

**Universitat Oberta
de Catalunya**

Aplicación móvil en HTML5 para la gestión de listas de la compra de alimentación

**Consultor: Victor Carceler Hontoria
Alumno: Juan Francisco del Río Pacheco**

Madrid, enero de 2013

Resumen

Memoria final PEC4 para el TFC – Desarrollo de aplicaciones móviles
Aplicación móvil en HTML5 para la gestión de listas de la compra de alimentos

Contacto:

Juan Francisco del Río Pacheco
jdelrio_@uoc.edu

Contenido

1. Introducción	4
1.1 Descripción del proyecto.....	4
1.2 Objetivos del proyecto.....	4
1.3 Hardware / software.....	5
1.4 Tecnologías elegidas.....	5
2. Planificación	7
2.1 Tareas principales.....	7
2.2 Entregables.....	7
3. Calendario	9
3.1 Fechas de entrega.....	9
3.2 Planificación temporal.....	9
4. Funcionalidades	
4.1 Home.....	11
4.2 Ingredientes.....	12
4.3 Consulta del catálogo de recetas.....	14
4.4 Consulta detalle de receta.....	15
4.5 Alta de receta / edición de receta.....	16
4.6 Alta de ingrediente a la receta.....	17
4.7 Baja de ingrediente de la receta.....	18
4.8 Baja de receta.....	19
4.9 Consulta del catálogo de listas de la compra.....	20
4.10 Consulta de lista de la compra y proceso de compra.....	21
4.11 Alta de lista de la compra / edición de lista de la compra.....	22
4.12 Alta de receta a la lista de la compra.....	23
4.13 Baja de receta de la lista de la compra.....	24
4.14 Alta de ingrediente a la lista de la compra.....	25
4.15 Baja de ingrediente de la lista de la compra.....	26
4.16 Baja de la lista de la compra.....	27
5. Mapa de la aplicación.....	27
6. Casos de uso.....	29
7. Diseño de la persistencia.....	34
8. Implementación.....	36
8.1 Modificaciones en la arquitectura de la información.....	36
8.2 Modificaciones en la tecnología usada en la aplicación.....	41
8.3 Angularjs.....	41
9. Glosario.....	45
10. Fuentes consultadas.....	46
11. Índice de figuras.....	47

1. Introducción

1.1 Descripción del proyecto

El proyecto elegido para desarrollar es una aplicación móvil en HTML5 para la gestión de listas de la compra de alimentos.

¿Por qué HTML5? Las aplicaciones móviles desarrolladas con tecnología web tienen la gran ventaja de ser muy reutilizables. Un desarrollo realizado para Android se puede usar en casi su totalidad para hacer la versión para iPhone o para hacer una página web o una app para Chrome. Y dentro de muy poco una aplicación para Windows 8.

¿Por qué listas de la compra? El motivo de elegir realizar una aplicación para gestionar una lista de la compra de alimentación surge principalmente de una necesidad. Existen multitud de aplicaciones para hacer una lista de la compra, tanto para Android como para iPhone. Como usuario de algunas de ellas en mi teléfono Android siempre echo en falta una funcionalidad para mí básica: no quiero introducir uno a uno los elementos de la lista de la compra, quiero elegir cuales son las recetas que voy a elaborar esta semana y que la aplicación me genere la lista de la compra.

La idea es pues en un principio fácil: una aplicación CRUD (create, read, update y delete) para gestionar ingredientes de cocina y recetas para poder generar una lista de la compra de una manera sencilla.

Además, hay muchas posibilidades de expansión las que ofrece una aplicación de este tipo. Se podría hacer que los datos se guardaran en una base de datos en un servidor para realizar aplicaciones de gestión de las listas mediante web o una app para una tablet. Esto facilitaría la gestión y el móvil se usaría principalmente para la consulta. También tiene posibilidades para añadir funcionalidades de red social o para integrar recomendaciones de marcas para monetizar el proyecto.

1.2 Objetivos del proyecto

El objetivo de este Trabajo de Fin de Carrera es la realización de una aplicación móvil para la gestión (consulta, actualización, creación y borrado) de listas de la compra de alimentación y de las recetas e ingredientes asociadas a éstas.

Con el desarrollo de esta aplicación se consiguen los siguientes objetivos:

- Conocer las herramientas necesarias para la generación de una aplicación móvil a partir de tecnologías web.
- Estudiar la metodología de desarrollo de software MVC.
- Estado del arte de las librerías javascript necesarias para el desarrollo de una aplicación móvil y para trabajar mediante el patrón MVC.
- Desarrollar competencias de definir, planificar y desarrollar un proyecto y generar toda la documentación necesaria.

1.3 Hardware / software

Para este proyecto se usará la siguiente infraestructura:

Ordenador personal: HP Pavillion P6590es, Intel Core i7 860, 8Gb de memoria RAM

Móvil Android: LG Optimus Black 970. Versión Android 2.2.2

Sistema operativo: Windows 7 Home Premium

Editor código: Sublime Text

Editor gráficos: Photoshop CS5

Editor texto: LibreOffice 3.6.1.2

Diagramas Gantt: GanttProject 2.5.5

Compilación de la app: PhoneGap 2.1.0 y Eclipse 4.2.0

1.4 Tecnologías elegidas

El proyecto consiste en la realización de una app móvil con tecnologías HTML5. HTML5 se ha convertido en una palabra comodín para englobar un conjunto de lenguajes usados en el desarrollo web. Estos lenguajes además del HTML5 propiamente dicho serían CSS y Javascript.

Hay multitud de librerías diferentes que ayudan a afrontar un desarrollo móvil basada en tecnologías web. En mi caso la elección ha sido la siguiente:

jQuery: librería genérica de javascript que resulta de gran utilidad para trabajar con el DOM del navegador, hacer peticiones Ajax o realizar animaciones y efectos. Es con mucha diferencia la librería de este tipo mas usada y tiene una enorme cantidad de plugins que lo complementan.

Backbone: es una librería javascript que proporciona una estructura básica para trabajar con el modelo de desarrollo MVC.

Underscore: librería de javascript con más de 80 utilidades para extender el javascript nativo con más funcionalidades.

Phonegap: herramienta para convertir una aplicación web en una aplicación móvil. Soporta multitud de SS.OO móviles pero en este proyecto solo se usará la conversión a aplicación para Android.

2. Planificación

2.1 Tareas principales

Definición del proyecto

Duración 9 días. Se realiza una investigación inicial y documenta una definición del proyecto, su alcance, objetivos y duración en el tiempo.

Fase UX (User eXperience)

Duración 28 días. Se realiza un prototipo, un diseño y finalmente una maqueta final HTML/CSS/Javascript.

Fase Backoffice

Duración 42 días. Se implementa toda la parte lógica de la aplicación mediante la creación de la base de datos, la definición de los modelos, vistas y controladores y finalmente añadimos Backbone.js al desarrollo.

Fase Final

Duración 28 días. Mediante PhoneGap generamos la aplicación lista para instalar en un móvil Android para testarla y hacer las modificaciones pertinentes si se encontrara alguna incidencia. Por último se redactan la presentación y la memoria final del proyecto.

2.2 Entregables

PEC 1: plan de trabajo

En esta fase se realiza un documento con el plan de trabajo a seguir para la consecución del proyecto. Se realiza una investigación inicial donde conocer la dificultad del proyecto, las tecnologías y equipo a utilizar y se esboza una primera definición de cual es el objetivo del proyecto. Con esta información se elabora una planificación temporal mediante un diagrama de Gantt de los tiempos y tareas del proyecto.

PEC 2: memoria parcial

Empezamos a modelar el proyecto mediante un prototipo que nos permita

interaccionar con él para, mediante iteraciones, ir modificando las características hasta lograr un modelo final. Seguidamente nos centramos en el aspecto visual de la aplicación trabajando el diseño gráfico. Y por último empezaremos el desarrollo del código HTML/CSS/Javascript que nos permita tener una maqueta que visualmente sea la más cercana al producto terminado. Por último elaboramos el documento que recoja el resultado de todo lo trabajado anteriormente.

PEC 3: memoria parcial

En esta fase empezamos a trabajar con el desarrollo de la base de datos, la conceptualización de los modelos, vistas y controladores de la aplicación y su implementación con Backbone. Finalmente, se entrega la documentación de todo el trabajo realizado en esta fase.

Memoria final

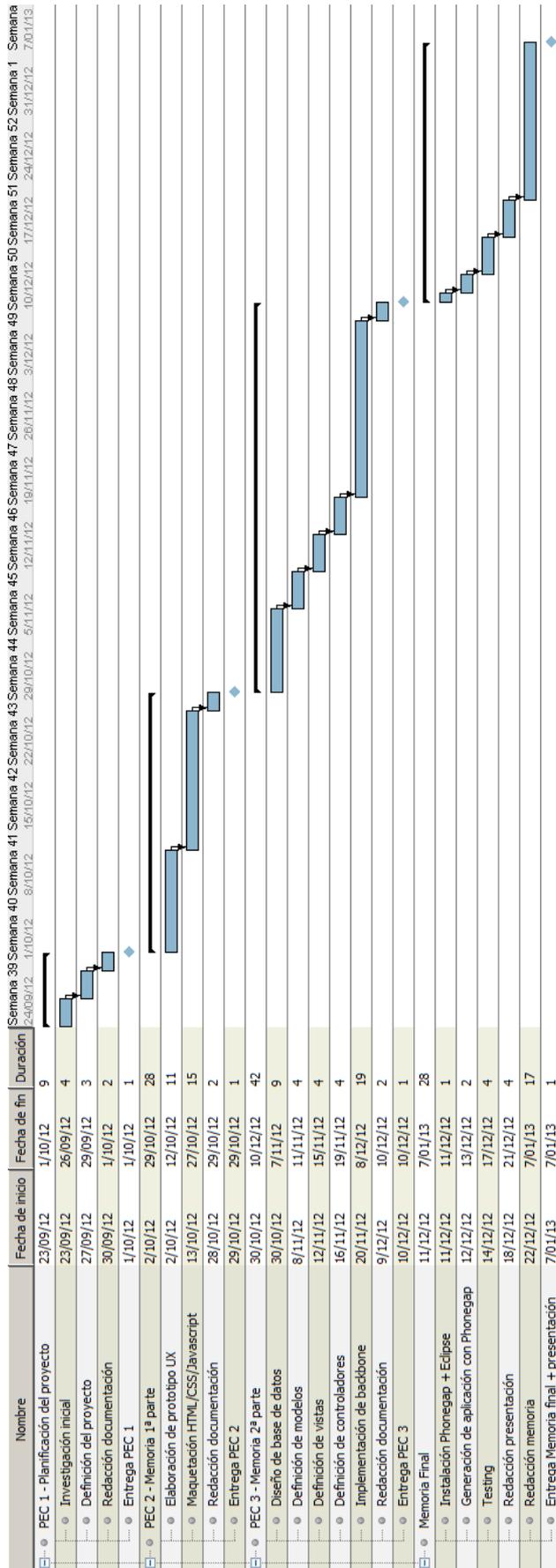
En esta última fase realizaremos la compilación final del proyecto utilizando PhoneGap, testaremos su funcionamiento, elaboraremos la presentación del proyecto y por último la memoria final del proyecto.

3. Calendario

3.1 Fechas de entrega

1. PEC 1: 1 de octubre
2. PEC 2: 29 de octubre
3. PEC 3: 10 de diciembre
4. Entrega final: 7 de enero
5. Debate virtual: del 21 al 24 de enero

3.2 Planificación temporal



4. Funcionalidades

4.1 Home

Pantalla inicial de la aplicación.



Figura 1 – Vista inicial prototipo

Esta es la pantalla principal de la aplicación. Aparecerá un texto de bienvenida y una pequeña descripción del funcionamiento de la aplicación.

Desde ella se accede a las funcionalidades de la aplicación mediante el menú principal situado en el pie.

Los tres accesos directos del pie permitirán la gestión de las listas, recetas e ingredientes necesarios en el funcionamiento de la app.

Este pie será común a todas las pantallas de la aplicación.

4.2 Ingredientes

Mediante el botón situado en el pie de la aplicación se accede a un listado de los ingredientes dados de alta en el sistema.

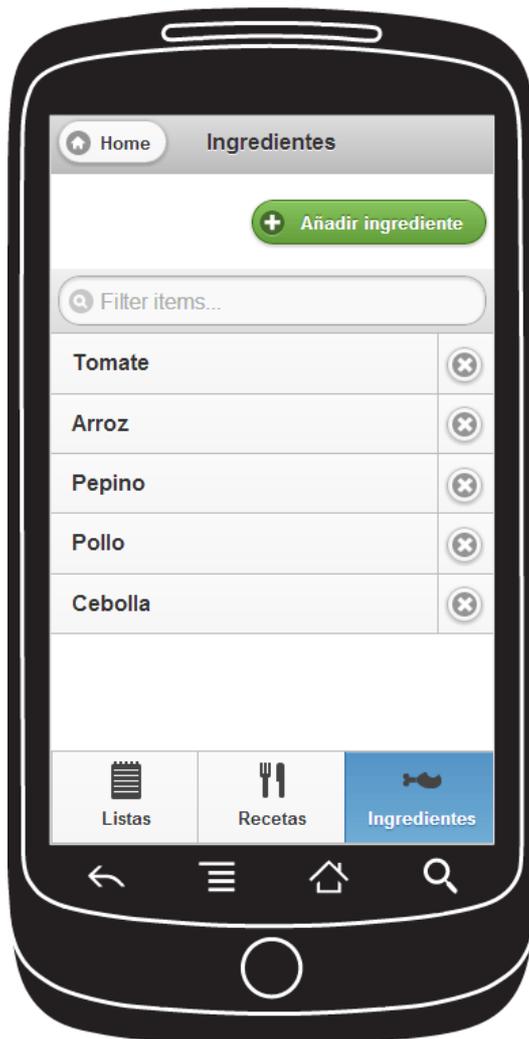


Figura 2 – Ingredientes prototipo

En esta misma pantalla, además de la visualización de los ingredientes de la aplicación, se accede a las funcionalidades de:

- Alta de nuevo ingrediente
- Baja de ingrediente
- Modificación de ingrediente

Estas funcionalidades aparecerán pulsando en los botones de “Añadir ingrediente”, imagen con aspa para borrar el ingrediente o pulsando directamente sobre el nombre realizaremos la modificación.

También se permitirá la posibilidad de filtrar los ingredientes a visualizar mediante un campo de texto.

Pantalla de filtrado:

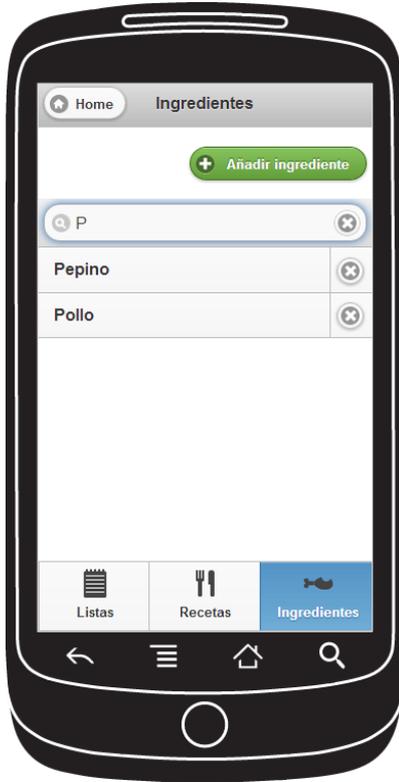


Figura 3 – Filtrado ingrediente prototipo

Pantalla de borrado:



Figura 4 – Borrado ingrediente prototipo

Pantalla de alta y edición:

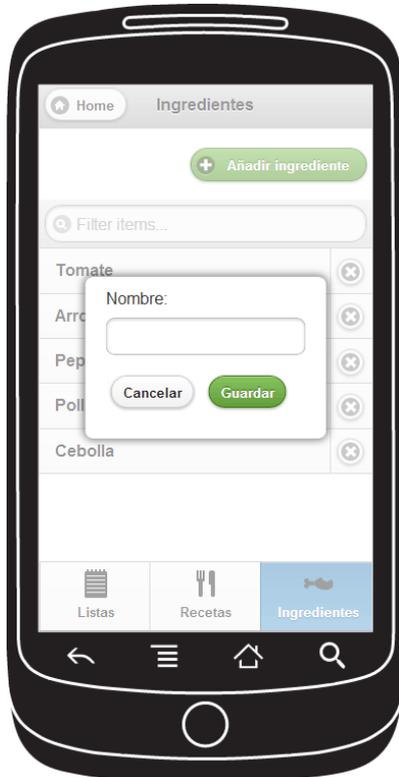


Figura 5 – Alta/edic. ingrediente prototipo

4.3 Consulta del catálogo de recetas

Pantalla de consulta del listado con todas las recetas disponibles en el sistema.



El acceso a esta pantalla se produce desde el botón correspondiente situado en el pie de la aplicación. En ella encontraremos además del catálogo de recetas los accesos a las funcionalidades de:

- Alta de nueva receta
- Baja de receta
- Detalle de receta

Figura 6 – Consulta recetas prototipo

4.4 Consulta detalle de receta

Pantalla de consulta del detalle de una receta

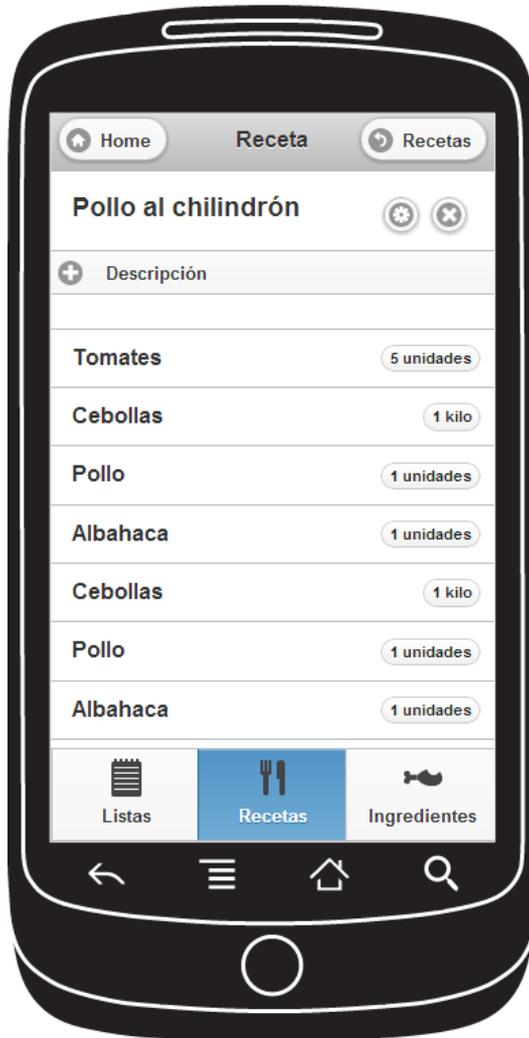


Figura 7 – Detalle receta prototipo

En esta pantalla podremos ver los ingredientes que componen la receta además de la cantidad necesaria para realizar la receta (y que hay que comprar). También podemos consultar una pequeña descripción de la receta.

A esta pantalla se llega desde la pantalla de consulta del catálogo de recetas y desde ella se puede acceder a las funcionalidades de borrado de la receta y modificación de la misma.

4.5 Alta de receta / edición de receta

Pantalla donde podremos dar de alta una nueva receta o modificar una ya existente.



Figura 8 – Alta/edic. receta prototipo

Los datos que podemos dar de alta o modificar de una receta son:

- Nombre
- Descripción
- Lista de ingredientes

El nombre y la descripción se modificará mediante sus correspondientes campos de formulario y los ingredientes podrán ser añadidos pulsando sobre el botón de “Añadir ingrediente”.

Si quisiéramos borrar algún ingrediente deberíamos pulsar en el aspa que aparece a la derecha del nombre. Y si quisiéramos modificar la cantidad del ingrediente tenemos que pulsar sobre el nombre del ingrediente. Aparecerá entonces una ventana pop-up que nos permitirá especificar la cantidad del ingrediente y tipo de medida.

Finalmente para guardar el alta o cambios realizados sobre la receta tendremos que pulsar en el botón de “Guardar”.

4.6 Alta de ingrediente a la receta

Pantalla donde añadiremos un ingrediente a la receta.

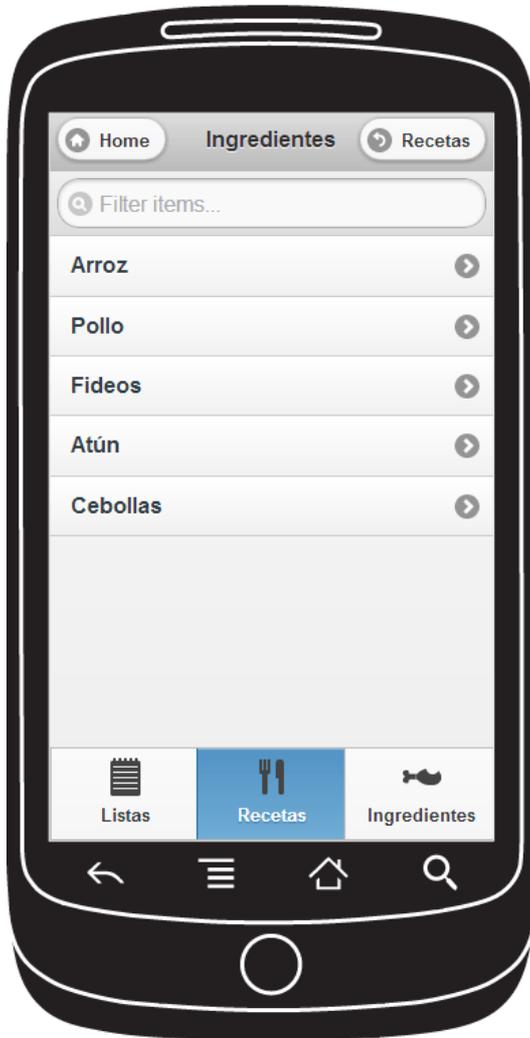


Figura 9 – Alta ingr. en receta prototipo

Llegamos a esta pantalla desde el alta o modificación de una receta. Para agregar el ingrediente pulsamos sobre él en el listado y aparecerá un pop-up en el que se preguntará sobre la cantidad y el tipo de cantidad del ingrediente (imagen inferior).

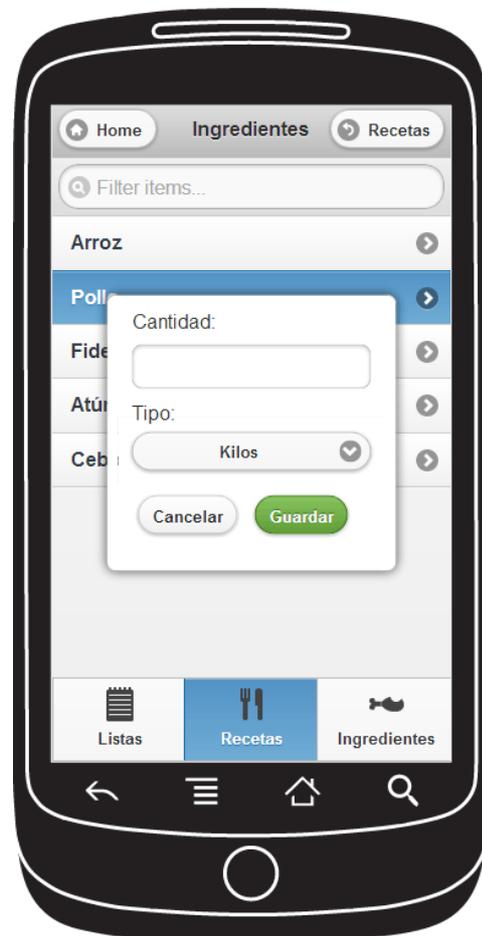


Figura 10 – Cantidad ingr. receta prototipo

4.7 Baja de ingrediente de la receta

Pantalla para quitar ingredientes de una receta.



Figura 11 – Baja ingr. receta prototipo

En la pantalla de edición o alta de una receta aparece el listado de ingredientes que la componen.

A la derecha del nombre del ingrediente aparece el botón (imagen en forma de aspa) para realizar la acción de borrado del ingrediente.

Al pulsar el botón aparecerá un pop-up pidiendo la confirmación de borrado del ingrediente.

4.8 Baja de receta

Pantalla de borrado de receta.



Figura 12 – Borrado de receta prototipo

El borrado de recetas se puede realizar desde dos pantallas diferentes:

- desde la pantalla que muestra el catálogo de recetas.
- en el detalle de una receta.

En los dos casos el botón de acción es el mismo (imagen con aspa).

Una vez pulsado el botón aparecerá una ventana modal o pop-up pidiendo confirmación del borrado.

4.9 Consulta del catálogo de listas de la compra

Pantalla de consulta del listado con todas las listas de la compra disponibles en el sistema.



En esta pantalla además de visualizar las listas de la compra (con la posibilidad de filtrado para ayudar en la búsqueda) se pueden realizar diferentes acciones de gestión de las listas de la compra como son el borrado de una lista, visualizar el detalle o añadir una nueva lista de la compra.

Figura 13 – Listas prototipo

4.10 Consulta de lista de la compra y proceso de compra

Pantalla donde visualizar los ingredientes de la lista la compra.

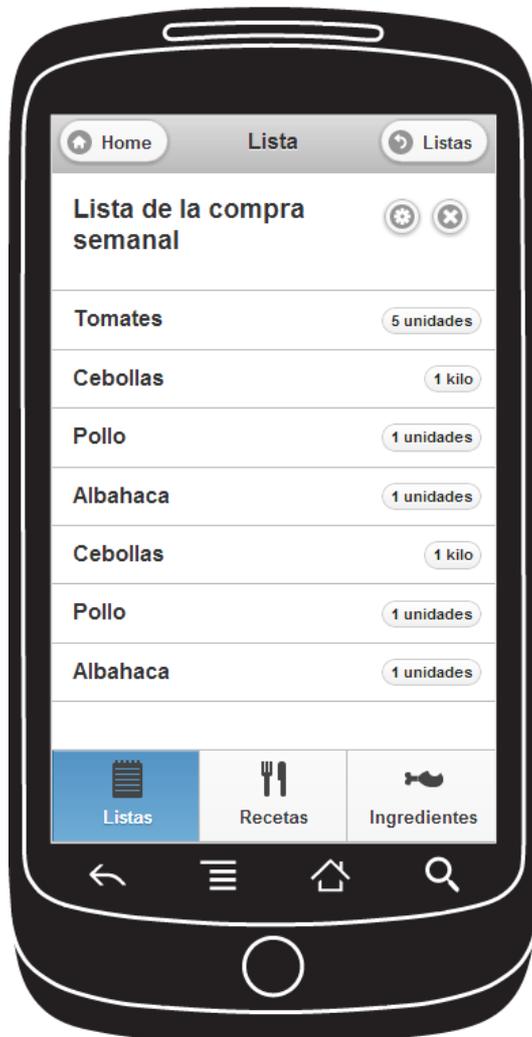


Figura 14 – Ingredientes en lista prototipo

Esta es la pantalla a usar cuando se está realizando una compra real. En ella aparecen listados todos los ingredientes que componen la receta seleccionada. Cuando se realiza la compra de un ingrediente la aplicación da la posibilidad al usuario de marcar la compra del ingrediente como realizada. Simplemente pulsando sobre el ingrediente aparecerá tachado y marcado con fondo que lo identificará como comprado.



Figura 15 – Selección ingrediente prototipo

En esta misma pantalla se da la posibilidad también de borrar o editar la lista de la compra.

4.11 Alta de lista de la compra / edición de lista de la compra

Pantalla donde editar o dar de alta una lista de la compra.



Figura 16 – Alta/edic. lista prototipo

Esta es la parte de la aplicación en la que se pondrá o modificará el nombre de una lista de la compra y además se añadirán las recetas y los ingredientes sueltos que se quieran incluir.

Las recetas e ingredientes aparecerán en dos bloques plegados para no saturar la pantalla de información. Si se quisiera añadir un receta (la operativa es la misma que para ingredientes) se pulsará en la barra de “Recetas” que desplegará un apartado donde se verán las recetas ya añadidas y un botón de “Añadir receta”. A la derecha de las recetas ya añadidas aparecerá el botón que permitirá borrarla de la lista.

4.12 Alta de receta a la lista de la compra

Pantalla donde se añade una receta a la lista de la compra.



Figura 17 – Alta receta en lista prototipo

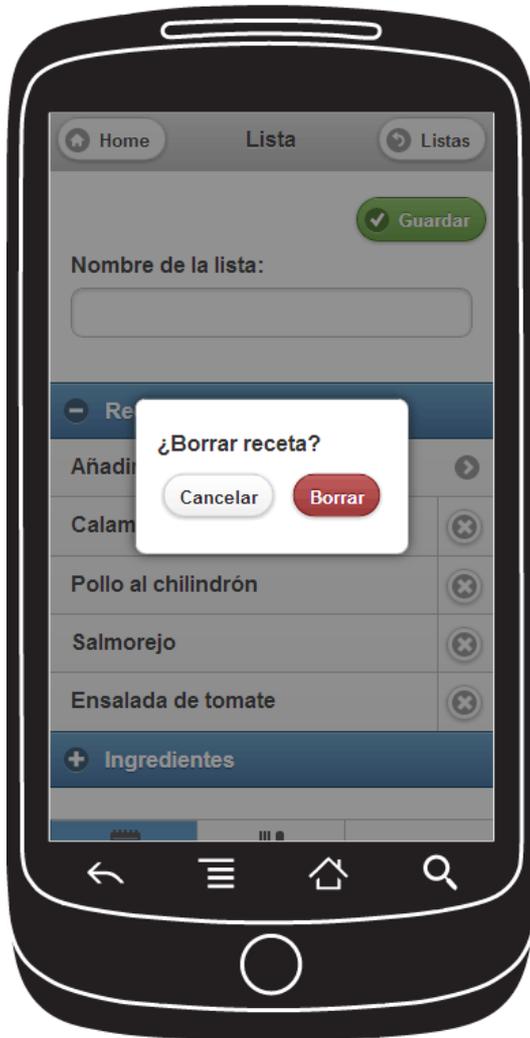
Esta es la pantalla donde te diriges después de pulsar “Añadir receta” en la funcionalidad anterior.

Es una pantalla muy simple donde aparecerá un listado de recetas que se pueden seleccionar para añadir a la lista de la compra que se está creando o editando.

Se da también la posibilidad de filtrar las recetas para ayudar en la búsqueda de la receta elegida.

4.13 Baja de receta de la lista de la compra

Funcionalidad de borrado de una receta.



Para borrar una receta de una lista de la compra, únicamente hay que pulsar en el icono de borrado que aparece a la derecha del nombre de cada receta. Una vez pulsado saltará un pop-up de confirmación de borrado.

Figura 18 – Baja de receta en lista prototipo

4.14 Alta de ingrediente a la lista de la compra

Pantalla donde se añade un ingrediente a la lista de la compra.

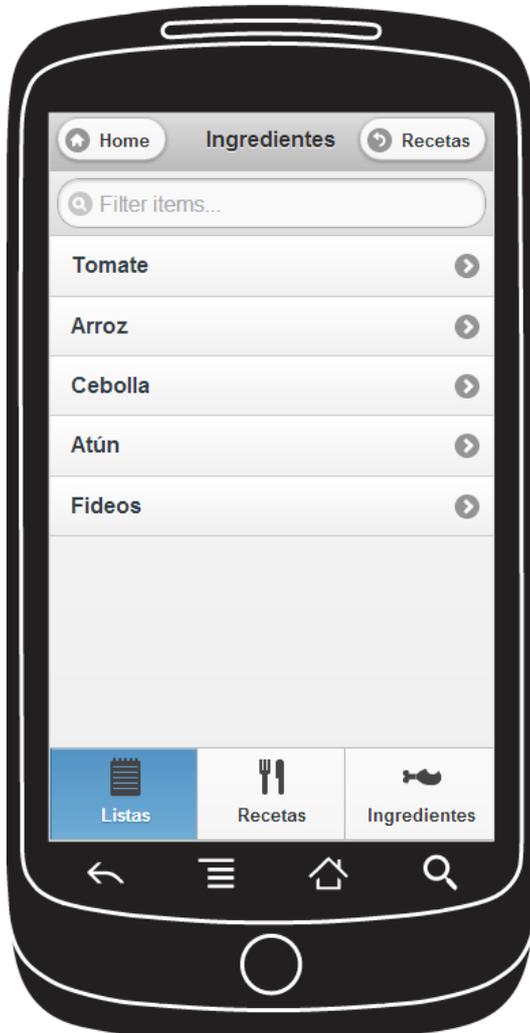


Figura 19 – Alta ingr. En lista prototipo

La operativa de esta pantalla es idéntica a la anterior: se llega desde la pantalla de creación de una lista de la compra y seleccionas el ingrediente pulsando sobre el. Una vez pulsado aparecerá un pop-up para indicar la cantidad y su tipo del ingrediente seleccionado.



Figura 20 – Cantidad ingr. lista prototipo

4.15 Baja de ingrediente de la lista de la compra

Pantalla donde dar de baja un ingrediente de la lista de la compra.



Si se quisiera borrar un ingrediente de la lista de la compra únicamente habría que pulsar sobre el icono de borrado que aparece a la derecha del ingrediente y confirmar el borrado en el pop-up que aparece.

Figura 21 – Baja ingr. de lista prototipo

4.16 Baja de la lista de la compra

Borrado de una lista de la compra.

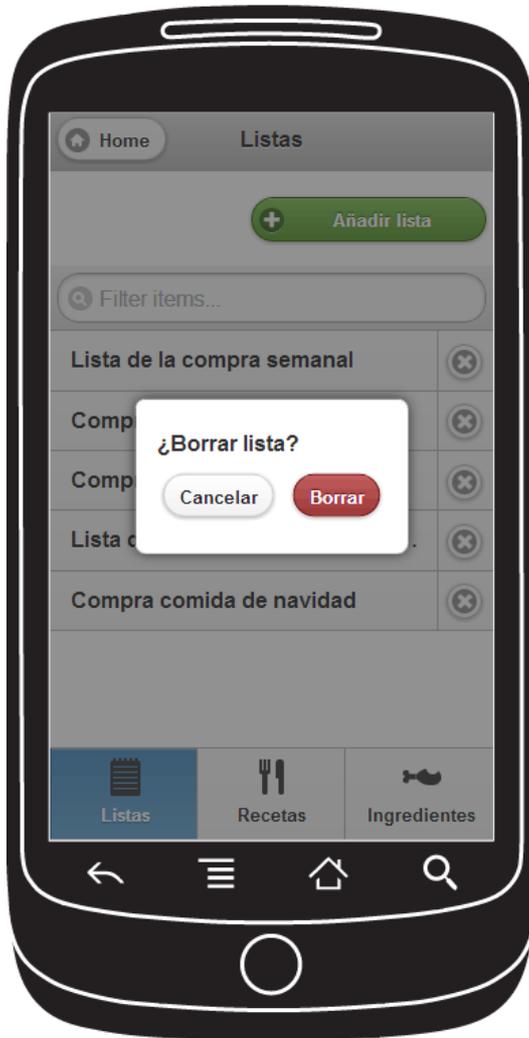
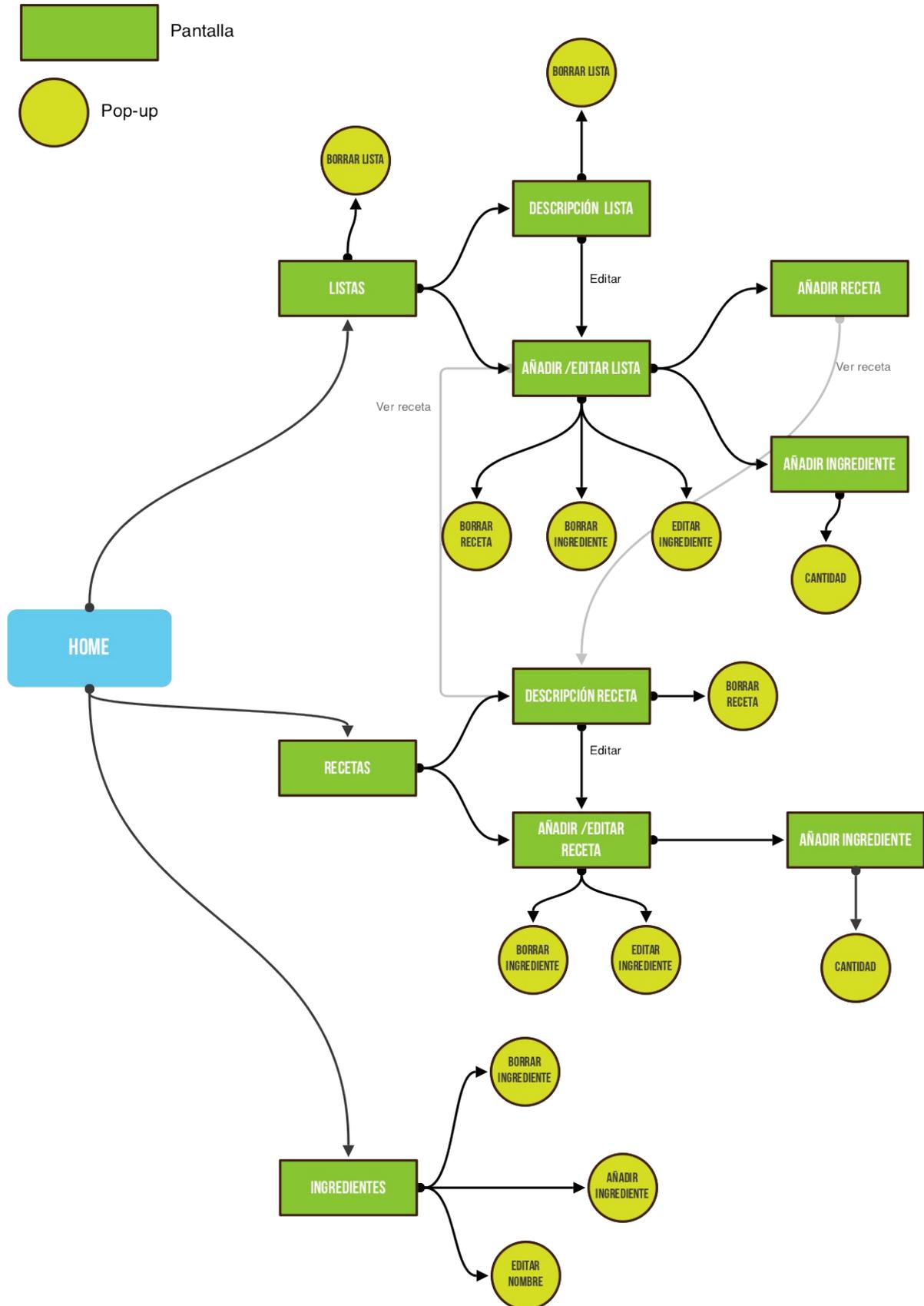


Figura 22 – Baja de lista prototipo

Cuando se quiera realizar el borrado de una lista de la compra se podrá realizar esta acción desde dos pantallas diferentes: directamente desde el listado de listas de la compra o desde el detalle de la lista de la compra. En los dos casos aparece un botón de borrado (en forma de aspa) que llama a la acción. Una vez pulsado aparecerá un pop-up de confirmación de borrado de la lista.

5. Mapa de la aplicación



6. Casos de uso

Caso de uso: añadir ingrediente

Resumen de la funcionalidad: gestión por la cual se añaden ingredientes que serán usados luego en las recetas.

Actores: usuario de la aplicación

Casos de uso relacionados: no tiene

Precondición: no tiene

Postcondición: el ingrediente es añadido a la base de datos

Proceso normal principal:

1. El usuario entra en la opción del menú superior "Recetas"
2. El usuario pulsa sobre el botón "Gestionar ingredientes"
3. El usuario introduce el nombre de ingrediente en el campo correspondiente
4. El usuario pulsa en el botón "Añadir"
5. El sistema graba el nombre del ingrediente en la base de datos

Caso de uso: añadir receta

Resumen de la funcionalidad: gestión por la cual se añaden recetas que serán usadas luego en las listas de la compra.

Actores: usuario de la aplicación

Casos de uso relacionados: añadir ingrediente

Precondición: deben existir ingredientes añadidos previamente

Postcondición: la receta es añadida a la base de datos

Proceso normal principal:

1. El usuario entra en la opción del menú superior "Recetas"
2. El usuario pulsa sobre el botón "Añadir receta"
3. El usuario introduce el nombre de la receta y su descripción en los campos correspondientes

4. El usuario pulsa en el botón “Añadir” para indicar los ingredientes usados en la receta
5. El usuario selecciona los ingredientes a usar indicando opcionalmente la cantidad y el tipo
6. El usuario confirma los ingredientes pulsando en el botón de confirmación
7. El usuario pulsa en el botón “Guardar”
8. El sistema graba la receta en la base de datos

Alternativas de proceso y excepciones:

- 5.a No hay ingredientes guardados en el sistema

Caso de uso: añadir lista

Resumen de la funcionalidad: gestión por la cual se añaden listas de la compra.

Actores: usuario de la aplicación

Casos de uso relacionados: añadir receta

Precondición: deben existir recetas añadidas previamente

Postcondición: la lista es añadida a la base de datos

Proceso normal principal:

1. El usuario entra en la opción del menú superior “Listas”
3. El usuario introduce el nombre de la lista en el campo correspondiente
4. El usuario pulsa en el botón “Añadir”
5. El usuario pulsa en el botón “Modificar recetas” para indicar las recetas usadas en la lista
6. El usuario confirma las recetas pulsando en el botón de confirmación
7. El usuario pulsa en el botón “Guardar”
8. El sistema graba la lista en la base de datos

Alternativas de proceso y excepciones:

- 5.a No hay recetas guardadas en el sistema

Caso de uso: eliminar ingrediente

Resumen de la funcionalidad: gestión por la cual se eliminan ingredientes

Actores: usuario de la aplicación

Casos de uso relacionados: añadir ingrediente

Precondición: deben existir ingredientes añadidos previamente

Postcondición: el ingrediente es eliminado de la base de datos

Proceso normal principal:

1. El usuario entra en la opción del menú superior “Recetas”
2. El usuario pulsa sobre el botón “Gestionar ingredientes”
3. El usuario pulsa sobre el botón “Borrar ingredientes”
4. El usuario pulsa sobre el botón con icono de aspa para borrar el ingrediente
5. El sistema borra el ingrediente en la base de datos

Caso de uso: eliminar receta

Resumen de la funcionalidad: gestión por la cual se eliminan recetas

Actores: usuario de la aplicación

Casos de uso relacionados: añadir receta

Precondición: deben existir recetas añadidas previamente

Postcondición: la receta es eliminada de la base de datos

Proceso normal principal:

1. El usuario entra en la opción del menú superior “Recetas”
2. El usuario pulsa sobre el botón con icono de aspa para borrar la receta
3. El sistema borra la receta de la base de datos

Caso de uso: eliminar lista

Resumen de la funcionalidad: gestión por la cual se eliminan listas

Actores: usuario de la aplicación

Casos de uso relacionados: añadir lista

Precondición: deben existir listas añadidas previamente

Postcondición: la lista es eliminada de la base de datos

Proceso normal principal:

1. El usuario entra en la opción del menú superior “Listas”
2. El usuario pulsa sobre el botón con icono de aspa para borrar la lista
3. El sistema borra la lista de la base de datos

Caso de uso: modificar receta

Resumen de la funcionalidad: gestión por la cual se modifican los datos de una receta

Actores: usuario de la aplicación

Casos de uso relacionados: añadir receta

Precondición: deben existir recetas añadidas previamente

Postcondición: la receta es modificada en la base de datos

Proceso normal principal:

1. El usuario entra en la opción del menú superior “Recetas”
2. El usuario pulsa sobre la receta que quiere modificar
3. El usuario modifica los datos de la receta
4. El usuario pulsa en el botón “Guardar”
5. El sistema graba la receta en la base de datos

Caso de uso: modificar lista

Resumen de la funcionalidad: gestión por la cual se modifican los datos de una lista

Actores: usuario de la aplicación

Casos de uso relacionados: añadir lista

Precondición: deben existir listas añadidas previamente

Postcondición: la lista es modificada en la base de datos

Proceso normal principal:

1. El usuario entra en la opción del menú superior “Listas”
2. El usuario pulsa sobre la lista que quiere modificar
3. El usuario modifica los datos de la lista
4. El usuario pulsa en el botón “Guardar”
5. El sistema graba la receta en la base de datos

Caso de uso: usar lista para realizar la compra

Resumen de la funcionalidad: gestión por la cual se modifican los datos de una lista

Actores: usuario de la aplicación

Casos de uso relacionados: añadir lista

Precondición: deben existir listas añadidas previamente

Postcondición: no tiene

Proceso normal principal:

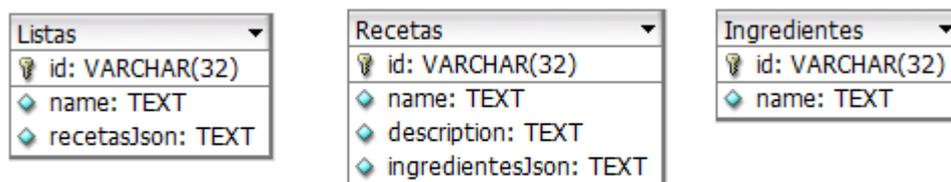
1. El usuario entra en la opción del menú superior “Listas”
2. El usuario pulsa sobre la lista que quiera usar
3. El usuario pulsa sobre el ingrediente del que ha realizado la compra
4. El sistema marca el ingrediente como comprado

7. Diseño de la persistencia

Para la persistencia de los datos de la aplicación se utiliza un modelo mixto de base de datos relacional con archivos json almacenados en campos de la base de datos. La aplicación hace uso de la base de datos implementada en phonegap la cual esta basada en la especificación de la W3C Web SQL Database. También usa la librería de javascript persistence.js como ayuda en el manejo de la base de datos, permitiendo hacer más fácil el mapeado de las entradas de la base de datos en objetos javascript y trabajar asincrónamente con la base de datos.

El diagrama de tablas de la base de datos es bastante simple siendo el diagrama de tablas y el modelo relacional de la base de datos los siguientes:

6.1 Diagrama de tablas de la base de datos



6.2 Modelo relacional de la base de datos

Listas (id, name, recetasJson)

Recetas (id, name, description, ingredientesJson)

Ingredientes (id, name)

Los ids usados como claves de las tablas son generados por la librería persistence.js mediante la generación automática de un UUID (identificador universalmente único) el cual tiene la ventaja de permitir la sincronización con otras bases de datos sin afectar a la unicidad requerida en las claves primarias de las tablas que se sincronizan.

6.3 Modelo de los json utilizados

En la tabla *listas* el campo *recetasJson* contiene archivos json que siguen este modelo: [{id, name},{id, name}...]

Ejemplo:

```
[  
  {id:"8D14770D3AAA4601818E4B2BF1E31FE4 ", name:"Pollo al chilidrón"},  
  {id:"4228BCF040C947C99059CA7DE4D525ED ", name:"Arroz a la cubana"}  
]
```

En la tabla *recetas* el campo *ingredientesJson* contiene archivos json que siguen este modelo: [{id, name,tipo,cantidad},{id, name,tipo,cantidad}...]

Ejemplo:

```
[  
  {id:"8D14770D3AAA4601818E4B2BF1E31FE4 ", name:"Pollo", tipo:"unidades",  
cantidad:"1"},  
  {id:"4228BCF040C947C99059CA7DE4D525ED ", name:"Tomate", tipo:"kilos",  
cantidad:"1"}  
]
```

8. Implementación

Durante el desarrollo de la aplicación he realizado cambios importantes sobre lo proyectado en un principio. Estos cambios son motivados principalmente a las pruebas realizadas sobre el prototipo realizado que me han llevado a modificar sustancialmente dos aspectos básicos de la aplicación: la arquitectura de la información y la tecnología a utilizar. Estos cambios entran dentro de la evolución normal de producción de software en su aspecto de desarrollo incremental iterativo.

8.1 Modificaciones en la arquitectura de la información

El desarrollo del prototipo ha permitido hacer una demo mediante PhoneGap totalmente apta para ser instalada en un móvil android. Esto ha hecho posible probar la aplicación en el dispositivo final y además por varios usuarios. Como resultado de las pruebas se llegó a la conclusión de que no cumplía con las expectativas de facilidad de uso. Aunque la estructura era bastante lógica, adolecía de un exceso de número de pantallas y cierta complejidad en el manejo.

Debido a los errores de usabilidad encontrados se decidió modificar completamente la arquitectura eliminando pantallas y una sección completa del menú: ingredientes. La funcionalidad de gestión de ingredientes estaba en un primer nivel, el mismo que funcionalidades mucho más importantes como son la gestión de las listas o las recetas. Se ha eliminado por lo tanto esta opción del menú y se ha optado por realizar la gestión de los ingredientes como un apartado de la sección de gestión de recetas.

El resultado de la refactorización de la arquitectura se muestra en las siguientes figuras:

Pantalla usada en un proceso de compra donde se muestran los ingredientes a comprar permitiendo marcarlos como ya comprados.

Pantalla de alta o modificación de las recetas incluidas en una lista de la compra.



Figura 23 – Proceso de compra - app

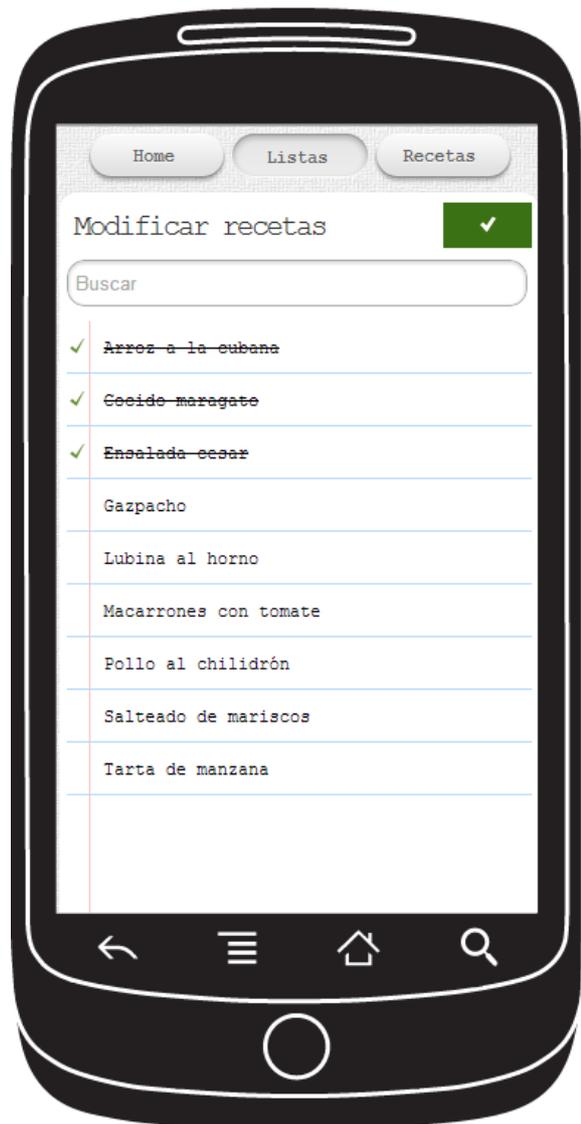


Figura 24 – Alta receta en lista - app

Pantalla de inserción o modificación de una lista de la compra.

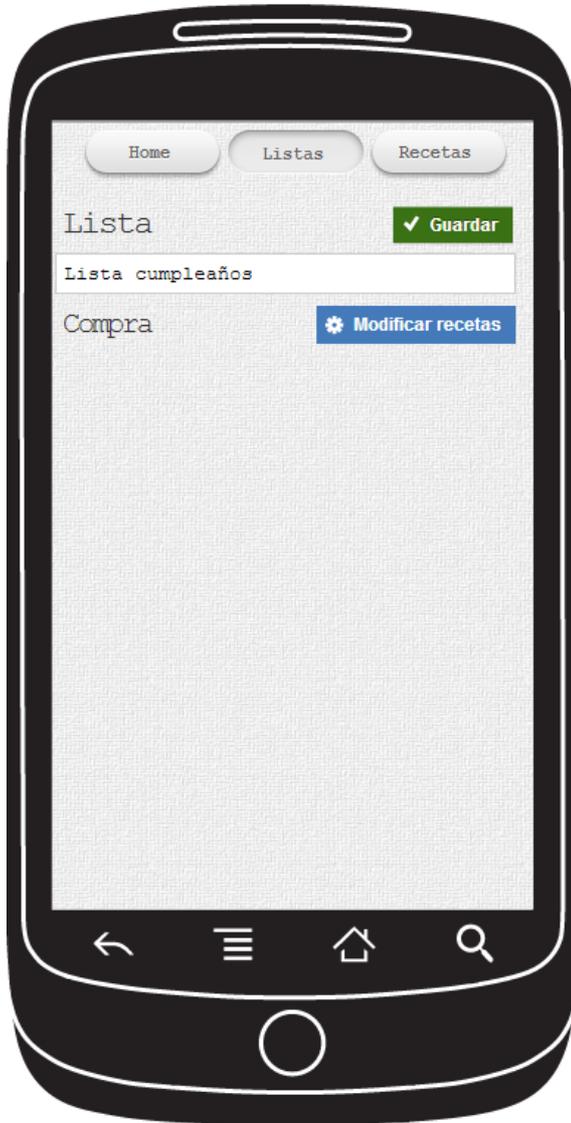


Figura 25 – Alta/modificación lista - app

Pantalla donde se listan las recetas de la aplicación.



Figura 26 – Lista de recetas - app

Pantalla de alta o edición de una receta. Pantalla donde agregar ingredientes a una receta.



Figura 27 – Alta/edición receta - app

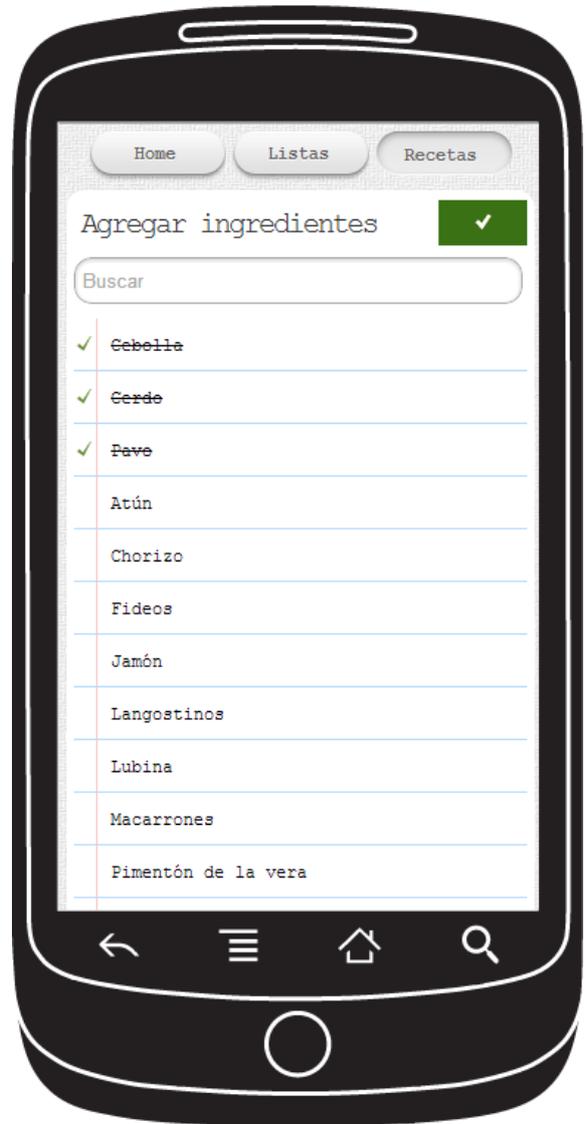


Figura 28 – Alta ingr. En receta - app

Pantalla de alta de un ingrediente.

Pantalla de borrado de ingredientes.

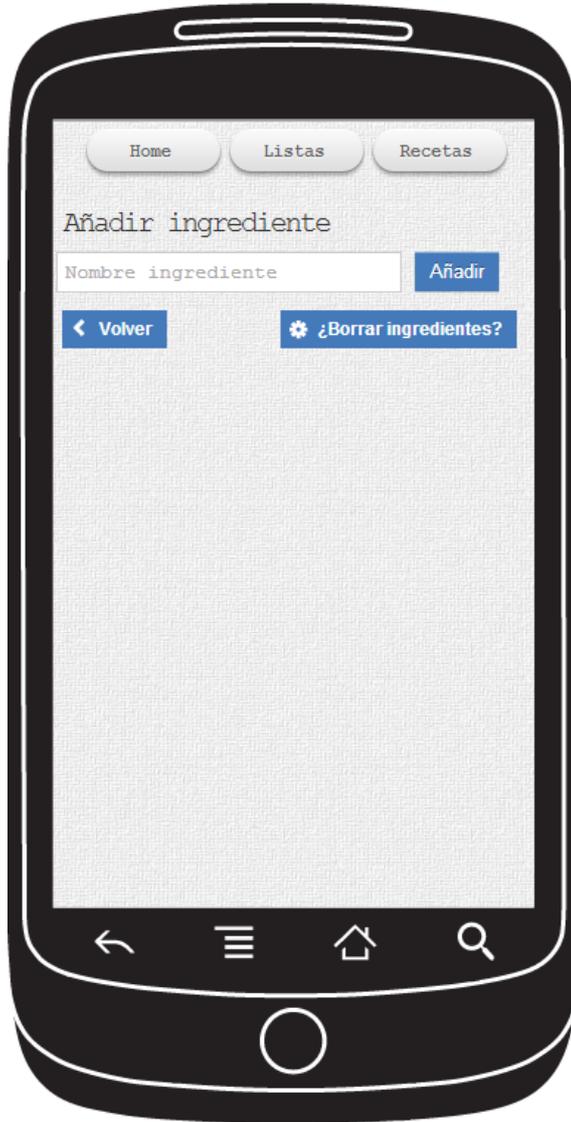


Figura 29 – Alta ingrediente - app

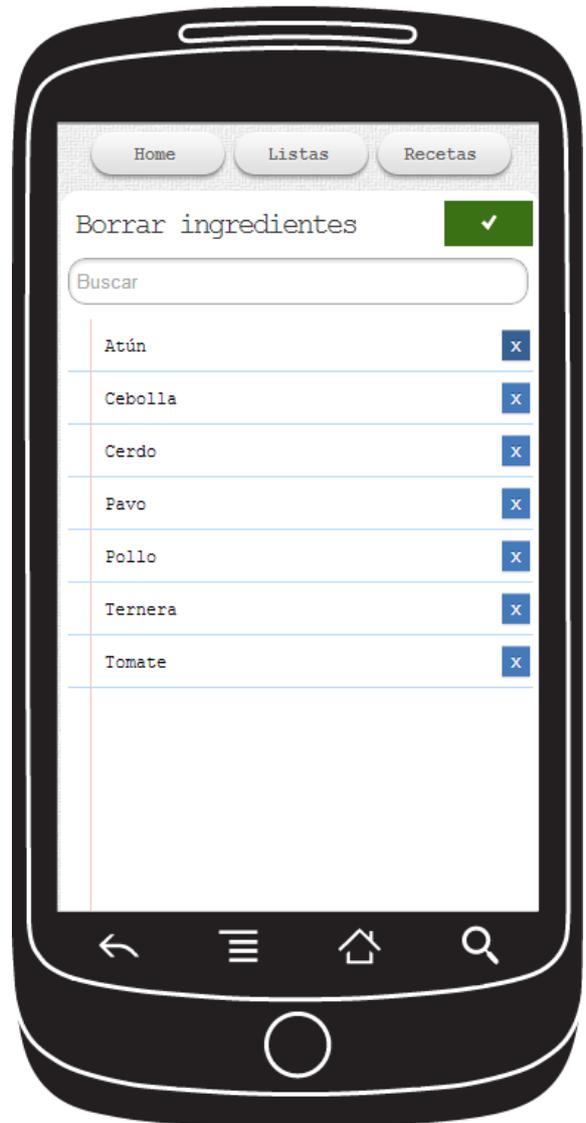


Figura 30 – Baja ingrediente - app

8.2 Modificaciones en la tecnología usada en la aplicación

Para el desarrollo del prototipo se usaron las librerías javascript jQuery y jQuery Mobile. En un principio la elección parecía la mas lógica ya que jQuery Mobile está pensada específicamente para los dispositivos móviles pero el resultado no fue el esperado en un móvil como el mio con la versión de Android 2.2.2. La aplicación resultaba en exceso lenta y con una sensación de que el movimiento entre las diferentes pantallas no era lo suficientemente fluida.

Por todo ello se decidió no usar jQuery, jQuery Mobile u otra librería javascript que ofreciera ayuda para la creación de la interfaz de usuario pero que finalmente penalizara la experiencia de uso en móviles que no sean de última generación o alta gama.

Para el desarrollo de la app se decidió seguir el modelo de desarrollo de una SPA (Single-page Application) para lo que pensaba utilizar la librería backbone.js. Con el cambio de planteamiento y debido a la dependencia que tiene esta librería respecto a jQuery se busco otra alternativa encontrandola en la librería desarrollada por Google angularjs.

8.3 Angularjs

Angularjs es un framework para el desarrollo de aplicaciones siguiendo el patrón de desarrollo MVC. Otra propiedad interesante de angularjs son que implementan un pequeño API para manipular el DOM: jqLite que permite prescindir de jQuery u otra librería de las mismas características. Además los métodos usados por jqLite tienen correspondencia con los métodos de jQuery con la misma funcionalidad, teniendo el mismo nombre y manera de usarlos.

Angularjs tiene otra característica que lo hace especial: la posibilidad de usar nuevos atributos o elementos HTML que aumentan la capacidad del marcado permitiendo nuevas funcionalidades como una conexión directa entre nuestro modelo de datos y el código HTML para que sin tener que realizar ninguna acción especial por nuestra parte el simple cambio del modelo se vea reflejado directamente en la vista del navegador.

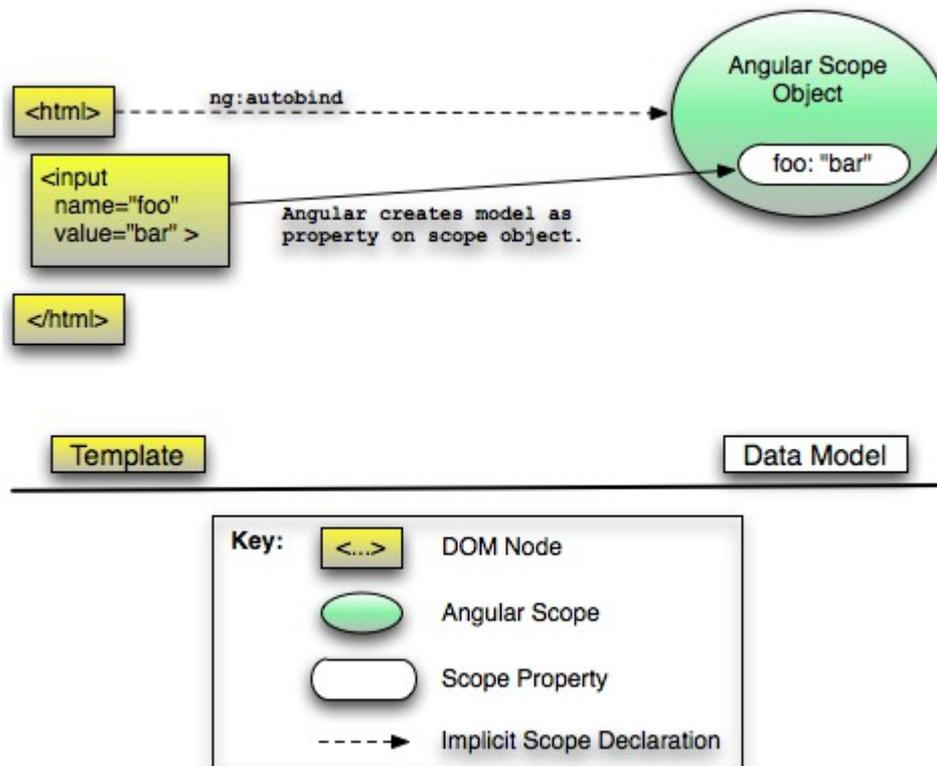
El modelo (del patrón MVC) en angularjs esta definido por cualquier dato accesible desde el objeto scope de angular. El tipo de datos puede ser un objeto, primitiva o array. Por ejemplo:

```
$scope.nombre = "Antonio"; //string
$scope.edad = 25; //number
$scope.persona = {nombre:" Antonio", edad:25}; //object
$scope.nombre = [{nombre:" Antonio", edad:25},{nombre:"Pedro", edad:30}];
//array
```

Angular también puede crear modelos implícitamente en los elemento de formulario HTML . Por ejemplo:

```
<input ng-model="nombre" value="Andrés"> sería equivalente a
$scope.nombre = "Andrés";
```

La siguiente imagen muestra lo explicado anteriormente:



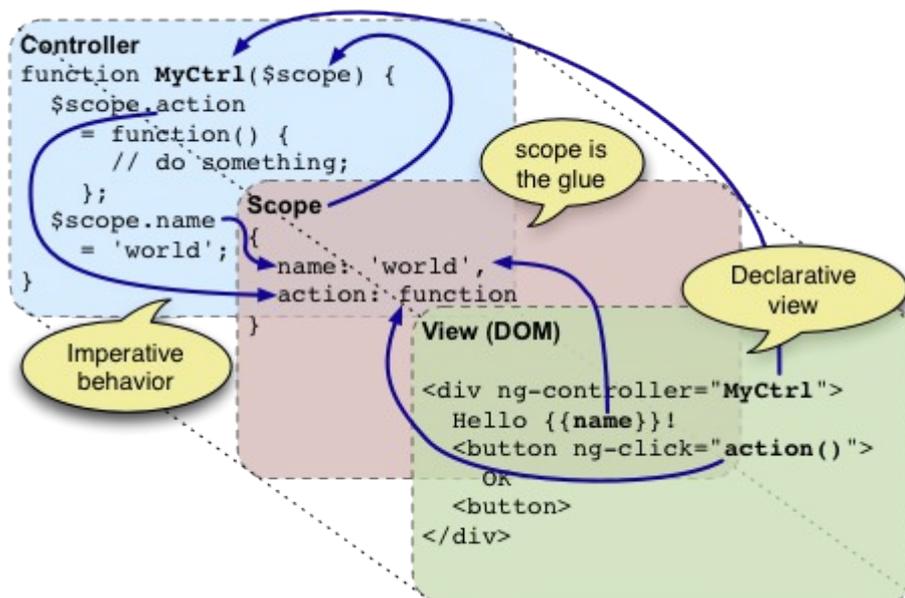
El controlador en angularjs es usado para inicializar el estado de objeto scope y añadirle comportamientos. Ejemplo:

```

<!-- html -->
<div ng-controller="datosCtrl">
  <input ng-model="nombre" value="">
  <p>{{ edad }}</p>
  <p>Años para la jubilación: {{ jubilacion }}</p>
</div>

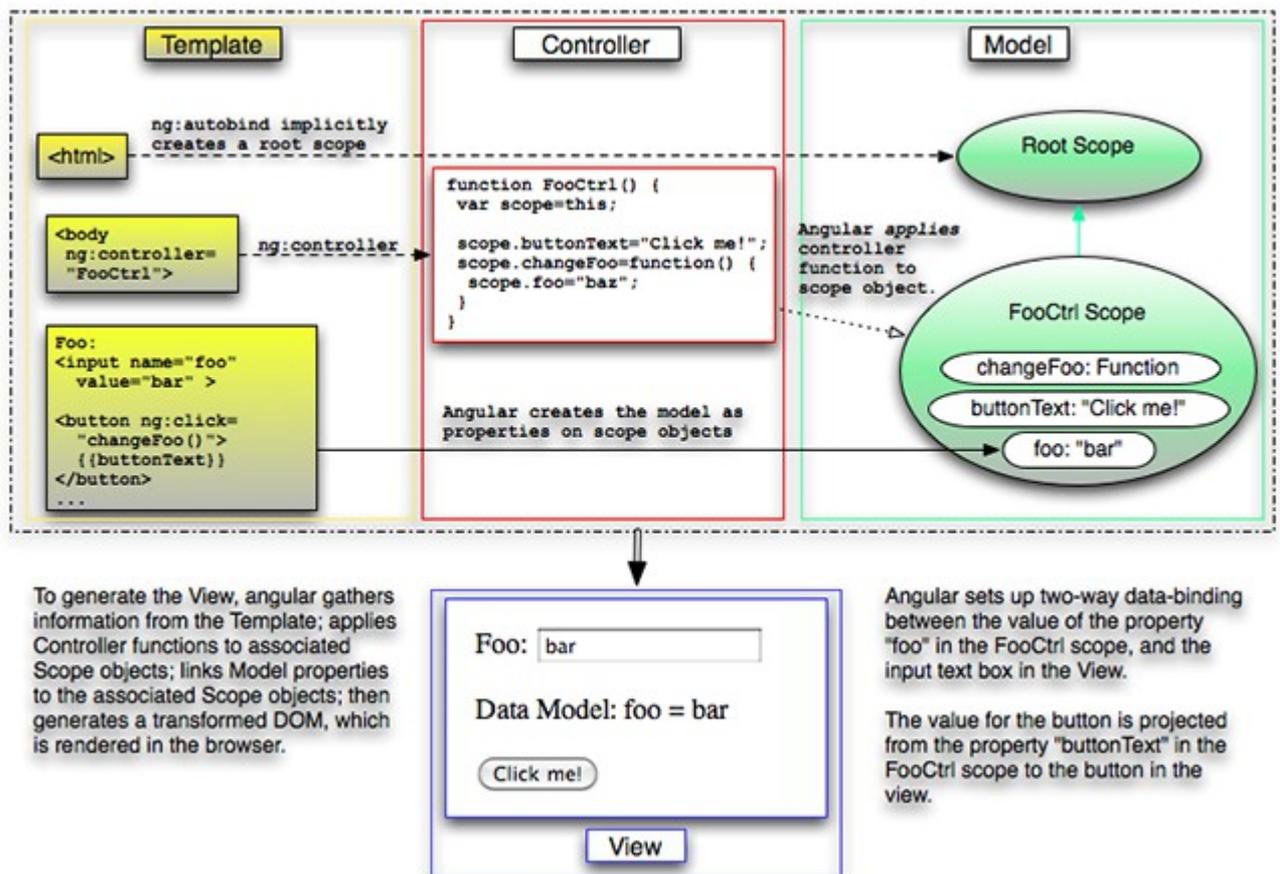
//javascript
function datosCtrl($scope) {
  $scope.nombre = 'Manuel';
  $scope.edad = 32;
  $scope.jubilacion = 65-32;
}

```



Por ultimo las vistas en angularjs son el propio DOM de la página HTML una vez que se ha cargado y renderizado por el browser. Dentro del HTML es posible usar datos del modelo o funciones del controlador usando la forma {{ }} popularizada por el motor de plantillas moustache.

Un aspecto importante de angularjs es que la relación ente el modelo y la vista es “two-way data-binding”. Cualquier cambio en el modelo se vera reflejado en la vista y viceversa.



9. Glosario

HTML5: versión quinta del lenguaje de marcado HTML el cual es la base más importante para la construcción de páginas web

CSS3: versión tercera del lenguaje usado para definir la presentación visual de los documentos marcados con HTML

Javascript: lenguaje de programación interpretado, usado principalmente en las páginas web para permitir realizar mejoras en la interacción con la página.

W3C: acrónimo de la World Wide Web Consortium, organismo internacional dedicada a la estandarización de tecnologías usadas en la web.

MVC: Modelo-vista-controlador es un patrón de desarrollo de software que hace hincapié en la separación en tres componentes distintos los datos, la interfaz de usuario y la lógica de negocio.

SQL: lenguaje declarativo de acceso a base de datos relacionales que permite realizar operaciones de consulta, borrado, etc.

WebSQL: API usado en navegadores web para almacenar datos y realizar consultas con una sintaxis muy parecida a la utilizada por SQL.

Prototipo: modelo realizado para mostrar como será un producto final de software.

API: acrónimo en inglés de Application Programming Interface. Es el conjunto de propiedades y métodos que ofrece una aplicación para ser utilizado por otra.

10. Fuentes consultadas

Para la realización de este trabajo he realizado consultas a las siguientes fuentes bibliográficas y direcciones de páginas web:

- **Object-Oriented Javascript**
Stoyan Stefanov
Packt Publishing – Julio 2008
- **20 Recipes for Programing Phonegap**
Jamie Munro
O’Reilly – 2012
- **Mobile Javascript Application Development**
Adrian Kosmaczewski
O’Reilly – 2012
- **Persistence.js**
<http://persistencejs.org/>
<https://github.com/zefhemel/persistencejs>
<http://jacobmumm.com/2011/09/20/asynchronous-javascript-with-persistencejs/>
- **Phonegap**
http://docs.phonegap.com/en/2.2.0/guide_getting-started_android_index.md.html#Getting%20Started%20with%20Android
<https://build.phonegap.com/docs/config-xml>
- **Angular**
<http://angularjs.org/>
<http://docs.angularjs.org/api/>
<http://coenraets.org/blog/2012/02/sample-application-with-angular-js/>
- **Otros**
<http://stackoverflow.com/>

11. Índice de figuras

Figura 1 – Vista inicial prototipo.....	11
Figura 2 – Ingredientes prototipo.....	12
Figura 3 – Filtrado ingrediente prototipo.....	13
Figura 4 – Borrado ingrediente prototipo.....	13
Figura 5 – Alta/edic. ingrediente prototipo.....	13
Figura 6 – Consulta recetas prototipo.....	14
Figura 7 – Detalle receta prototipo.....	15
Figura 8 – Alta/edic. receta prototipo.....	16
Figura 9 – Alta ingr. en receta prototipo.....	17
Figura 10 – Cantidad ingr. receta prototipo.....	18
Figura 11 – Baja ingr. receta prototipo.....	19
Figura 12 – Borrado de receta prototipo.....	20
Figura 13 – Listas prototipo.....	21
Figura 14 – Ingredientes en lista prototipo.....	21
Figura 15 – Selección ingrediente prototipo.....	21
Figura 16 – Alta/edic. lista prototipo.....	22
Figura 17 – Alta receta en lista prototipo.....	23
Figura 18 – Baja de receta en lista prototipo.....	24
Figura 19 – Alta ingr. En lista prototipo.....	25
Figura 20 – Cantidad ingr. lista prototipo.....	25
Figura 21 – Baja ingr. de lista prototipo.....	26
Figura 22 – Baja de lista prototipo.....	27
Figura 23 – Proceso de compra - app.....	37
Figura 24 – Alta receta en lista - app.....	37
Figura 25 – Alta/modificación lista - app.....	38
Figura 26 – Lista de recetas - app.....	38
Figura 27 – Alta/edición receta - app.....	39
Figura 28 – Alta ingr. En receta - app.....	39
Figura 29 – Alta ingrediente - app.....	40
Figura 30 – Baja ingrediente - app.....	40