

MORATALAZ

EFA MORATALAZ

Departamento de informática

GRADO SUPERIOR EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Proyecto de desarrollo de aplicaciones multiplataforma

Quesoteca - El mundo Frontend

Autor: Juan Manuel Ros Márquez

Tutor: Ángel Caballero

RESUMEN

En este proyecto titulado “Quesoteca - El mundo Frontend” veremos el desarrollo de dos aplicaciones, una web y otra móvil.

¿Por qué dos aplicaciones?. Se han desarrollado dos aplicaciones para que la experiencia del usuario sea la mas cómoda y gratificante utilizando las tecnologías oportunas para cada sector aprovechando así el potencial de ambas plataformas.

Veremos la evolución de una idea creativa e innovadora de cómo nace, como se va desarrollando y cual es el resultado de un duro proceso de investigación y desarrollo ya que las tecnologías utilizadas en Quesoteca no tienen mucho tiempo en el mercado del desarrollo por lo que el proceso de búsqueda de información será muy diferente al de otras tecnologías que estén implantadas ya mucho tiempo en el mercado.

Las dos plataformas creadas han sido creadas en su plena plenitud sin partir de ningún tipo de plantilla haciendo así una de las fases mas duras la del diseño. Se explicará así el motivo de elegir una tipografía y su elección de una paleta de colores.

Por ultimo, comentar que el desarrollo se ha hecho desde los principios de Clean Code y este a su vez es totalmente modular dando la posibilidad al desarrollador de cambiar la cabecera de la web con muy poco esfuerzo.

ÍNDICE

1	INTRODUCCIÓN	1
1.1	PLANTEAMIENTO DEL PROBLEMA	2
1.2	OBJETIVOS.....	2
1.3	METODOLOGÍA.....	3
2	DESCRIPCIÓN DEL PROBLEMA	4
3	EL PROYECTO SOFTWARE.....	5
3.1	APLICACIÓN WEB.....	5
3.1.1	ASPECTOS DEL DISEÑO	11
3.1.2	ASPECTOS TÉCNICOS	12
3.2	APLICACIÓN MÓVIL.....	16
3.2.1	ASPECTOS DEL DISEÑO	22
3.2.2	ASPECTOS TÉCNICOS	23
4	SOLUCIÓN	25
4.1	SOLUCIÓN – APLICACIÓN WEB.....	25
4.2	SOLUCIÓN – APLICACIÓN MÓVIL	30
5	EVALUACIÓN.....	32
5.1	ESTADÍSTICAS APLICACIÓN WEB	33
5.2	ESTADÍSTICAS APLICACIÓN MÓVIL.....	34
5.3	PUNTOS FUERTES / PUNTOS A MEJORAR.....	35
6	CONCLUSIÓN	36
7	LISTA DE REFERENCIAS	36

ÍNDICE DE FIGURAS

- Figura 1 - Frontend vs Backend
- Figura 1.3 - SCRUM
- Figura 3.1 - HTML / CSS / JS
- Figura 3.1 - SASS
- Figura 3.1 - Comparación de código SASS - CSS
- Figura 3.1 - Bootstrap
- Figura 3.1 - jQuery
- Figura 3.1.1 Paleta de colores
- Figura 3.1.1 Tipografía - Helvetica
- Figura 3.1.2 Código AngularJS (2 imágenes)
- Figura 3.1.2 Grid Bootstrap
- Figura 3.2 Diseño Phonegap
- Figura 3.2 Phonegap - Lenguajes - Plataformas
- Figura 3.2 Ionic - Phonegap - AngularJS
- Figura 3.2 Bower
- Figura 3.2 Yeoman
- Figura 3.2.1 Aspectos del diseño - Menú Hamburguesa
- Figura 3.2.2 Código Ionic
- Figura 3.2.2 Estructura proyecto móvil.
- Figura 4.1 Cabecera
- Figura 4.1 Pie
- Figura 4.1 Contenido Modal
- Figura 4.1 Buscador
- Figura 4.1 Carrusel
- Figura 4.1 Ferias

- Figura 4.2 Cabecera móvil
- Figura 4.2 Menú lateral móvil
- Figura 4.2 Feria móvil
- Figura 5.1 Estadísticas aplicación web
- Figura 5.2 Estadísticas aplicación móvil

1 Introducción

En este proyecto veremos el mundo del frontend en sus profundidades. Para ello tendremos como parte práctica una aplicación web y una aplicación móvil las cuales estarán realizadas por medio de una estructura HTML / CSS / JS. En primer lugar vamos a definir que es esto del frontend.

¿Qué es el frontend?

El front es la parte visual (parte del cliente) de la web.

Esto no quiere decir que el desarrollador frontend se dedique a la maquetación si no que también es el encargado de factores como la usabilidad, la accesibilidad y otras características que harán al usuario el poder consumir dicha aplicación web sin ningún tipo de dificultad o desacuerdo lógica.

Encontramos diferentes tipos de lenguajes y tecnologías como: HTML, CSS, JavaScript, jQuery, AngularJS, BackboneJS etc.

La web también se compone de otra parte llamada Backend, la cual se encarga de recibir los datos de otras fuentes como por ejemplo pueden ser bases de datos externas (parte del servidor).

Se compone de lenguajes como: PHP, Ruby on Rails, Django, Node.js, .NET etc.

La unión de estas dos partes forma una completa aplicación web las cuales las podemos ver desde páginas web destinadas a las compras como puede ser el caso de Amazon o páginas web simples sin ningún tipo de dato dinámico, es decir, una página web estática.



Figura 1. Frontend vs Backend

1.1 Planteamiento del problema

La idea de crear el negocio Quesoteca surge de la gran importancia que tiene el queso en la región manchega siendo uno de los entrantes/atractivos mas importantes de la zona.

Tras analizar el otro gran símbolo de Castilla La Mancha, el vino, y ver que goza de un gran grado de popularidad con múltiples accesorios para su conservación, muchos sitios hablando de el, sus tipos, calidades, elaboración, beneficios, etc me di cuenta que el mundo del queso también podría tener un sitio así y poder ser una idea de negocio factible y sostenible.

1.2 Objetivos

El objetivo principal de este proyecto es sacar adelante una aplicación web y una aplicación móvil lo más eficiente, modulable y escalable posible.

Siempre usando las tecnologías más actuales en el mundo del frontend y en el mundo del desarrollo de aplicaciones hibridas.

Sin olvidar el presentar todas las tecnologías usadas en el desarrollo actual del mundo del frontend desde frameworks para utilizar grids, preprocesadores, frameworks para dar estilos etc.

Como objetivo mas abstracto podemos hacer alusión a la propulsión de una nueva idea de negocio como es Quesoteca. No encontramos en el mercado nada similar pudiéndose convertir en un principio de una pequeña Startup.

1.3 Metodología

La elección de la metodología es un punto muy importante a la hora del desarrollo de un proyecto del tipo que sea tanto de software como de cualquier otro sector. Por lo tanto fue una decisión determinante y más al contar con tan poco tiempo de investigación y desarrollo.

Como dijo Ángel Sevilla (ponente en la charla de Metodología) ninguna metodología es mejor que otra, simplemente tenemos que elegir la que más se adapta a nuestras necesidades.

En este caso la necesidad de realizar un proyecto en muy poco tiempo hizo que me decantara por una metodología ágil.

Dentro de estas metodologías agiles hay una que es la que se ha ajustado a las necesidades actuales la cual es:

- SCRUM: Con constantes comunicaciones con el tutor del proyecto para definir el proyecto, pidiendo opinión y consejo sobre otro punto de vista.

Dentro de esta metodología hay definidos distintos Sprint donde se proponen objetivos de trabajo. Esto último ha sido imprescindible y muy motivacional a la hora de terminar el desarrollo, ya que la fijación de diferentes metas temporales hace que se tome como una especie de reto.

Además de todo esto, es una metodología fácil de aprender y modificar, permitiéndome diferentes cambios de ritmo según se van terminando las diferentes tareas.



Figura 1.3. - SCRUM

2 Descripción del problema

Los principales problemas que se pueden llegar a encontrar es la dificultad de las nuevas tecnologías con las que se ha pensado desarrollar este proyecto.

Al ser tecnologías no vistas en el curso, llevar poco tiempo en el mercado y no tener demasiado tiempo para la investigación y desarrollo de las aplicaciones hace complejo la fase de documentación y extracción de información de otros ejemplos de internet, ya que la información es escasa y siempre proviene de las APIS de dichas tecnologías.

Sin olvidar también de carecer de un modelo de negocio similar al del entorno software que se va a desarrollar.

No hay un claro ejemplo de un servicio como este por lo que el proceso de búsqueda de información para como plantearlo se hace mas complicado, teniendo que innovar, diseñar y crear todo desde 0.

3 El proyecto software

El proyecto software se divide en dos partes. La primera es en una aplicación web y segundo una aplicación móvil multiplataforma. A continuación se detallaran estas dos.

3.1 Aplicación Web.

La primera parte del proyecto consta de una aplicación web destinada exclusivamente al visionado por medio de una plataforma de escritorio, es decir, la pagina esta hecha exclusivamente para ser consumida por el navegador de un ordenador.

Esta pagina web tiene diferentes secciones que la componen:

- Inicio: Pequeña introducción a la pagina web.
- Historia: Cuenta la historia del producto, como surge el nombre y su historia en nuestra región.
- Tipos: Enumera los diferentes tipos de quesos que podemos encontrar en el mercado según el origen de la leche y la textura de la pasta.
- Beneficios: Aporta datos nutritivos sobre el queso, por que debemos comerlo y distintos alimentos con los que podemos acompañarlo.
- Ferias: Las diferentes ferias que se organizan en el ámbito nacional destinadas a la promoción y el autoaprendizaje del mundo del queso.

En ella encontramos diferentes tecnologías:

-HTML5 – CSS3 – Javascript.

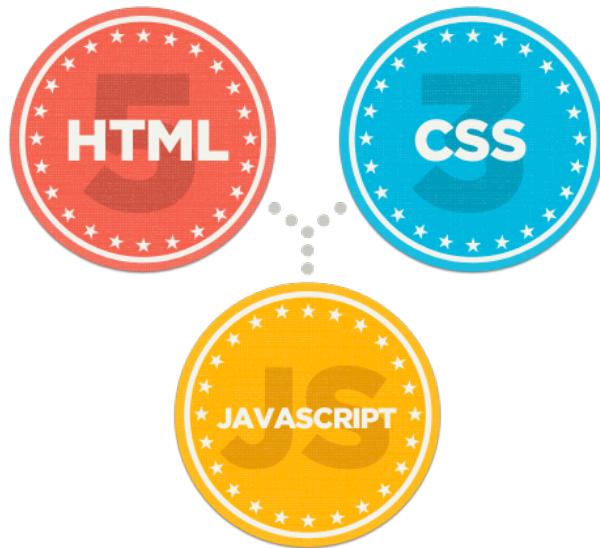


Figura 3.1. HTML / CSS / JS

Este es el full-stack de cualquier desarrollo web utilizando HTML5 como el lenguaje básico para la formación de una pagina web.

Componemos nuestra web en base a la inclusión de etiquetas formando así distintas capas con distintos elementos ya sean títulos, enlaces, imágenes etc.

La renovación de HTML 4 a la versión 5 ha supuesto unos cambios abismales a la hora de la explotación de la web ya que para los usuarios es mucho mas fácil el incluir contenido multimedia, estructurar la web y la inclusión de otro tipo de elementos como pueden ser elementos 3D.

Con CSS3 aplicamos un estilo a nuestra web, es decir, unos colores, unos tamaños de letra, separaciones, tipografías etc etc. Nos permite la incorporación de nuevas funciones como la inclusión de propiedades relativas a los bordes, fondos, colores, texto, interfaz, comodidad a la hora del usor de selectores, degradados, media queries etc etc.

Javascript comúnmente conocido como JS es la forma que tiene la aplicación web de interactuar con el usuario. Desde mensajes de alerta, validación de formularios, sincronización de datos con bases de datos externas, manejar y manipular el DOM, contenido interactivo como juegos, animación de los diferentes elementos de la página etc. Con javascript tenemos un sinfín de posibilidades. El límite lo ponemos nosotros.

-SAAS



Figura 3.1 - SASS

Usaremos SASS como preprocesador CSS.

¿Para qué? Con esta tecnología ahorraremos muchas líneas de código de estilos, nos compilará nuestro código a css avisándonos así de si hubiese algún tipo de error, nos amplia funcionalidades como el manejo de distintos tipos de datos como por ejemplo, números, cadenas, colores (código hexadecimal o nombre) y booleanos.

Nos permite también la creación de distintas funciones (mixins) los cuales podremos utilizar por ejemplo para calcular la equivalencia de 1em a pixel, importar imágenes con una serie de patrones estilísticos etc. La creación de variables para almacenar datos y ser utilizados en nuestro código, el uso de directivas de control (if/else), bucles y distintas librerías ya creadas con la inclusión de paletas predefinidas, mixins ya creados etc. Sin olvidar también la posibilidad de modularizar nuestro código ya que podremos importar diferentes hojas de estilo haciendo así nuestro desarrollo mas optimo y limpio reduciendo el numero de líneas por archivo.

Con SASS la capacidad de dar estilo a una web es muchísimo mas amplia y mucho mas potente ya que optimiza al máximo nuestro código CSS restante.

Ejemplo básico de código y su conversión equivalente.

SCSS	CSS
<pre> 1 section { 2 height: 100px; 3 width: 100px; 4 5 .class-one { 6 height: 50px; 7 width: 50px; 8 9 .button { 10 color: #074e68; 11 } 12} 13 </pre>	<pre> 1 section { 2 height: 100px; 3 width: 100px; 4 } 5 6 section .class-one { 7 height: 50px; 8 width: 50px; 9 } 10 11 section .class-one .button { 12 color: #074e68; 13 } </pre>

Figura 3.1 Comparación de código SASS - CSS

-Bootstrap



Figura 3.1 - Bootstrap

Bootstrap es un framework creado por Twitter que nos amplia y ayuda en distintas funcionalidades tanto de CSS como de JS.

El principal fuerte de bootstrap es su grid, haciendo al desarrollador muy cómoda la tarea de maquetar una web full responsive diciéndole así el numero de columnas que quiere que ocupe dicho elemento.

También usaremos bootstrap como base de desarrollo aplicándonos por defecto unos estilos limpios, colores minimalistas y distintos elementos ya creados teniendo solo que importarlos y modificarlos mínimamente.

Encontramos diferentes elementos ya creados ya pueden ser carruseles de imágenes, barras de navegaciones, pies, elementos laterales, formatos de imágenes, formularios y su validación etc.

Con este framework hacemos que nuestra web sea valida para todos los navegadores por lo que el desarrollo invertido para que la web sea equivalente en todos los navegadores es mínimo.

-AngularJS



Figura 3.1- AngularJS

Framework Javascript desarrollado por Google de código abierto que abre permite crear aplicaciones MVC (Modelo Vista Controlador) y SPA (Single Page Application).

¿Por qué usar AngularJS?.

Con AngularJS creamos una aplicación web extremadamente rápida ya que solo va a realizar una petición al servidor (SPA) haciendo la experiencia de usuario realmente gratificante sobre todo si es un portal con distintas partes o tabs.

Ayuda muchísimo a la modularidad del portal ya que podremos importar diferentes partes de la pagina en nuestro código haciendo así los archivos .html con muy pocas líneas.

Sin olvidar también de que la manipulación del DOM con AngularJS es muy eficiente, haciendo prácticamente invisible cualquier cambio en el.

Al seguir el modelo MVC permite disociar el lado cliente del lado servidor de la aplicación haciendo así una mejora para su manutención y un desarrollo en paralelo.

En [1] <http://trends.builtwith.com/javascript/Angular-JS> podremos ver la evolución de sitios web que están apostando fuertemente por el uso de AngularJS como motor principal.

-jQuery



Figura 3.1 - jQuery

Es una biblioteca Javascript que permite la forma de interactuar con los Documentos HTML, manipular el árbol DOM, manejar distintos eventos, animaciones, manipulación de estilos etc.

El código es totalmente libre por lo que se abre la posibilidad de a los desarrolladores de seguir optimizándolo y mejorándolo.

En resumen, usamos jQuery para una mayor optimización del manejo del DOM, tiempo de escritura de código Javascript y distintos efectos ya implementados.

3.1.1 Aspectos del diseño

Al no partir de ningún tipo de plantilla ni base ya creada, el proceso del diseño ha sido con diferencia de los mas costosos por la poca experiencia en el sector.

La intención con el diseño es extrapolarnos a una web actual, con diseño minimalista y sin excesos de componentes que hagan una experiencia de usuario amable y sin ningún tipo de complicación a la hora de navegar por ella.

Los colores utilizados han sido generados a partir de una paleta de colores combinándolos entre sus colores análogos, complementarios y tonales. Se ha procurado una familiarización lógica para la temática de la página con los colores y tonos seleccionados siendo así el amarillo claro a marrón los elegidos.

A continuación la paleta que se ha seguido.



3.1.1 - Paleta de colores

Como la intención es un aspecto minimalista se han tenido en cuenta distintas tipografías utilizadas de distintas empresas líderes en el sector de la tecnología e innovación como pueden ser Apple, Google, Twitter, Facebook. Tipografías limpias y claras que nos transportan a un diseño conceptual muy actual. En este caso la fuente primordial elegida ha sido Helvetica Neue utilizada por Apple en sus diseños de interfaz gráfica para sus dispositivos.

3.1.1 - Tipografía - Helvetica

the quick brown fox
jumps over the lazy dog.

THE QUICK BROWN FOX
JUMPS OVER THE LAZY DOG.

3.1.2 Aspectos técnicos

Referente al apartado técnico, se han usado las tecnologías mencionadas anteriormente para lo siguiente:

- **HTML5:** Lenguaje con el que crearemos la estructura de una página web. Creando sus diferentes elementos ya sea organizativos como secciones, capas como multimedia imágenes, videos, mapas etc.
- **SASS:** Crear un estilo para la aplicación web. Gracias a SASS tenemos un código limpio y modulable teniendo así distintos archivos los cuales se compilan y crean un solo archivo CSS eficaz y fiable.
- **Bootstrap:** Se ha utilizado mayormente para su grid ahorrando así tiempo de elección de medidas para los componentes. También ha ayudado mucho a la hora de establecer un diseño preliminar aunque posteriormente se ha ido ajustando a la temática de la web.
- **jQuery:** Librería de JavaScript usada para la importación de elementos multimedia como el carrusel de imágenes. Es imprescindible jQuery en nuestro proyecto ya que sin este componente otros dejarían de funcionar como por ejemplo Bootstrap.
- **AngularJS:** El motor de la aplicación web. Se ha usado para la limpieza y modularidad de toda la gestión de HTML. Se han usado diferentes directivas como:
 - **NG-App:** Damos nombre a la aplicación Angular.
 - **NG-Repeat:** Para la creación de elementos repetidos cambiando el contenido. Se encuentra en la parte de tipos de queso.
 - **NG-Include:** Para la incorporación de elementos fijos a la web como pueden ser la cabecera y el pie.
 - **NG-View:** Para la gestión del contenido dinámico. En esta parte se mostrarán los diferentes parciales .html de nuestra aplicación.
 - **NG-Route:** Gestiona el enrutamiento de las diferentes páginas web. Con esto nuestro portal web aumenta en seguridad ocultando los archivos .html que está cargando y facilitándonos así la gestión de dichas partes.
 - **NG-Controller:** Controlador para el elemento que asignamos. Tenemos dos controladores, uno para cada categorización de quesos. Así mismo hace la parte de lógica ya que por el controlador ofrecemos los datos a la aplicación.

También usamos data binding para la asignación de los datos en los elementos que queremos. Lo podemos encontrar donde usemos la directiva **NG-Repeat**.

Conceptos relacionados con AngularJS:

- Directiva:** El concepto de directiva consiste en una funcionalidad que aporta dicho elemento. Es decir, al igual que en java u otros lenguajes, ya hay funciones creadas para cualquier operación (matemática, manejo de cadenas, leer ficheros etc.) en angular coge el nombre de directiva.
- Data binding:** Es el término utilizado para refrescar la vista automáticamente cada vez que el modelo cambie y viceversa. Esto es posible debido a que se ha eliminado la manipulación del DOM.

Ejemplos de código AngularJS en el proyecto.

```

1. <div class="row" ng-controller="tipos">
2.   <ul class="aux-tipos" ng-repeat="x in tipos | filter: {titulo:busqueda}">
3.     <li class="tipo-leche clearfix">
4.       <div class="col-lg-4">
5.         <div class="img-tipo">
6.           
7.           <p class="espefi-tipo">{{x.espef}}</p>
8.         </div>
9.       </div>
10.      <div class="col-lg-8">
11.        <div class="texto-tipo">
12.          <h3>{{x.titulo}}</h3>
13.          <p>{{x.texto}}</p>
14.        </div>
15.      </div>
16.      <div class="raya-tipo"></div>
17.    </li>
18.
19.  </ul>
20. </div>

```

Figura 3.1.2 - Código AngularJS (2 imágenes)

```

1. <!DOCTYPE html>
2. <html lang="es" ng-app="quesoteca">
3. <head>
4.   <meta charset="UTF-8">
5.   <title>Quesoteca</title>
6.   <link rel="stylesheet" type="text/css" href="styles/main.css">
7.   <link rel="stylesheet" type="text/css" href="lib/css/bootstrap.css">
8.   <link rel="stylesheet" type="text/css" href="lib/css/hover-min.css">
9.   <link rel="stylesheet" type="text/css" href="lib/css/jquery.fullPage.css">
10.  <link rel="stylesheet" type="text/css" href="lib/css/font-awesome.min.css">
11.  <link rel="stylesheet" type="text/css" href="lib/css/slider.css">
12.
13.  <script type="text/javascript" src="lib/js/angular.js"></script>
14.  <script type="text/javascript" src="lib/js/angular-route.js"></script>
15.  <script type="text/javascript" src="js/route.js"></script>
16. </head>
17.
18. <body>
19.
20.   <div ng-include src=" 'html/cabecera.html' "></div>
21.   <div ng-view></div>
22.   <div ng-include src=" 'html/pie.html' "></div>
23.
24.   <script type="text/javascript" src="lib/js/jquery.js"></script>
25.   <script type="text/javascript" src="lib/js/bootstrap.min.js"></script>
26.   <script type="text/javascript" src="js/controller.js"></script>
27.   <script type="text/javascript" src="js/main.js"></script>
28.   <script type="text/javascript" src="lib/js/script.js"></script>
29.   <script type="text/javascript" src="lib/js/wowslider.js"></script>
30.

```

Conceptos relacionados con Bootstrap:

- **Grid:** Una definición de grid podría ser una división de un área de trabajo en cuadriculas. ¿Cómo aplicamos esto a un diseño de una pagina web?. Pues bien, comúnmente un diseño web se realiza en columnas teniendo dos variaciones 10 y 12 columnas siendo esta ultima la mas utilizada en el desarrollo web. En la actualidad existen muchísimos tipos de grid de diferentes frameworks teniendo cada uno sus medidas y sus características.
El uso de un grid es imprescindible para un diseño full responsive que aunque en este caso no es necesario ya que existe una aplicación móvil, su implantación no seria muy costosa ya que estamos usando un diseño basado en columnas por lo que pequeñas empresas se pueden olvidar sin ningún pudor de un desarrollo móvil.

Como vemos en la siguiente imagen hay una división por columnas. Cualquier división tiene que sumar 12 por lo que si queremos un elemento al lado de otro del mismo tamaño deberemos dividirlo en dos col-6, o si queremos un elemento que ocupe todo el ancho de la pantalla usamos un col-12 y así sucesivamente. De esta manera nos permite organizar nuestros elementos de una forma fácil y rápida.

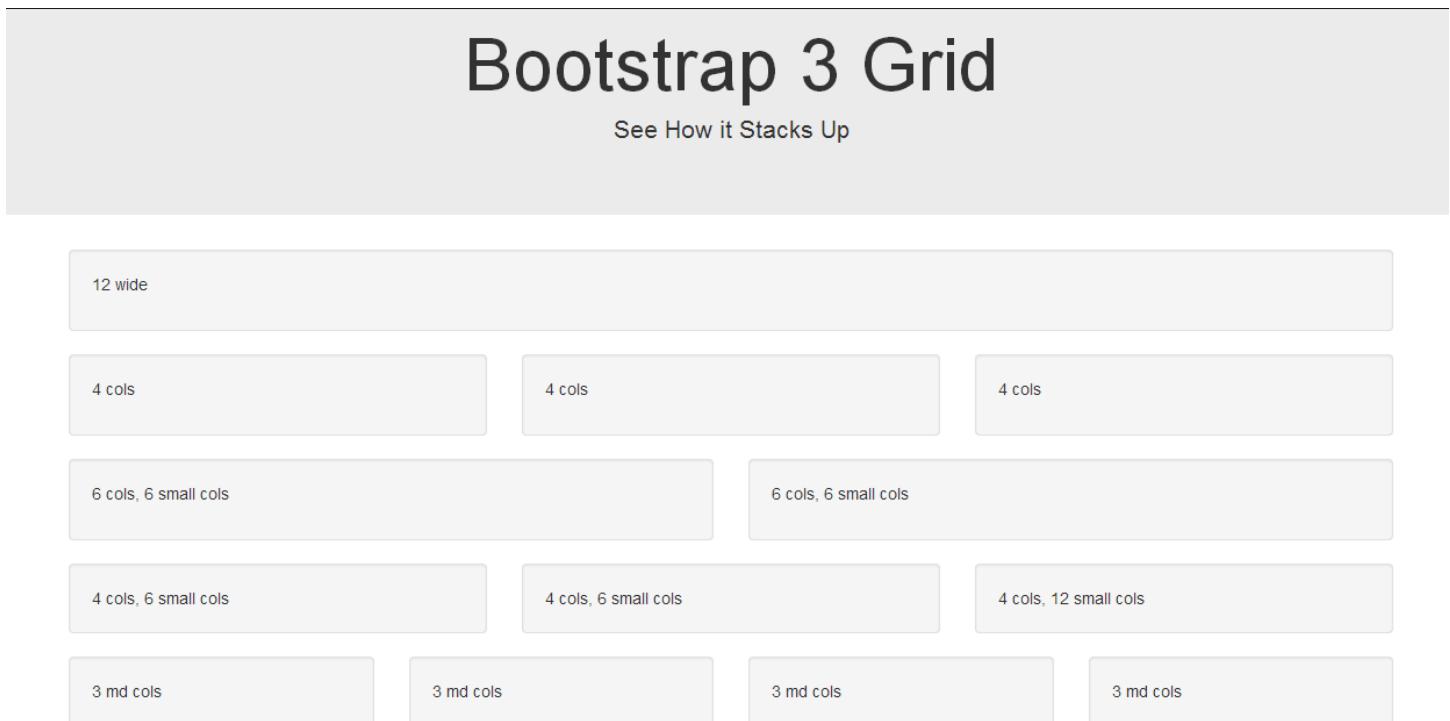


Figura 3.1.2 - Grid Bootstrap

El proyecto se estructura de la siguiente manera.

- **Html:** Carpeta donde se guardan los distintos archivos .html. Desde los fijos hasta los dinámicos.
- **Img:** Se albergan las diferentes imágenes que se han usado en el desarrollo.
- **Js:** Contiene los diferentes archivos javascript. Tanto de javascript como de AngularJS.
- **Lib:** Encontramos las diferentes librerías que hacen falta para que el proyecto funcione correctamente. A su vez, esta dividido en mas carpetas dependiendo de sus tecnologías. Estas divisiones son de css, js, y fonts (fuentes).
- **Styles:** Encontramos los archivos .scss (SASS) y su archivo .css compilado que es el que se importa al index.html.
- **Raíz:** Encontramos el archivo index.html en el cual están las importaciones de las librerías, los includes de las partes fijas de la web (cabecera y pie) además de la parte dinámica y por ultimo los diferentes imports de archivos .js que añaden funcionalidad a nuestro sitio web.

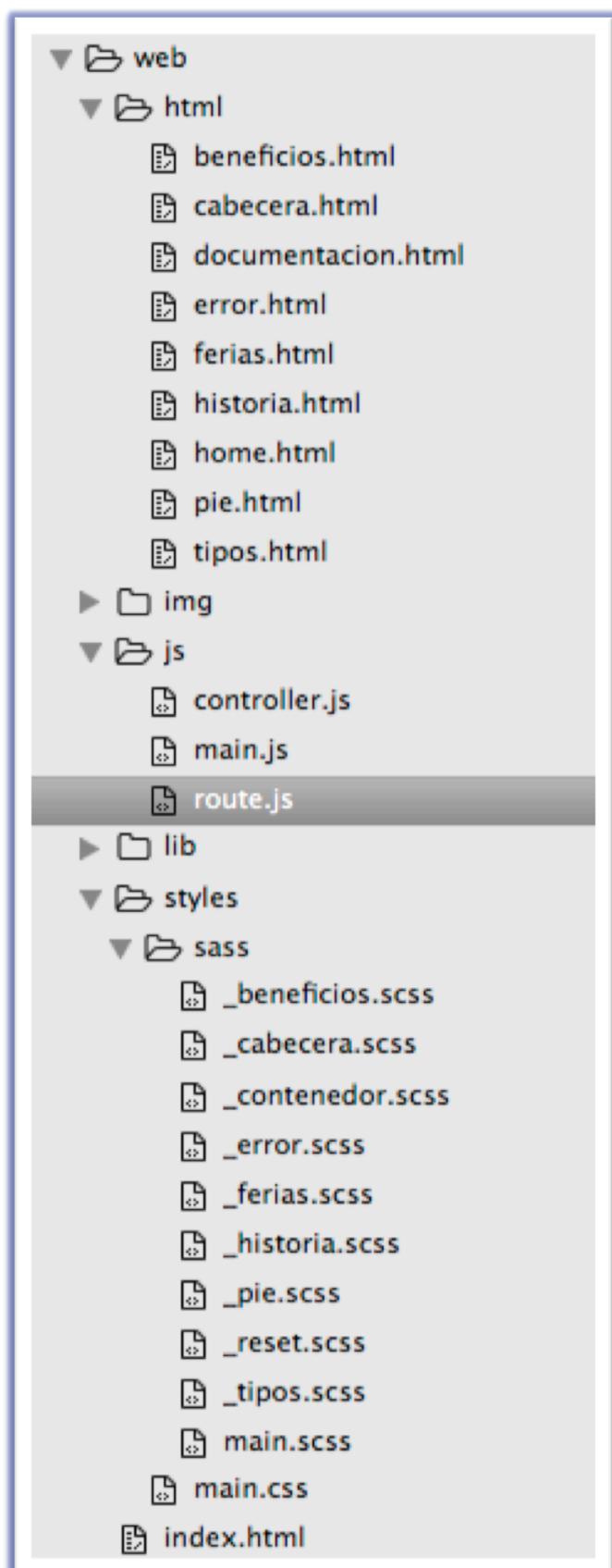


Figura 3.1.2 - Estructura proyecto web

3.2 Aplicación móvil

La aplicación móvil será el perfecto ejemplo de cómo las tecnologías de desarrollo web han dado un avance impresionante y son capaces de ejecutar una aplicación con un lenguaje de marcas (HTML), un estilo (CSS) y acciones con las que interactúa el usuario (JavaScript) a un rendimiento de una aplicación de desarrollo nativo como puede ser iOS con el lenguaje Objective - C y Android con Java.

De la mano de Phonegap podemos usar los lenguajes de la web para el desarrollo de aplicaciones multiplataforma haciendo de un solo desarrollo una aplicación para plataformas como: iOS, Android, Windows Phone, WebOS, BlackBerry OS, Symbian etc.

Así mismo, seguiremos el patrón anterior explicando las tecnologías utilizadas, el aspecto de diseño y por último el aspecto técnico.

Al ser la aplicación móvil una extensión de la aplicación web muchos de los conceptos aplicados ya se han explicado anteriormente. En esta sección solo se mencionarán rápidamente los ya utilizados y explicados en profundidad los nuevos.

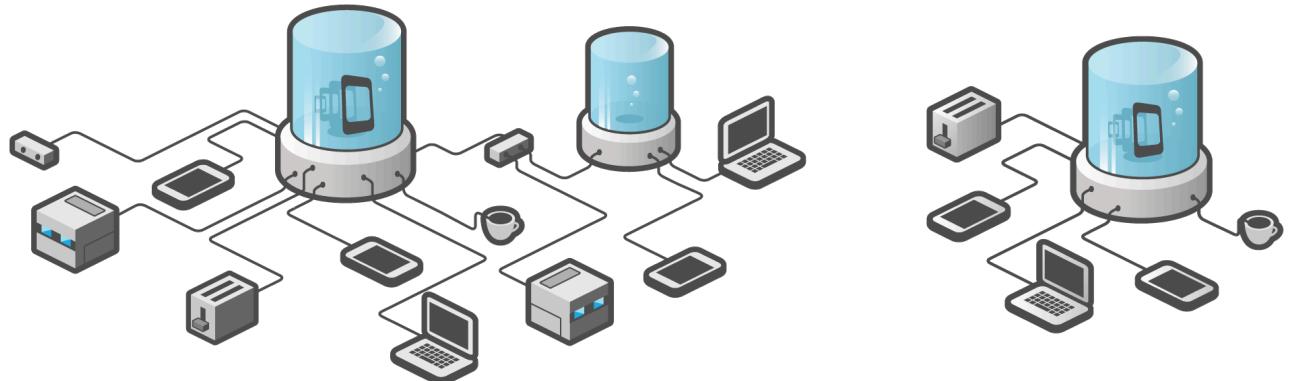


Figura 3.2 Diseño Phonegap

La aplicación móvil de quesoteca utiliza las siguientes tecnologías:

-Phonegap (Cordova)



Figura 3.2 Phonegap - Lenguajes - Plataformas

Phonegap es un framework para el desarrollo de aplicaciones móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3. La aplicaciones resultantes son de tipo hibridas, es decir, no son aplicaciones realmente nativas pero tampoco son aplicaciones web que trabajan con el navegador web del dispositivo.

El punto fuerte de Phonegap es que de un solo desarrollo obtenemos una aplicación para distintas plataformas, además de contar con una extensa comunidad de desarrolladores que suben continuamente distintos plugins que hacen de este framework prácticamente una aplicación nativa ya que usa accesorios como la cámara, agenda de contactos, sistemas de geocalización, lector de huellas etc.

Phonegap también es conocido como Cordova que prácticamente es lo mismo cuya única diferencia es que Phonegap tiene acceso a servicios de compilación en la nube proporcionados por Adobe Creative Cloud. Tanto uno como otro, son frameworks de código abierto.

-Ionic



Figura 3.2 - Ionic - Phonegap - AngularJS

Ionic es un framework de Apache Cordova que a su vez tiene por debajo AngularJS.

Con este framework tenemos las ventajas de poder realizar distintas aplicaciones para distintas plataformas con una sola compilación, la velocidad y eficiencia de AngularJS cuyo rendimiento se debe a la mínima manipulación del DOM, con cero jQuery y con aceleraciones de transiciones por hardware.

Ionic no solo se ve bien, sino que su arquitectura central es robusta y seria para el desarrollo de aplicaciones tanto personales como comerciales. La integración con AngularJS es totalmente perfecta.

Además, Ionic se inspira en las SDK de desarrollo móviles más populares, por lo que es fácil de entender para cualquier persona que ha construido una aplicación nativa para iOS o Android.

Debajo de Ionic tenemos a NodeJs haciendo donos de servidor, es decir, con unos simples comandos podremos desde crear, construir, probar a compilar nuestras aplicaciones en cualquier plataforma.

Por ultimo y casi lo mas importante, el diseño. El diseño es bonito, sencillo y funcional siendo diseñado para poder trabajar con todos los dispositivos móviles actuales. Dando así una impresión de aplicación nativa.

-SASS

Explicado anteriormente en la parte de la aplicación web.

-Gulp.js



Figura 3.2 Gulp.js

Gulp es un sistema de construcción que permite automatizar tareas comunes de desarrollo, tales como la minificación de código Javascript, recarga del navegador, compresión de imágenes, validación de sintaxis de código y un sin fin de tareas mas.

Gulp esta construido con Javascript, funciona sobre Node.js y es totalmente Open Source.

Tenemos la posibilidad de crear diferentes tareas y el solo las ejecutará. De las tareas mas comunes que se suelen realizar es la de compilación de SASS a CSS. Ejecutar una tarea es tan fácil como poner en la terminal (estando situados en el sitio oportuno) `gulp nombreTarea`

-Bower



Figura 3.2 Bower

Bower es un manejador de paquetes para la web, es decir, cualquier librería de código abierto que podamos necesitar en nuestros desarrollos frontend la podremos descargar y gestionar con bower.

Su uso va muy ligado con automatizadores como Gulp.js o GruntJs siendo tan fácil descargar nuestras librerías como especificarlas en el archivo de configuración config.xml que se crea al instalar bower y con nuestro automatizador indicarle donde guarda las librerías.

-Yeoman



Figura 3.2 Yeoman

Usaremos Yeoman para crear nuestro proyecto de manera rápida, sencilla y sin ningún tipo de complicación.

Con Yeoman tenemos la estructura de nuestra aplicación (según las tecnologías que le indiquemos que instale) a golpe de teclado.

Existen unos 20 generadores oficiales mantenidos por el equipo de Yeoman siendo de los mas utilizados los de las aplicaciones basadas en Angular. Con esto nos ahorraremos muchísimo tiempo a la hora de empezar nuestros proyectos sin necesidad de tener que estar haciendo a mano las importaciones de librerías, de estilos, creando carpetas etc etc.

Todo esto se generará a partir de comandos que nosotros vayamos introduciendo en nuestra terminal. El comando para invocar a Yeoman es yo y a continuación el tipo de aplicación o tecnología que queremos que nos genere.

3.2.1 Aspectos del diseño

En lo referente a aspectos del diseño no varia apenas sobre el diseño de la aplicación web ya que hay que intentar seguir el mismo formato, colores y tipografía de la misma marca en distintas plataformas.

Se ha procurado seguir la misma línea que el diseño web, elegante, minimalista, claro y funcional. Usando un color mas oscuro para la cabecera y colores blancos con diferencias tonales para el fondo.

En el tema de usabilidad se ha incorporado un menú hamburguesa a la izquierda del menú superior para el cambio de sección.

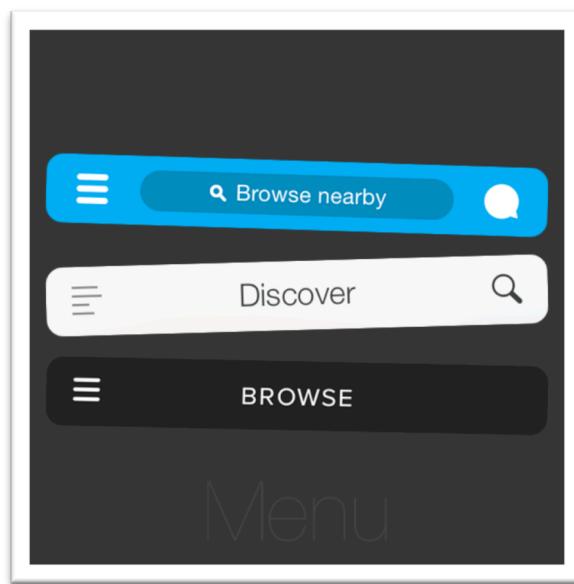


Figura 3.2.1 Aspectos del diseño - Menú Hamburguesa

3.2.2 Aspectos técnicos

En base a los aspectos técnicos de la aplicación móvil no difieren mucho de la aplicación web ya que Ionic es el encargado de montar todo el entramado del proyecto.

La premisa de una aplicación web y una aplicación híbrida consiste en la reutilización de componentes cumpliéndolo en este caso a la perfección. Volveremos a utilizar los archivos AngularJS con la información sobre los quesos, los elementos del texto, estilos predefinidos etc.

El cambio funcional es prácticamente nulo siendo lo único que debemos adaptar son los tamaños de los diferentes componentes ya que no podemos utilizar la misma organización de los componentes debido al cambio de las dimensiones de la pantalla.

Un archivo .html Ionic tiene esta estructura.

- ion-view title: El nombre del título del apartado.
- ion-content: El contenido que interpretará Ionic.

```
1. <ion-view title="Quesoteca">
2.   <ion-content class="has-header">
3.     <div class="list card">
4.       <div class="item item-thumbnail-left">
5.         
10.        <p>{{tipos.espef}}</p>
11.      </div>
12.      <div class="item tabs tabs-secondary tabs-icon-left">
13.        <a class="tab-item">
14.          <i class="icon ion-thumbsup"></i>
15.          Like
16.        </a>
17.        <a class="tab-item">
18.          <i class="icon ion-chatbox"></i>
19.          Comment
20.        </a>
21.        <a class="tab-item" ng-click="share()">
22.          <i class="icon ion-share"></i>
23.          Share
24.        </a>
25.      </div>
26.    </div>
27.  </ion-content>
28. </ion-view>
```

Figura 3.2.2 Código Ionic.

A continuación se mostrará la estructura de un proyecto hecho con Ionic.

- Hooks: Diferentes scripts de Apache Cordova.
- Node_modules: Esta es la carpeta que se genera al instalar diferentes dependencias de las tecnologías comentadas anteriormente.
- Platforms: Las plataformas para las que va a estar disponible esta aplicación. Se puede cambiar el estilo de la aplicación simplemente cambiando el .css de dicha plataforma.
- Plugins: Los plugins de Cordova que necesita para el correcto funcionamiento de la aplicación.
- Scss: Los archivos sass de los que esta creado Ionic.
- www: Estructura donde encontramos los archivos de los que esta compuesta la aplicación ya sea .html, css, js etc.
 - Css: Donde están ubicados los estilos de la aplicación.
 - Img: Aquí se ubican las imágenes de la aplicación.
 - Js: Archivos javascript. Tanto js como de AngularJS.
 - Lib: las librerías utilizadas para el desarrollo.
 - Templates: Todas las páginas web que se cargan en nuestra aplicación.
- Bower.json: Donde tiene bower indicado las dependencias que hacen falta.
- Config.xml: Archivo de configuración de Cordova. Se indica la descripción del proyecto, el nombre, el autor etc.

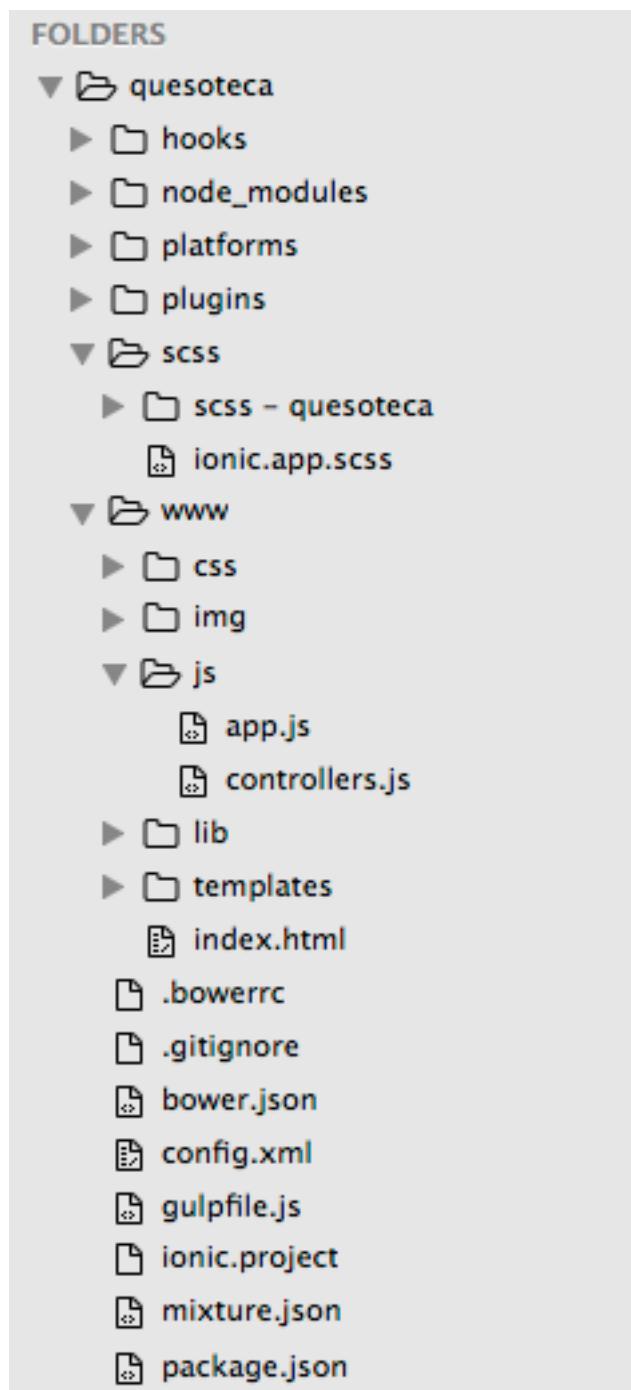


Figura 3.2.2 Estructura proyecto móvil

- Gulpfile.js: Archivo de configuración de gulp. Aquí encontramos las diferentes tareas existentes. También podemos añadir las nuestras.
- Package.json: Toda la información de la aplicación tanto de dependencias como de la propia aplicación (versión, descripción, plataformas etc.)

4 Solución

4.1 Solución – Aplicación web

A continuación mostraremos el resultado de la aplicación web. Se mostraran parte por parte ya que si hiciésemos muestra de la web con una imagen de toda tendríamos un tamaño de imagen muy pequeño con lo que no se podría apreciar correctamente.

Se detallaran los diferentes elementos de la aplicación web, como son la cabecera, el contenedor, el pie, elementos de búsqueda, mapas etc.

Cabecera



Figura 4.1 Cabecera

La cabecera será uno de los medios por los que el usuario podrá interactuar con la página. Cada elemento del menú tiene un efecto hover que mejorar la experiencia de usuario. El color empleado hace alusión a la corteza del queso.

Pie



Figura 4.1 Pie

El pie será otro modo de interacción con el usuario. Tenemos distintos elementos en la parte inferior derecha que corresponden con las diferentes secciones que la pagina web tiene creadas.

Contenido modal

Quesoteca
Viviendo el queso

Bienvenido

En Quesoteca descubriremos el amplio mundo de uno de los alimentos mas consumidos a nivel mundial, el queso. En este pequeño rincón veremos diferentes temas relacionados con el queso, como pueden ser, su historia, sus principales tipos de queso, beneficios que nos aporta consumir queso y las diferentes ferias relacionadas con la promoción y degustación de nuestro producto mas admirado. El queso.



• —————— •



Historia

Descubre la historia del queso con Quesoteca. En ella encontraremos como fué el nacimiento de este producto con sus diferentes descubrimientos hasta la historia del queso en esta nuestra región. Castilla la Mancha.



Tipos de Queso

El queso es uno de los alimentos con mas variaciones que existen, desde el tipo de leche, el tipo de pasta, la textura, según el origen etc. Haciendo uno de los alimentos mas difíciles de clasificar. En Quesoteca encontramos dos de las clasificaciones mas importantes.



Beneficios del queso

A parte de ser delicioso, el queso es muy beneficios para la salud, tan solo una porción de queso diaria nos puede ayudar al aporte lacteo que necesitamos para estar sano. En esta sección encontrarás diferentes datos apoyando todos los beneficios del queso. A su vez, descubrirás otros alimentos con los que podemos combinarlo. ¡Convertete en todo un gourmet!



Ferias

El queso es un alimento todavía poco conocido en el panorama Español. Actualmente, existen distintas ferias que ayudan a que todos los usuarios conozcan este manjar. Veremos las principales ferias celebradas en el panorama nacional y su localización.

Figura 4.1 Contenido modal

El contenido es la parte que proveerá de información al usuario. Siempre concordará el elemento que haya seleccionado el usuario con la información recibida.

La combinación de colores siempre será modulaciones tonales del amarillo, blanco y marrón.

Buscador

Se ha integrado un buscador en la página Tipos de Queso para que al usuario le sea mucho mas cómodo encontrar uno de los tipos de las diferentes categorías que existen. El origen de la leche y el tipo de la pasta.

El usuario introducirá el texto y la página se encargará de mostrar los resultados que concuerden con lo introducido.

Clasificación según el origen de la leche.

Buscador



Figura 4.1 Buscador

Carrusel



Figura 4.1 Carrusel

En la parte de beneficios del queso encontramos un carrusel el cual proveerá al usuario de información gráfica referida con los diferentes elementos con los que se puede combinar el queso.

En la lista que encontramos debajo del carrusel nos da una información mas extensa.

Ferias

En la sección ferias encontramos un modo de interacción con la pagina diferente al resto. Al hacer slide hacia abajo se mostrará un efecto de retardo el cual no pasará desapercibido para el usuario creando una experiencia mucho mas llamativa y enriquecedora.

En la primera pantalla encontramos un icono indicandonos que hay que ir hacia abajo para ver mas contenido.

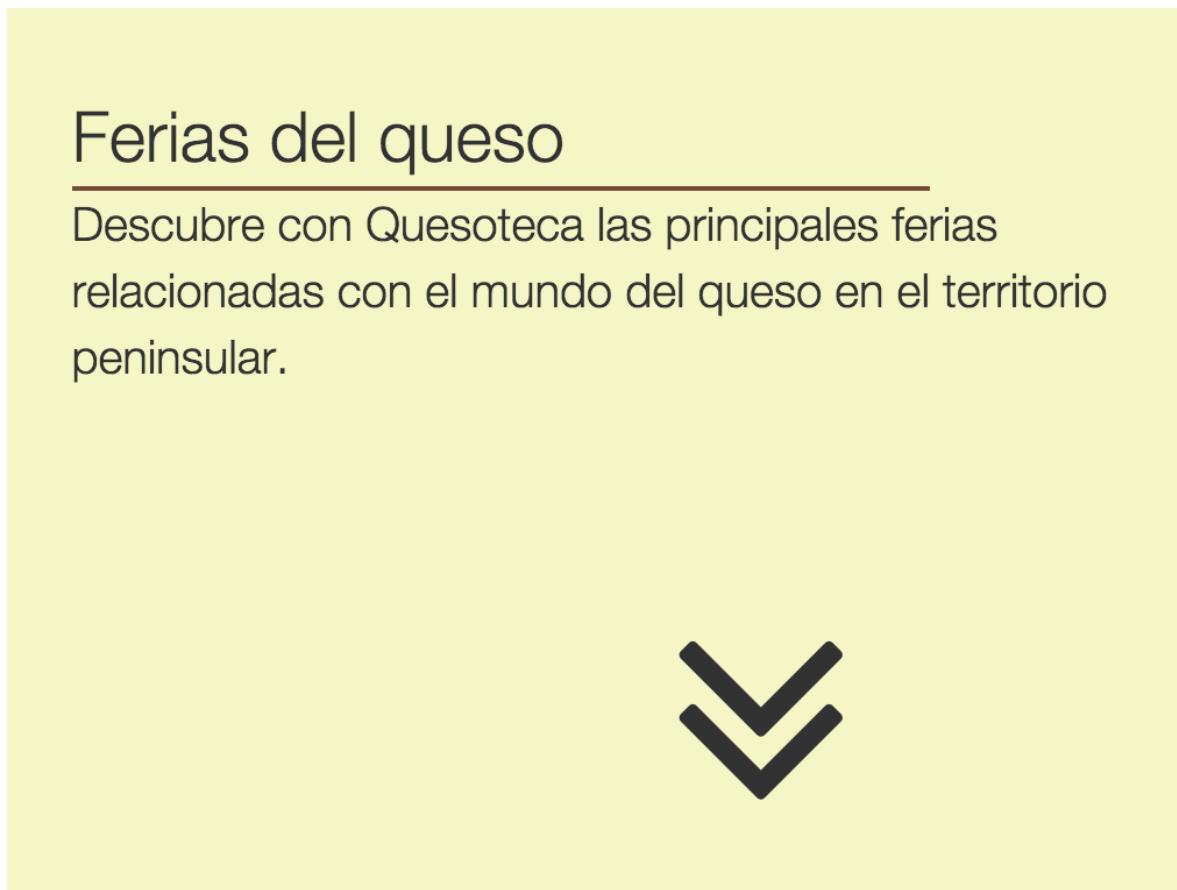


Figura 4.1 Ferias

En ferias vemos otro tipo de mostrar la información, ocupando toda la pantalla la información dando un aspecto muy minimalista. Estas especie de tarjetas nos proveerá de la ubicación de la feria, el nombre de la feria, una breve descripción y un mapa con el que podremos interactuar incluso indicarnos como llegar hacia la feria seleccionada.

Feria de Quesos Artesanos

Madrid

I Feria de Quesos Artesanos de España en Madrid, en la sede de la Cámara Agraria de Casa de Campo, gracias a la iniciativa de la Quesería La Cabeza (Fresnedillas de la Oliva) y la colaboración de la Cámara Agraria de la Comunidad de Madrid. Con ella han conseguido que los días 28, 29 y 30 de septiembre se reúnan casi una treintena de queserías artesanas para acercar sus productos a los consumidores.

El acceso a la Feria de Quesos Artesanos de España está abierto a todos y es gratuito, lo que nos va a permitir conocer muchos de los mejores quesos de nuestro país, las especialidades que cada región produce, desde los quesos de Asturias a los de Menorca, pasando por los quesos extremeños, los cántabros o los zamoranos, aunque hay una gran ausencia de representantes de otras comunidades.



Figura 4.1 Ferias - Tarjetas

4.2 Solución – Aplicación móvil

En el apartado de la solución ofrecida para la aplicación móvil si que podremos realizar imágenes a tamaño completo aunque también se especificaran las partes de dicha aplicación.

Cabecera

En el menú superior encontramos un menú hamburguesa situado a la izquierda de dicha barra el cual nos proporcionará las opciones para poder cambiar de apartados.

En el centro encontramos el nombre de la aplicación a ejemplo de logo.



Figura 4.2 Cabecera Móvil

Como en la aplicación web, el color de la cabecera representa la corteza del queso.

Menú lateral

Al pulsar sobre el ícono del menú hamburguesa se despliega un menú lateral. También podemos acceder a él haciendo scroll hacia la derecha emulando un gesto de aparición.

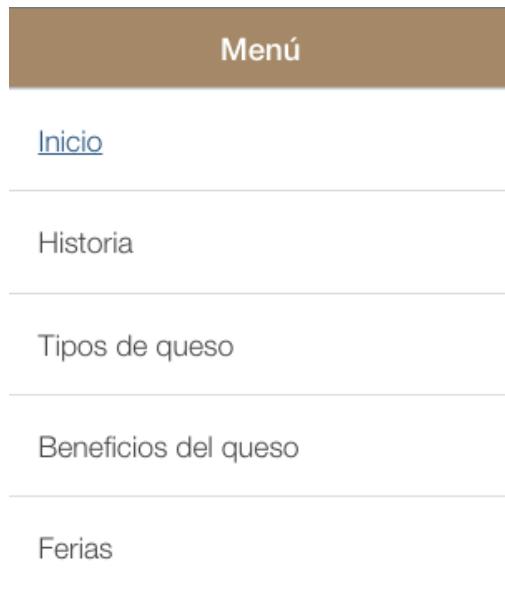


Figura 4.2 Menú lateral móvil

Ferias

También en la aplicación móvil encontramos un cambio de la interacción del usuario. En este caso deslizando hacia la derecha.

Siguiendo el mismo formato que en la web, la información ofrecida consta de un título, una descripción de la feria y un mapa con el que podremos interactuar.

Los iconos situados en la parte inferior de la pantalla nos indican en qué zona de la sección nos encontramos.

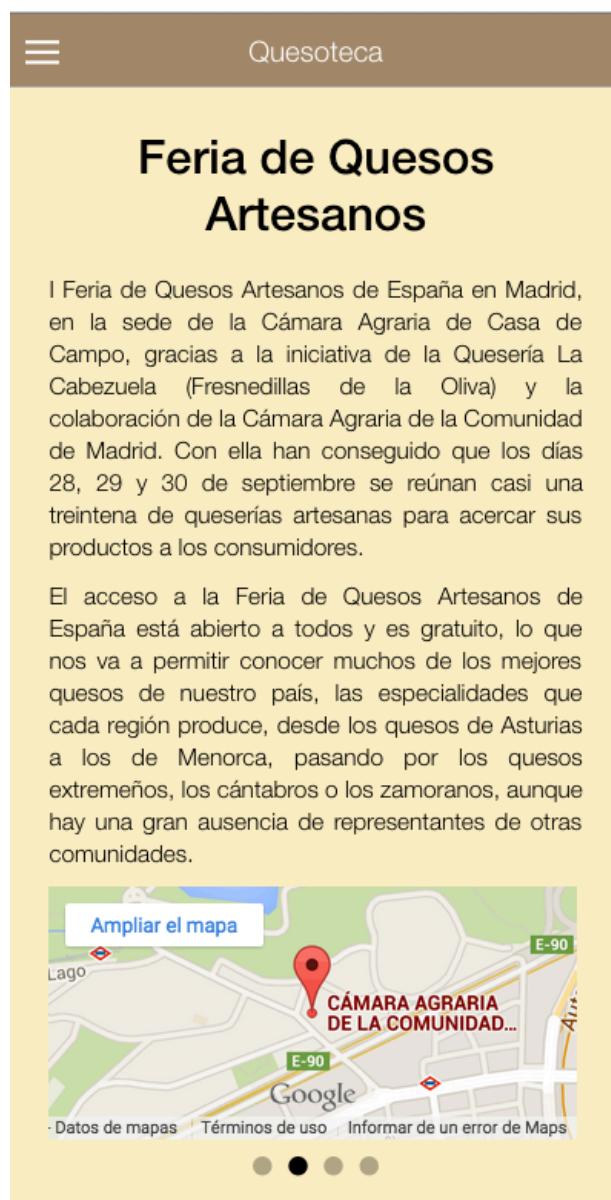


Figura 4.2 Feria Móvil.

5 Evaluación

Después de todo el desarrollo realizado obtenemos una aplicación web muy profesional, muy rápida, muy segura, con diversos tipos de componentes y con distintas formas de navegación.

Al ser una aplicación web estática y no coger datos de ningún tipo de servicio externo la probabilidad de fallo es totalmente nula.

Respecto a la aplicación móvil obtenemos una aplicación muy vistosa y elegante, con un aspecto totalmente nativo y con una velocidad de funcionamiento digna de desarrollo nativo.

Al igual que la aplicación web, los datos que obtiene los estáticos por lo que la posibilidad de fallo es nula.

Las pruebas se han ido haciendo en navegador web ya que es la forma mas rápida de comprobar el resultado.

A la hora de probar la aplicación móvil se ha probado en un iPhone 6 con iOS 8.3 con la ayuda del servicio que aporta Ionic, Ionic View. El funcionamiento de este consiste en descargarte la aplicación de la tienda de aplicaciones de tu Smartphone, crearte una cuenta y desde la terminal de tu ordenador (situado en la raíz del proyecto) introducir el comando Ionic upload. Una vez este subida nos logueamos con nuestra cuenta y contraseña en la aplicación Ionic View y nos aparecerá la aplicación para poder probarla. El rendimiento es realmente sorprendente.

Además de probarlo en un dispositivo real se ha probado en navegador con lo que se ha probado la aplicación con distintos tamaños de ventana viéndose así correctamente.

5.1 Estadísticas aplicación web

Las estadísticas de carga se han realizado con la consola de la suite de herramientas de desarrollo de Google Chrome.

En diferentes pruebas aplicadas quitando la caché y demás no sube de los 400ms en cargar todo el DOM. Cambiemos a la página que sea ese será el mismo tiempo de carga ya que solo realiza una petición al servidor, mejorando así el rendimiento y reduciendo la tasa de transferencia de MB.F

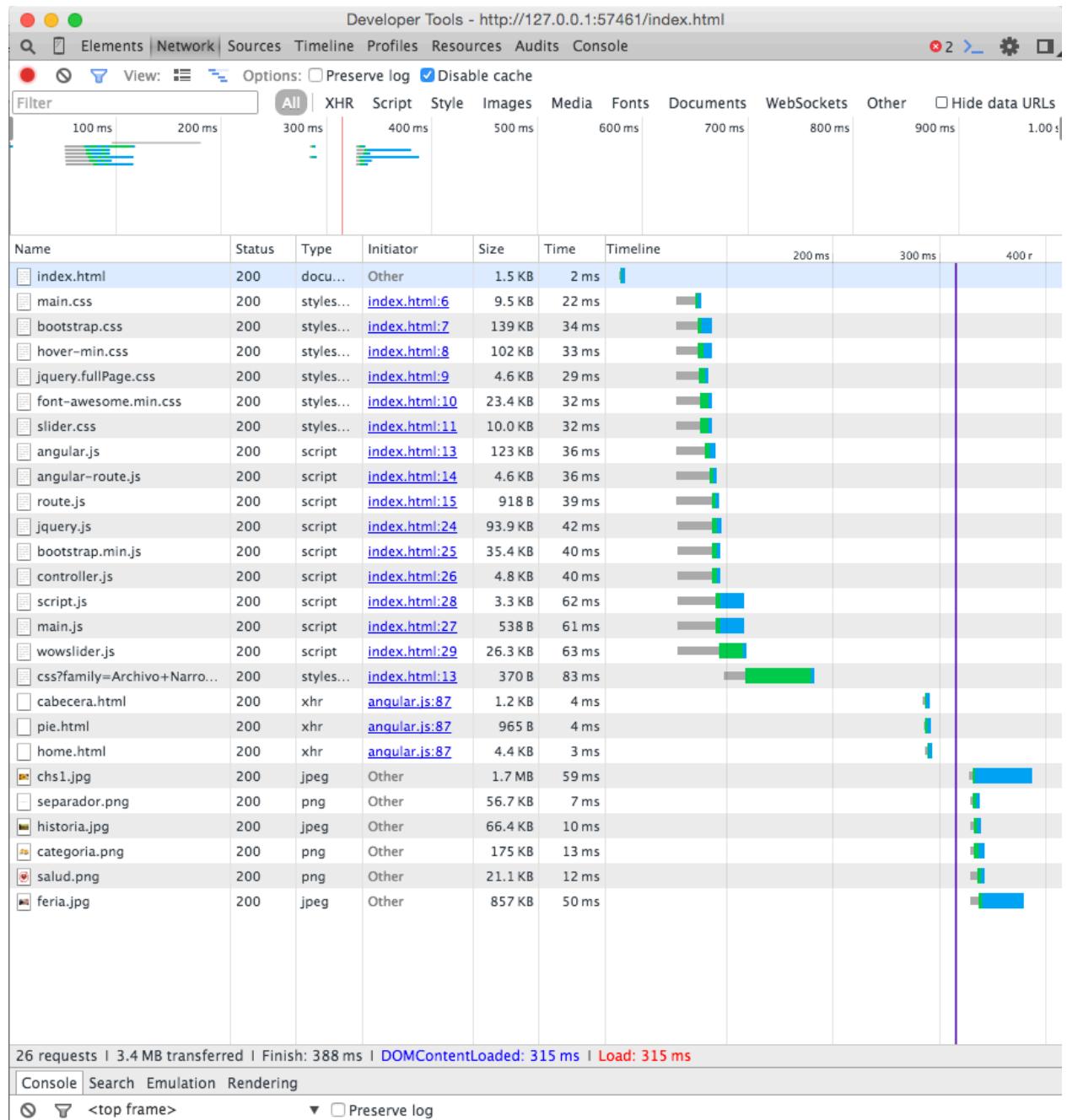


Figura 5.1 Estadísticas aplicación web

5.2 Estadísticas aplicación móvil

Aplicando la misma estrategia para comprobar el rendimiento de la aplicación móvil, obtenemos prácticamente los mismos resultados aunque algo mas reducidos.

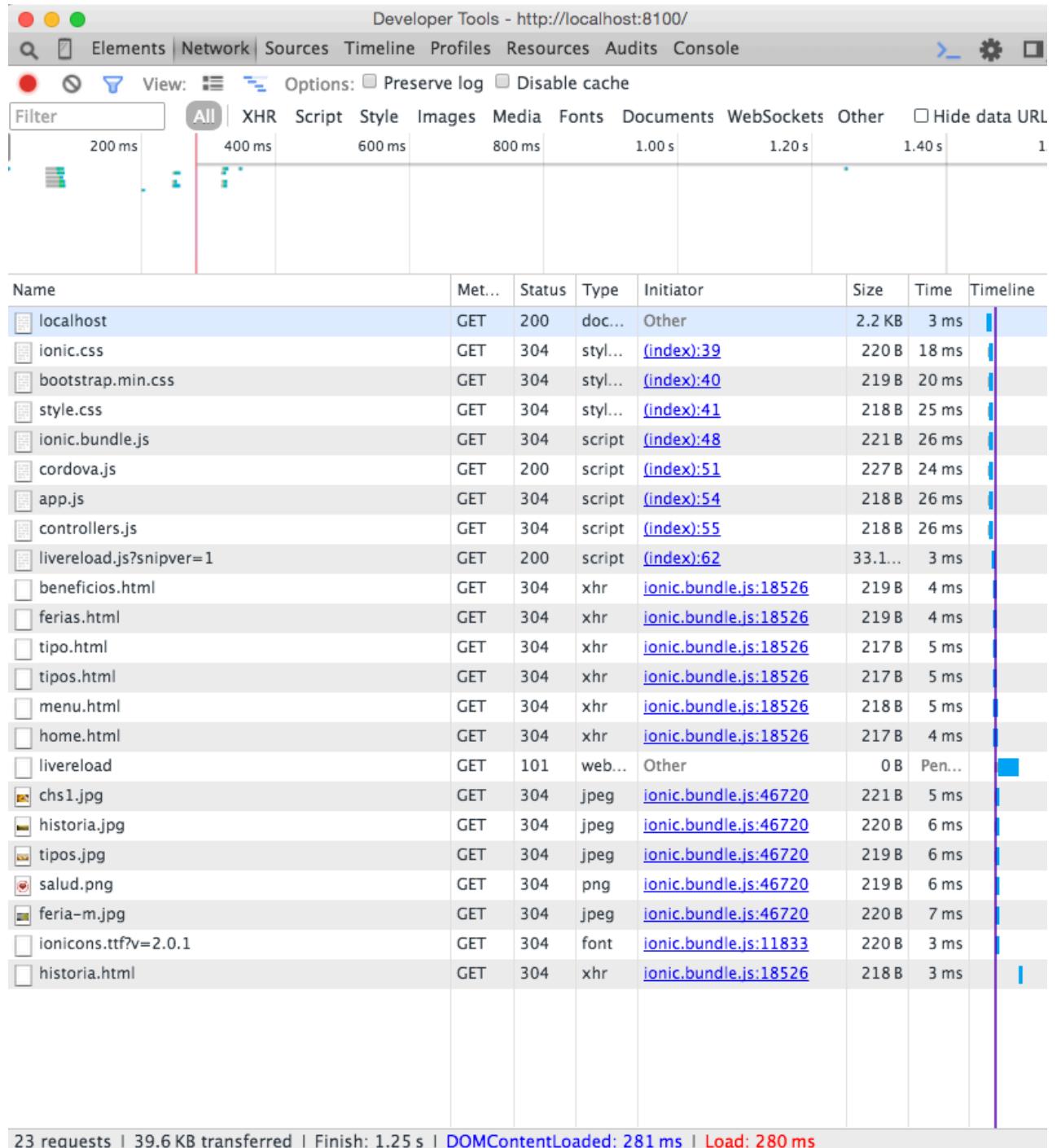


Figura 5.1 Estadísticas aplicación móvil

5.3 Puntos fuertes / Puntos a mejorar

-Puntos fuertes

- Increíble diseño.
- Estética acorde con el tema tanto por las tipografías como por los colores escogidos.
- Gran rendimiento de la aplicación gracias a las tecnologías escogidas.
- Aplicación de una sola creación que se puede exportar a diferentes plataformas (aplicación móvil).
- Usabilidad clara y concisa.
- Uso de tecnologías modernas y punteras.
- Abierta una posibilidad de negocio totalmente valida.
- Modularidad y facilidad de cambio de las aplicaciones.
- Patrón de diseño MVC.
- Sin existencia de fallos.

-Puntos a mejorar

- Cambiar la lógica para que obtenga los datos de un servidor externo.
- La aplicación web se ha podido hacer full responsive, que aunque se haya hecho la aplicación móvil y no haga falta, no habría sido mala solución implantarlo.
- En el tema del diseño, distintos elementos multimedia han podido mostrarse en forma de galería o de popup.

6 Conclusión

Llegado a este punto nos ha quedado claro que el mundo de las aplicaciones web e hibridas están cogiendo un papel muy importante en nuestro día a día ya que el mundo de la web esta empezando a aprovechar y mejorar el motor de todo Javascript.

Encontramos frameworks muy útiles, completos y funcionales como es el caso de AngularJS que como hemos podido comprobar el rendimiento de estas SPA o aplicaciones hibridas es totalmente increíble.

Personalmente opino que el desarrollo de aplicaciones nativas va a dar una considerable bajada de trabajo ya que con frameworks como Phonegap (Cordova) podemos hacer cualquier tipo de aplicación pudiendo acceder a distintas funcionalidades del teléfono, sea cual sea. Sin olvidar la reducción de tiempo y coste del desarrollo de la misma aplicación en distintas plataformas, olvidándonos de esto completamente.

Un desarrollo - múltiples plataformas.

Hay un gran futuro en este mundo como hemos podido comprobar con Quesoteca.

7 Lista de referencias

- [1] Estadísticas de uso de AngularJS -
<http://trends.builtwith.com/javascript/Angular-JS>

