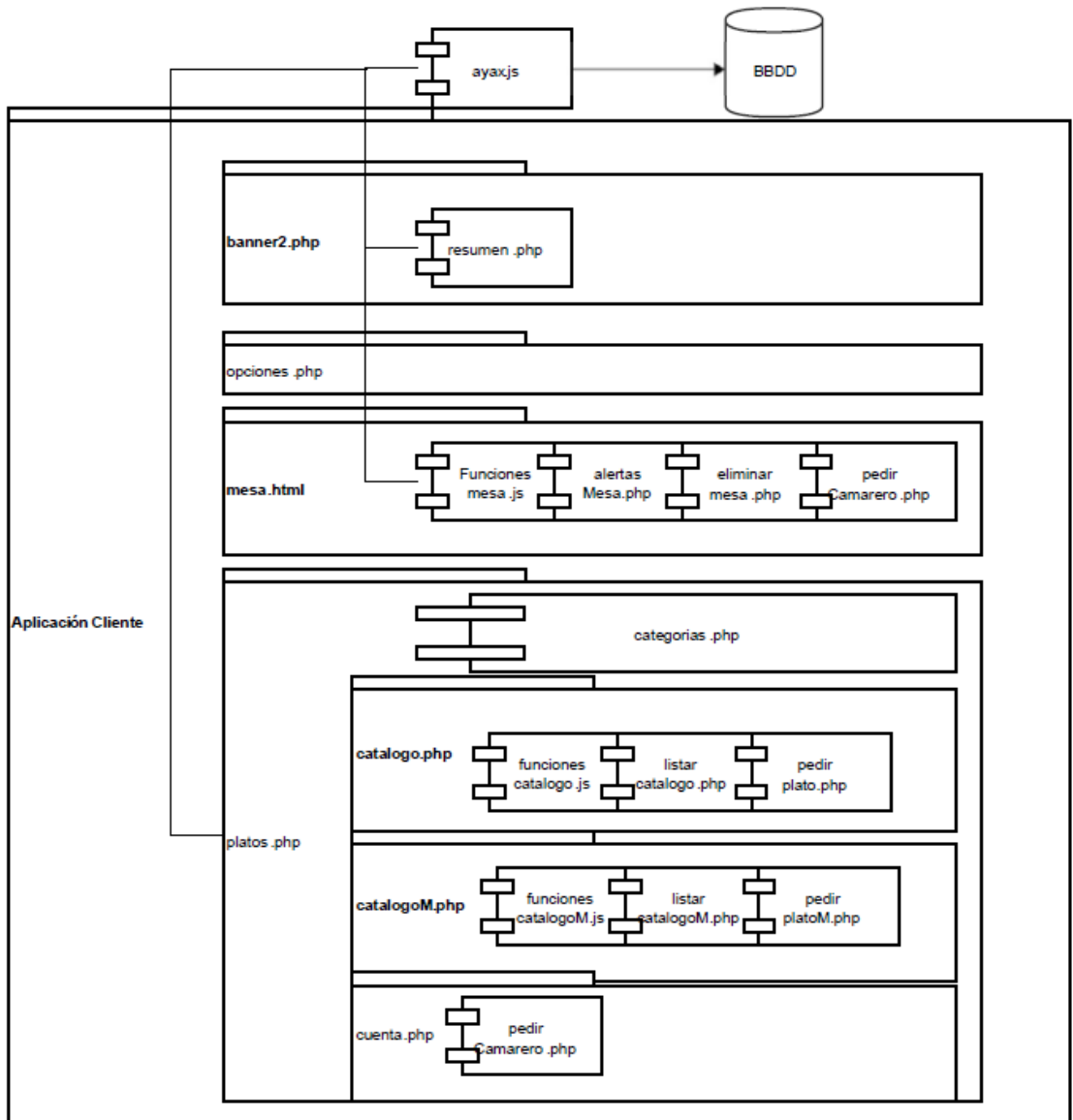


Ilustración 38: Diagrama de componentes de aplicación de cliente

Anexo 1: Manual Técnico de instalación de servidor

Capítulo 1 Preparación del servidor.

A continuación se expone un detallado manual para que cualquiera que siga las instrucciones con un mínimo conocimiento de informática pueda dejar listo el servidor necesario para hacer funcionar la aplicación

Instalación del SO:

Para la instalación del SO se ha optado por usar una ISO original de CentOS (http://isoredirect.centos.org/centos/7/isos/x86_64/CentOS-7-x86_64-Minimal-1511.iso). El método de instalación a ser por el clásico arranque por DVD.

Configurar una SSH:

Como el asunto del consumo de energía debe preocuparnos a todos, la instalación del SO es minimalista y sin interface gráfica. Al no tener un monitor, el consumo eléctrico se reduce aún más. Pero necesitamos poder usar el servidor para configurarlo. Por este motivo voy a instalar una SSH que permita el acceso al servidor desde otro ordenador con un uso menos específico.

Una vez en el servidor instalamos un servidor SSH, en nuestro caso: "open ssh server" con el comando:

```
"yum install openssh-server"
```

Ahora nos vamos a dedicar a cambiar algunas opciones del servidor (/etc/ssh/sshd_config) como por ejemplo:

- Puerto del 22 al 23033 este paso es solo para dar un mínimo más de seguridad
- Protocolo 2
- Permitir acceso de Root: no (de momento voy a dejarlo en si hasta que cree nuevos usuarios)

Con el comando siguiente reseteamos el servicio ssh.

```
"systemctl restart sshd"
```

Ya está listo nuestro servicio SSH. Ahora para su correcto funcionamiento, voy a configurar la ip del servido de forma estática. Esto lo voy a conseguir editando los archivos ifcfg-eth0 y ifcfg-lo (/etc/sysconfig/network-scripts)

En los cuales se han añadido las líneas

```
BOOTPROTO=static  
IPADDR=192.168.1.50  
NETMASK=255.255.255.0  
NETWORK=192.168.1.0  
GATEWAY=192.168.1.1
```

Después reseteamos la conexión con "systemctl restart network"

Después de todo esto ya puedo trabajar desde casa.

Capítulo 2 Instalación de un servicio web:

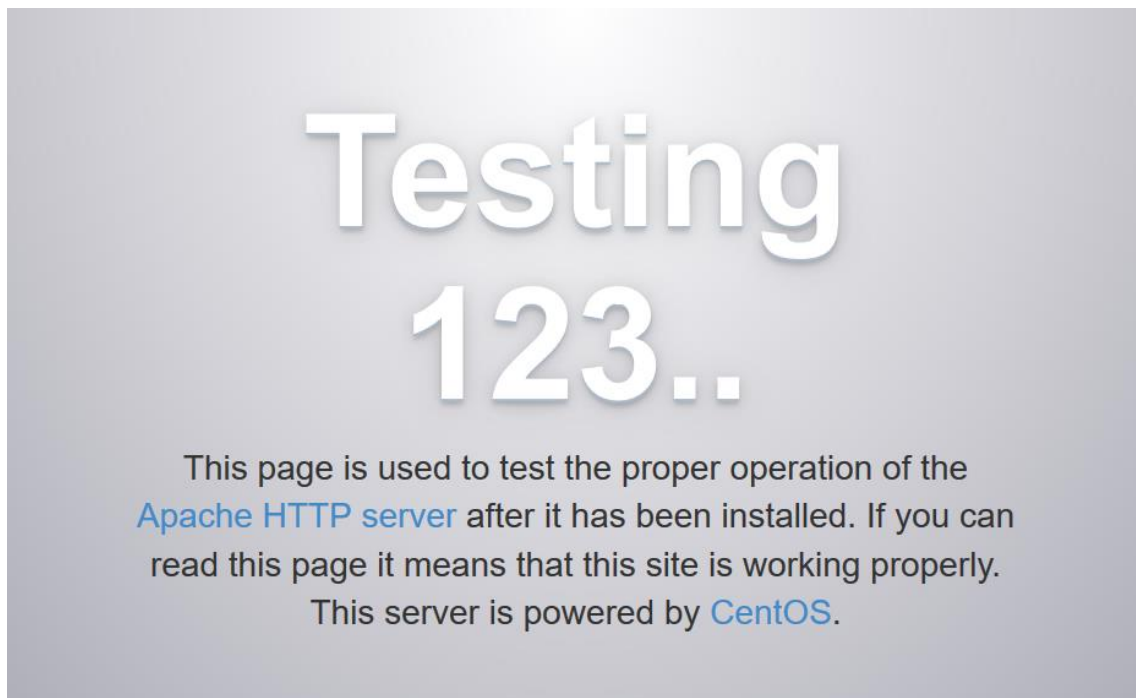
Me decanto por instalar un servidor Apache por su popularidad y por su gran potencial con PHP.

```
“yum install httpd”
```

Es el momento de realizar una prueba y comprobar que ha funcionado la instalación. Empezamos lanzando el servicio y habilitándolo:

```
“systemctl start httpd.service” y “systemctl enable httpd.service”
```

Usando el navegador web entro en la ip de mí maquina por el puerto 80:



Estas son buenas noticias.

Extra: Podemos dar servicio fuera de la red local. En este caso no es útil para el proyecto. Pero puede ser un campo aun por explorar en caso de que solo queramos tener un único servidor para varios establecimientos.

```
“firewall-cmd –permanet –zone=public –add-service=http”
```

```
“firewall-cmd --permanent --zone=public --add-service=https”
```

```
“firewall-cmd –reload”
```

Nota: es posible que firewall no esté instalado. No es lo común pero a mí me ha pasado. Para solucionar esto la instalamos:

```
“yum install firewall”
```

Ahora que tenemos un servicio web tenemos vamos a instalar php en nuestro servidor.

```
“yum -y install php”
```

```
“systemctl restart httpd.service”
```

Y comprobamos que funciona correctamente creando un fichero en /var/www/html/

```
“vi /var/www/html/info.php”
```

Con el código:


```
“<?php  
phpinfo();  
?>”
```

Accedemos al fichero que hemos creado, en mi caso: <http://192.168.1.50/info.php>



System	Linux localhost.localdomain 3.10.0-327.el7.x86_64 #1 SMP Thu Nov 19 22:10:57 UTC 2015 x86_64
Build Date	May 12 2016 13:46:18
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/fileinfo.ini, /etc/php.d/json.ini, /etc/php.d/phar.ini, /etc/php.d/zip.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,NTS
PHP Extension Build	API20100525,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.4.0, Copyright (c) 1998-2013 Zend Technologies

Powered By 

Como se puede apreciar, el proceso ha ido bien.

Instalar una base de datos MySQL Server

Para instalar MySQL Server en Linux CentOS 7 Minimal necesitaremos descargar el rpm del repositorio de MySQL e instalarlo previamente. No podremos instalar MySQL Server con Yum directamente pues las distribuciones CentOS vienen preparadas para instalar MariaDB con un sencillo yum install mariadb-server.

```
“yum install wget”
```

Descargaremos el paquete RPM del repositorio de MySQL Server con el comando:

```
“wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm”
```

RPM del repositorio de MySQL Server con el comando:

```
“rpm -ivh mysql-community-release-el7-5.noarch.rpm”
```

Podremos verificar que los repositorios de MySQL han quedado instalados ejecutando:

```
“ls -l /etc/yum.repos.d/mysql-community*”
```

Nos devolverá:

```
“ /etc/yum.repos.d/mysql-community.repo  
 /etc/yum.repos.d/mysql-community-source.repo”
```

Ahora sí que podemos descargar MySQL server con un yum

```
“yum install mysql-server”
```

Una vez instalado es el turno de lanzar el servicio con el comando:

```
“systemctl start mysqld”
```

Ya que la base de datos es una herramienta exclusiva de la aplicación web, no daremos acceso a la BBDD desde el exterior. La gestión de esta será vía web con la herramienta “phpmyadmin”. Al igual que MySQL Server, los repositorios por defecto de Linux CentOS 7 no incluyen la instalación de phpMyAdmin. Para instalarlo, en primer lugar ejecutaremos el siguiente comando para habilitar el repositorio:

```
“rpm -iUvh http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm”
```

E instalamos phpMyAdmin con:

```
“yum -y install phpmyadmin”
```

Configurar la seguridad de nuestro servidor MySQL:

```
“mysql_secure_installation”
```

Con el anterior comando lanzaremos un sencillo asistente de texto para establecer los valores de configuración.

Extra:

Podemos hacer que solo un equipo tenga acceso a la aplicación phpMyAdmin. Debido a

la naturaleza de la aplicación esto es altamente recomendable. El fichero e configuración es el siguiente:
“/etc/httpd/conf.d/phpMyAdmin.conf”

```
<Directory /usr/share/phpMyAdmin/>
  AddDefaultCharset UTF-8

  <IfModule mod_authz_core.c>
    # Apache 2.4
    <RequireAny>
      Require ip 127.0.0.1
      Require ip 192.168.1.33
      Require ip ::1
    </RequireAny>
  </IfModule>
  <IfModule !mod_authz_core.c>
    # Apache 2.2
    Order Deny,Allow
    Deny from All
    Allow from 127.0.0.1
    Allow form 192.168.1.33
    Allow from ::1
  </IfModule>
</Directory>

<Directory /usr/share/phpMyAdmin/setup/>
  <IfModule mod_authz_core.c>
    # Apache 2.4
    <RequireAny>
      Require ip 127.0.0.1
      Require ip 192.168.1.33
      Require ip ::1
    </RequireAny>
  </IfModule>
  <IfModule !mod_authz_core.c>
    # Apache 2.2
    Order Deny,Allow
    Deny from All
    Allow from 127.0.0.1
    Allow from 192.168.33
    Allow from ::1
  </IfModule>
</Directory>
```

Teniendo en cuenta que el ordenador desde el que estoy accediendo es el 192.168.1.33

Podremos acceder a phpMyAdmin con la dirección web

<http://192.168.1.50/phpMyAdmin>

Recuerdo una vez más que el servidor está montado en 192.168.1.50

Anexo 2: Guía de usuario para administrador de la cocina
Interface principal

UJa.

Selección de servicio




Añade o modifica platos en el menú de platos
Añade un modifica menús en Menús
Cambia una configuración en Configuración

jps00008@red.ujaen.es

En la aplicación de administración puede encontrar 3 elementos.

: Da acceso al menú de edición de platos.

: Da acceso a la edición de menús

: Da acceso a la administración interna de la aplicación

Editar platos

1

2

3 Examinar... No se ha seleccionado ningún archivo.

4 ID: 1

5 Plato: Panecillos

6 Disponible

7 Descripción: Panecillos al horno con ajo y perejil

8 Categoría: Entrantes

9 Precio: 2.50

10 Tiempo preparación: 5

Acciones:

11 12

1: Lista de platos ordenados según su categoría. Al seleccionar uno se mostrará su información en la parte derecha de la pantalla.

2: Foto del plato.

3: Botón para buscar una foto nueva para el plato.

4: Identificador del plato. Este parámetro no es editable, su función es meramente informativa y se autocompleta.

5: Nombre del plato.

6: Disponibilidad. Habilita o deshabilita el plato en la carta.

7: descripción de plato.

8: Categoría del plato.

9: Precio del plato.

10: tiempo de preparación estimado en minutos.

11: Al pulsar el botón de añadir, el plato que se muestra en pantalla se **añadirá como un plato nuevo.**

12: El botón de modificar sustituye o **actualiza** el plato seleccionado.

Editor de menús:

1

2

3 Examinar... No se ha seleccionado ningún archivo.

4 ID: 1 Disponible

6 Descripción: Sopa y pescado

7 Plato 1: Panecillos

8 Plato 2: Lubina a la plancha

9 Plato 3: Flan

10 Precio: 15.00

Acciones: Añadir Modificar

11 12

- 1: Listado de menús disponibles. Al seleccionar uno con el ratón, a la derecha se mostrará toda la información sobre este.
- 2: Foto del menú.
- 3: Botón para buscar una foto nueva para el menú.
- 4: Identificador del Menú. Este parámetro no es editable, su función es meramente informativa y se autocompleta.
- 5: Disponibilidad. Habilita o deshabilita el plato en la carta.
- 6: descripción de menú.
- 7: Lista desplegable que contiene todos los primeros platos.
- 8: Lista desplegable que contiene todos los segundos platos.
- 9: Lista desplegable que contiene todos los postres.
- 10: Precio del menú en euros
- 11: Al pulsar el botón de añadir, el plato que se muestra en pantalla se **añadirá como un menú nuevo.**
- 12: El botón de modificar sustituye o **actualiza** el menú seleccionado.

Editor de opciones

UJa.

Configuración

UJa. ¹

No se ha seleccionado ningún archivo. ²

Nombre: ³

Mostrar Tiempos ⁴

Pin Cocina: ⁵

Pin Camarero: ⁶

Pin Admin: ⁷

⁸

Indicaciones

Nombre: nombre en todas las páginas

Idioma por defecto: Lenguaje de la interface

Mostrar tiempos: Habilita/deshabilita la aproximación de tiempo en la carta

Cocina: pin para acceder a la cocina

Camarero: pin para acceder a la aplicación de camareros

Admin: Pin para acceder a este menú

jps00008@red.ujaen.es

1: Logo actual del restaurante.

2: Botón de examinar. Busca en tu equipo un nuevo logo.

3: Nombre del restaurante. Este nombre aparecerá en diferentes secciones de la carta.

4: Marca esta casilla para que en la carta aparezcan los tiempos estimados de preparación de los platos.

5: Pin de cocina. Clave que se usa para tener acceso a la cocina.

6: Pin de Camarero. Clave que se usa para tener acceso a la aplicación del camarero.

7: Pin administración. Clave que permite acceder al menú de administración.

8: Botón para guardar los cambios.

Cocina

En la página de la cocina podemos diferenciar 3 secciones.



La primera es la lista de comandas que están en espera. Estas son las comandas que están pendientes de cocinar.

La segunda es la lista de platos en proceso. Son platos que están siendo elaborados en estos momentos.

La tercera y última es la lista de terminados. Estas comandas ya están terminadas y listas para llevar al usuario final.

¿Como pasar de una comanda a otra?

Método A; Usando los botones ← y →:

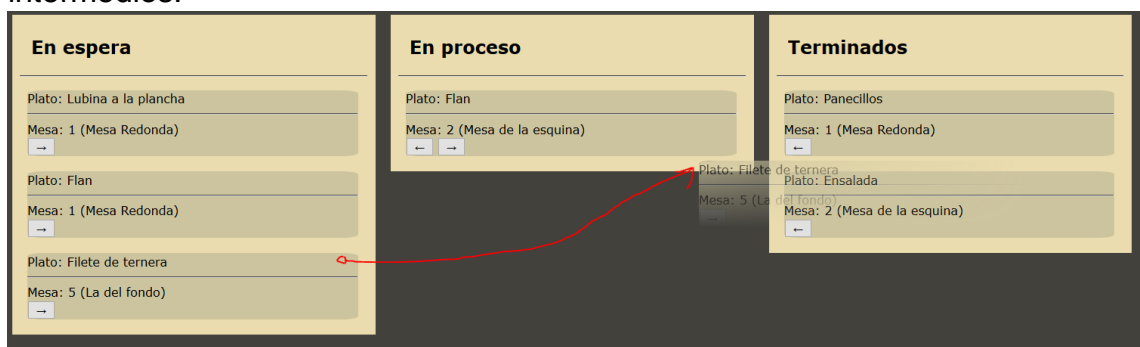
Todas las comandas tienen al menos un botón. Las comandas en espera un “→” que cambiará el estado de la comanda a en proceso.

Las comandas en proceso disponen de dos botones. “←” Devuelve la comanda al estado de en espera. El otro botón “→” cambia el estado a Terminados.

Por último, en terminados, solo se puede apreciar un botón “←” que cambia el estado de la comanda a en proceso.

Método B; Arrastrando:

Las comandas se puede diferenciar por tener un color ligeramente más oscuro. Pulsando en esta área de color, podemos arrastrar el elemento a la lista de comandas que queramos sin necesidad de pasar por estados intermedios.



En espera	En proceso	Terminados
Plato: Lubina a la plancha	Plato: Flan	Plato: Panecillos
Mesa: 1 (Mesa Redonda)	Mesa: 2 (Mesa de la esquina)	Mesa: 1 (Mesa Redonda)
Plato: Flan	Plato: Filete de ternera	Plato: Ensalada
Mesa: 1 (Mesa Redonda)	Mesa: 5 (La del fondo)	Mesa: 2 (Mesa de la esquina)

Camareros

La pagina de los camareros recibirá una notificación en el momento en el que una mesa pida ayuda de un camarero.

Alertas

Mesa: 2 (Mesa de la esquina)

Mesa: 5 (La del fondo)

Cuando un camarero se disponga a atender la petición, solo tendrá que pulsar sobre el botón de “atendido” para descartar la notificación.

Anexo 3: Manual de cliente:

Página principal del cliente y elementos

La página inicial muestra una lista de menús además de las opciones básicas. Estas opciones están siempre disponibles en pantalla.



1: Mesa y numero de mesa.

2: Importe total de las comandas

3: Mesa. En este menú se puede ver las comandas realizadas y su estado.

4: Carta. Listado de platos y menús

5: Cuenta. Lista detallada de la factura. Desde aquí también se puede pedir la cuenta

6: Elemento de catalogo. Mediante el botón "Pedir" se añade la lista de platos, en caso de menú, o un plato individual, a la lista de comandas.

Página de mesa

The screenshot shows a digital interface for a restaurant table. At the top left is the logo 'uja.' in white on a green background. To the right, it displays 'Mesa: Mesa Redonda (1)' and 'Cuenta: 15€'. Below the logo are three icons: a stylized 'KTX', a fork and knife, and a hand holding a card. The main area contains a list of orders, each with a number, a description, a 'Cancelar' button, and a status indicator. The orders are: 1. Panecillos (terminado), 2. Panecillos al horno con ajo y perejil (cancelado), Lubina a la plancha (proceso), Filete de lubina con limon y finas hierbas (cancelado), Flan (espera), and Flan casero de huevo (cancelado). At the bottom, there is a red number '8' and a button labeled 'Llamar a un Camarero'.

Order ID	Description	Action	Status
1	Panecillos		terminado
2	Panecillos al horno con ajo y perejil	Cancelar	
	Lubina a la plancha		proceso
	Filete de lubina con limon y finas hierbas	Cancelar	
	Flan		espera
	Flan casero de huevo	Cancelar	

8
Llamar a un Camarero

1: Nombre de plato

2: Descripción del plato

3: Botón de cancelado de pedido (deshabilitado)

4: Botón de cancelado de pedido. Este botón esta habilitado mientras que la comanda este en estado de "espera"

5: Estado de comanda: Terminado. El plato ya ha terminado su proceso de preparación.

6: Estado de comanda: Proceso. El plato se encuentra actualmente en proceso de preparación.

7: Estado de comanda: Espera. El plato aún no se está procesando.

8: Botón "Llamar a un Camarero": Este botón manda una petición de atención a los camareros para ofrecer ayuda.

Página de Platos

The screenshot shows a web interface for a restaurant menu. At the top left is the logo 'uja.'. At the top right, there are two boxes: 'Mesa: Mesa Redonda (1)' and 'Cuenta: 15€'. Below the logo are three icons: a stylized 'XIX', a fork and knife, and a hand holding a card. A horizontal navigation bar contains five buttons: 'Entrantes', 'Carnes', 'Pescados', 'Postres', and 'Menus'. A red box highlights these buttons, with a red '1' next to it. Below the navigation bar, there are three menu items, each with a small image on the left, a title and description in the middle, and a price on the right. A 'Pedir' button is located below each item. Red numbers 2 through 6 are placed on the page to indicate specific elements: 2 points to the image, 3 to the title, 4 to the description, 5 to the price, and 6 to the 'Pedir' button.

2	3	4	5	6
	Panecillos	Panecillos al horno con ajo y perejil	Precio 2.50€	Pedir
	Ensalada	Ensalada de lechuga y maiz	Precio 1.50€	Pedir
	Sopa	Sopa de pollo	Precio 2.00€	Pedir

1: Botones de categoría: Al pulsar sobre cada una de las categorías, el listado de platos cambiará mostrando los correspondientes a cada categoría.

2: Ilustración del plato.

3: Nombre del plato.

4: Descripción del plato.

5: Precio.

6: Botón de "Pedir": añade a la comanda el plato seleccionado.

Página de Cuenta.

En esta página se muestran el ticket de la cuenta que acumulan todas las comandas. Está dividido en dos partes. Una donde muestra los platos individuales. La otra muestra el menú sin detallar con el descuento aplicado.

Mesa: Mesa Redonda (1)
Cuenta: 18€

EMenú

Cuenta mesa: 1

Platos: **1**

- Tarta----3.00€

Menus: **2**

- Sopa y pescado----15.00€

Total: 18 € **3**

Pedir Cuenta **4**

1: Listado de platos pedidos de forma individual y su importe.

2: Listado de Menús pedido y su importe.

3: Importe total.

4: Botón "Pedir Cuenta". Envía una alerta a la cocina para iniciar el proceso de cobro.

Referencias:

Información SCRUM:

<https://proyectosagiles.org/que-es-scrum/>

AJAX:

http://aprenderaprogramar.es/index.php?option=com_content&view=article&id=882:igu-e-es-y-para-que-sirve-ajax-ventajas-e-inconvenientes-javascript-asincrono-xml-y-json-cu01193e&catid=78:tutorial-basico-programador-web-javascript-desde-&Itemid=206

Montaje de servidores

[servidohttp://www.ajpdsoft.com/modules.php?name=News&file=article&sid=702r](http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=702r)

Asociación Interacción Persona-Ordenador:

<http://aipo.es/>

Editor diagramas online:

<https://www.gliffy.com>

<https://www.tomsplanner.es>

Calculadora COCOMO

<http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/baker/cocomo.html>

Calculadora de salarios

http://cincodias.com/herramientas/calculadora_sueldo_neto/#tabla_resultados

Estadísticas iOS y Android

<http://7am-lab.com/OS-in-Europe/>

Introducción a diagramas:

http://users.dsic.upv.es/asignaturas/facultad/lisi/ejemplorup/Casos_Uso.html

Referencias de aplicaciones similares:

<https://www.youtube.com/watch?v=4XtRpJvCA3M>

<https://www.youtube.com/watch?v=cQij855W0T0#t=56>