ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación :

INGENIERO TÉCNICO DE TELECOMUNICACIÓN, ESPECIALIDAD EN SONIDO E IMAGEN

Título del proyecto:

DESARROLLO DE WEB APP DE UN CLUB DEPORTIVO OPTIMIZADO PARA DISPOSITIVOS MÓVILES

Ion Iturri Gil
Tutor: Marko Galarza Galarza
Pamplona, 28 de junio de 2012

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

I. Objetivos

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El objetivo del proyecto a realizar consiste en la creación de un portal web de un club deportivo optimizado para dispositivos móviles. Para ello se utilizará lenguaje de programación HTML5 y el framework más adecuado, obteniendo una Web App adaptada a dispositivos móviles.

A la hora de la ejecución del proyecto se tendrán en cuenta que la web tenga un diseño estético y funcional, es decir, la realización de interface agradable al usuario, elección de colores adecuados y tipografías, además de conseguir la mayor facilidad a la hora de navegar por dispositivos móviles.

En la memoria se explicará detalladamente los conceptos para entender el desarrollo de la aplicación web, siempre intentando usar un lenguaje sencillo para que sea entendible por todos los usuarios.

Una vez vistos todos los conceptos relacionados, se estudiara a fondo cada uno de los frameworks más importantes fijándose en las ventajas e inconvenientes de cada uno, para asi, despues de su estudio, seleccionar el framework más conveniente para la creación de la aplicación web.

Por otra parte, se detallan todas las tecnologías utilizadas durante el proyecto, para posteriormente, en la explicación de como se ha procesado el desarrollo de la aplicación web, mostrar ejemplos de cada tecnología utilizada.

Para finalizar se expondrán líneas futura de mejora, para poder mejorar la aplicación, seguido de las conclusiones sobre la aplicación web creada.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

II. ¿Por qué adaptar la aplicación para el móvil?

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Desde que los Smartphones, Tablets, etc. han llegado a la vida de las personas, las conexiones a Internet desde estos dispositivos móviles ha crecido de forma insospechable, y sigue creciendo hasta el punto de que nadie sabe exactamente hasta dónde puede llegar.

Durante el año 2011 Google ha realizado un estudio en 30 países, incluido España, con el objetivo de conocer el uso de internet sobre los smartphones. Los datos obtenidos detallan la enorme importancia que están adquiriendo estos dispositivos en el mundo, ya que de los 7000 millones que habitan el planeta 5000 millones tienen móvil, y el 20% de estos accede a internet desde su móvil.

España concretamente es el principal país de acceso a internet desde el móvil con un 33% del total de usuarios con móvil, mientras que por ejemplo en Inglaterra tienen un 30% y en Francia un 27%. También es destacable que el 93% de los usuarios que tienen internet en un dispositivo móvil lo siguen usandolo en casa a pesar de tener internet en su ordenador.

Respecto al uso de aplicaciones en los dispositivos móviles, las estadísticas marcan que un 84% de los usuarios con estos dispositivos usan apps.

En un futuro cercano se estima que para finales de 2013 los usuarios que tienen Internet en un dispositivo móvil accederán más a Internet desde su propio dispositivo que desde su ordenador. Y en el 2015 se realizarán más búsquedas en todo Internet desde dispositivos móviles que desde el PC.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A continuación, en la *Figura 1*, se muestra un gráfico de la comparación de cómo evoluciona y una estimación de cómo evolucionará Internet en móviles y en ordenadores.

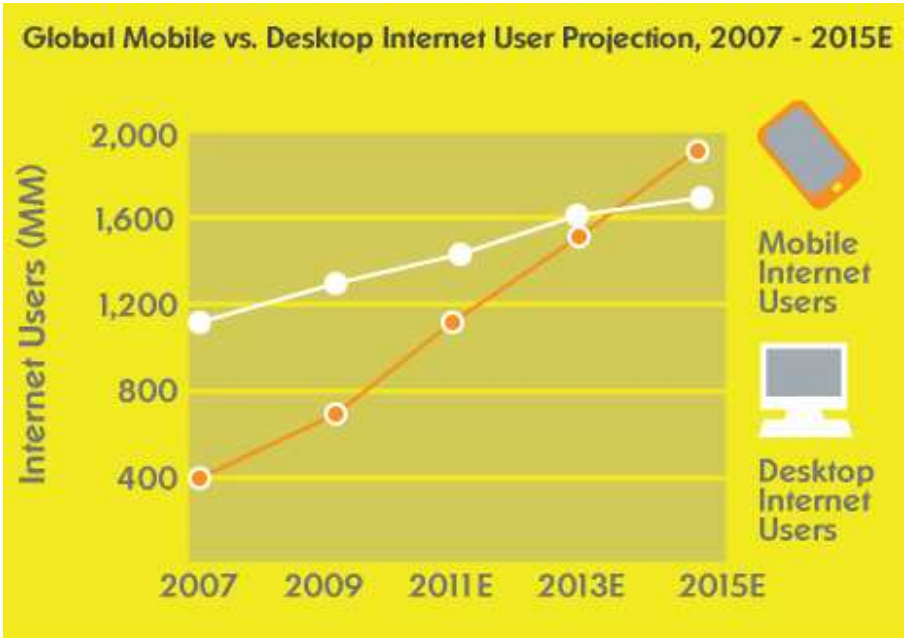


Figura 1: Comparación uso de Internet en móviles y ordenadores.

El uso de los dispositivos móviles tiene las siguientes ventajas y desventajas respecto a los ordenadores.

Ventajas:

Movilidad: La movilidad es una ventaja ya que son fáciles de transportar a donde se desee, por lo general tienen batería que dura tiempo razonable,lo que nos hace más fácil su movilidad. Además debido a la movilidad, una vez detectado mediante satélite, proporciona poder hacer funcion de GPS, encontrar lugares cercanos de interés o proporcionar el lugar del usuario en caso de pérdida entre otras muchas funciones.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Conectividad: Actualmente los dispositivos móviles cuentan con conexión Wifi, Bluetooth, 3G, 4G, que hacen fácil su traslado a cualquier lugar teniendo acceso a Internet desde cualquier parte, cosa que también está relacionada con la movilidad.

Funcionalidades: Aunque antiguamente los móviles simplemente se limitaban a proporcionar funcionalidades de llamadas, actualmente los smartphones brindan otras funciones como fotografías, agenda, albumes, libros electrónicos, juegos, calculadora, Gps, aunque esto también puede verse como una desventaja que más adelante se explicará.

Desventajas:

Funcionalidades: Aunque actualmente con los dispositivos móviles podemos hacer cosas que antes no hubiéramos imaginado, actualmente no cubren las mismas necesidades que un ordenador de escritorio cumple, aunque quizá en algunos años esto pueda ser posible.

Pantalla: El tamaño de la pantalla siempre será limitada, tienen que ser pequeños para poder llevarlo encima con facilidad, ya que de esto depende su movilidad.

Precio: Por sus características precisamente de ser los dispositivos móviles tan pequeños y con muchas funciones , el precio es bastante caro.

Durabilidad: El tiempo de vida de un dispositivo móvil normalmente es menor al de cualquier ordenador. También al ser más pequeños y móviles es más fácil perderlos o que se caigan y se estropeen.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Dejando a un lado las ventajas y desventajas, es destacable que a pesar de que la evolución de los dispositivos móviles se trata de un fenómeno en una aceleración continua, en España sólo el 5% de los grandes anunciantes tiene un sitio optimizado para estos, lo que alerta del desfase existente respecto a la aceleración anteriormente comentada de los dispositivos móviles.

Por lo tanto se llega a la conclusión de que es mucho mejor crear una aplicación web para dispositivos móviles en vez de crearla para una web nativa, ya que en muy pocos años Internet tendra muchos mas usuarios desde dispositivos móviles que desde ordenadores.

A la hora de la elección de hacer la aplicación web para dispositivos móviles, además de lo anteriormente mencionado, toma importancia la movilidad, ya que una parte de la aplicación tiene funcionalidades para dispositivos móviles que pierden su valor en ordenadores de mesa. Este punto se explicará en el desarrollo (apartado IX).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

III. ¿Qué es Web App?

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Una “Web App” es una aplicación software que se codifica en un lenguaje que los navegadores soportan, o dicho de forma más sencilla, es aquella aplicación que el usuario puede utilizar simplemente accediendo a internet mediante un navegador.

Consiste en una aplicación cliente/servidor, donde tanto el cliente (el navegador) como el servidor (web) y el protocolo mediante el que se comunican (HTTP, FTP...) no han de ser creados por el programador de aplicaciones sino que están estandarizados. Para entenderlo mejor se definirán cliente y servidor:

- Cliente: El cliente web es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP o FTP. Esta parte cliente de las aplicaciones web suele estar formada por el código HTML que forma la página web más algo de código ejecutable realizado en lenguaje de script del navegador, (JavaScript) o plugins para visualizar contenido multimedia, aunque esto puede tener problemas de compatibilidad con los diferentes navegadores. Resumiendo, la misión del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen. Las tecnologías más utilizadas para programar el cliente son HTML, CSS, JavaScript y tecnologías que requieren los plugins para su correcto funcionamiento.

- Servidor: El servidor web es un programa que está esperando permanentemente las solicitudes de conexión mediante el protocolo HTTP o FTP por parte de los clientes web. Está formado por diferentes elementos como lo son los documentos HTML, recursos o documentos adicionales (para emplearlos dentro de las páginas o tenerlos disponibles para su descarga y ejecución en el cliente) y programas o scripts (que se ejecutan en el servidor cuando el navegador del cliente lo solicita).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En la *figura 2* se muestra un dibujo del funcionamiento de las transferencias entre cliente y servidor:

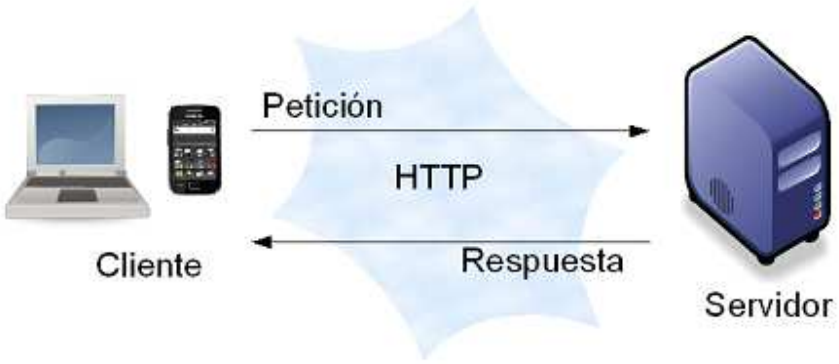


Figura 2: Transferencias Cliente - Servidor

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

III. I. Comparación Web Apps y Native Apps

Cuando se nombran las aplicaciones móviles nos podemos encontrar con 2 tipos: Las aplicaciones web mencionadas en el punto anterior (aplicaciones web) y las aplicaciones nativas. La gran diferencia es que las nativas la tenemos que instalar mientras que las web no. No se puede decidir cual de las 2 es mejor, sino que la elección se basa dependiendo de las necesidades, ya que para algunos campos son mejores las Web Apps y para otros las nativas, todo depende del cliente, de su estrategia a seguir y sus necesidades. Se deben tomar 4 puntos en cuanta a la hora de elegir cual es la más adecuada para cada momento:

- No siempre se tiene conexión a Internet: Si necesitamos un producto sin necesidad de conectarnos a internet en algún momento, la elección debería ser una aplicación nativa, ya que aunque es necesaria su instalación, no se depende de cobertura cada vez que es necesaria su utilización.

- El coste económico: En este apartado la elección no es clara, ya que depende de los recursos que necesitemos puede ser más rentable una aplicación nativa, aunque por lo general suele ocurrir que la web lo sea más.

- Conste de creación: en cuanto a la duración de crear una aplicación o otra, claramente es mejor una Web App, ya que un desarrollador experto puede hacer que en prácticamente una semana tengas dicha aplicación disponible, mientras que una nativa puede llegar a costar en torno a dos meses).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Cliente: En este apartado toma una gran ventaja las aplicaciones web, ya que se debe tener en cuenta que una aplicación nativa es no es igual para un sistema operativo que para otro (Android, iPhone...), por lo tanto es necesario e imprescindible conocer ante qué tipo de clientes estamos, ya que pueden tener dispositivos diferentes. En cambio una aplicación web es compatible con todos los dispositivos utilizables.

Una vez realizada la comparación, se detallarán las principales ventajas y desventajas de las aplicaciones web, que son las que interesan para el proyecto realizado:

Ventajas:

- Portables, se acceden desde cualquier dispositivo con conexión a Internet.
- Los datos están disponibles desde el momento que la aplicación está cargada.
- Siempre se accede a la última versión de la web app, ya que al cargarla, siempre estaremos ante las versiones mejoradas de las mismas.
- Son más seguras que las nativas ya que se ejecutan en un navegador web, es decir, no corremos el riesgo que haya virus y por lo tanto no se pueden infectar las demás aplicaciones.
- No ocupan espacio en el disco duro del dispositivo que la ejecute.
- Son compatibles con todos los sistemas operativos.
- Con la continua actualización de los navegadores, se ofrece mejores funcionalidades.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Desventajas:

- Es imprescindible tener conexión a internet para ejecutarlas, y si ésta se interrumpe mientras la estamos ejecutando, no es posible utilizar la Web App.

- Son menos funcionales que las nativas ya que el uso del navegador limita en gran parte a estas, aunque la aparición de HTML5 ha supuesto un gran avance en este aspecto.

- Al ejecutarse siempre la última versión, el usuario no tiene la capacidad de elegir entre una de ellas.

- Si el desarrollador quiere, esa aplicación puede desaparecer en ese mismo momento, pudiendo dejar al usuario sin capacidad para entrar de nuevo, algo que no ocurre en las nativas, ya que tu las puedes ejecutar hasta que se desinstalen.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

III. II. Comparación Web Apps y Web Desktop

Dado que en el punto anterior están explicadas las Web App, en este apartado se explicará la aplicación desktop con sus ventajas y desventajas respecto a la aplicación web, para así hacer una pequeña comparación.

Una aplicación desktop, también llamada aplicación de escritorio, es aquella que está instalada en el ordenador del usuario, que es ejecutada directamente por el sistema operativo, ya sea Windows, Mac o Linux, y cuyo rendimiento depende de diversas configuraciones de hardware como memoria RAM, disco duro, memoria de video, etc.

Ventajas:

- Habitualmente su ejecución no requieren comunicación con el exterior, sino que se realiza de forma local. Esto influye en mayor velocidad de procesamiento, y por tanto en mayores capacidades a la hora de programar herramientas más complicadas o funcionales.

- Suelen ser más robustas y estables que las aplicaciones web.

- Rendimiento: el tiempo de respuesta es muy rápido.

- Seguridad: pueden ser muy seguras (dependiendo del desarrollador).

Desventajas:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Su acceso se limita al ordenador donde están instaladas.

- Son dependientes del sistema operativo que utilice el ordenador y sus capacidades (video, memoria, etc).

- Requieren instalación personalizada.

- Requieren actualización personalizada.

- Suelen tener requerimientos especiales de software y librerías.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IV. ¿Qué es HTML5?

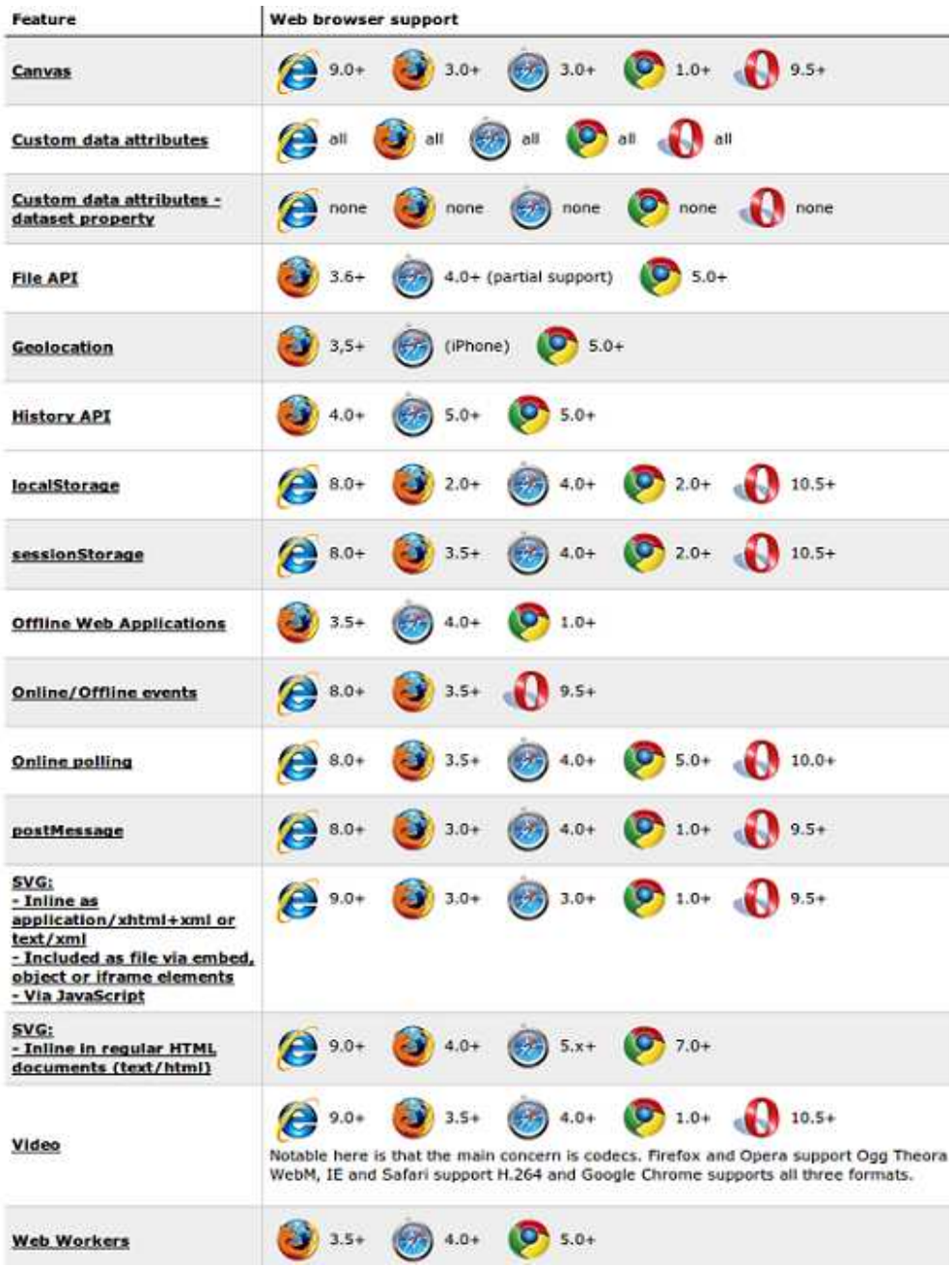
upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El HTML5 ,HyperText Markup Language o lenguaje de marcado de hipertexto, es la quinta mejora del lenguaje de marcado predominante a la hora de crear páginas web, el HTML. Esta nueva versión pretende remplazar al actual HTML, corrigiendo problemas con los que los desarrolladores web se encuentran, así como rediseñar el código actualizándolo a nuevas necesidades que demanda la web de hoy en día, lo cual presenta un problema, la actualización de los navegadores y la compatibilidad con estos. Muchos de los navegadores no están actualizados, por lo que no se puede apreciar el potencial de HTML5, ya que es capaz de mucho más de lo que los navegadores le permiten.

Desde su lanzamiento se esperaba que HTML5 revolucionaria el mundo del diseño web, y que 2012 sería la fecha en que paulatinamente flash dejará de usarse para dar paso a HTML5. Todos los navegadores mayoritarios (Firefox, Chrome, Safari y Opera) han anunciado que en breve darán soporte, por el contrario Internet Explorer hasta su versión 9 no se espera que sea compatible. Todo este proceso, avanzado entre otros por el genio del marketing Steve Jobs antes de su muerte, está teniendo un avance más lento del esperado, cosa que previsiblemente mejorará cuando desaparezcan las incompatibilidades con los navegadores.

En la *figura 3* se muestra la compatibilidad de los navegadores web más utilizados, como lo son Internet Explorer, Mozilla Firefox, Safari, Chrome y Opera, respecto a HTML5:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira



Feature	Web browser support
Canvas	9.0+ 3.0+ 3.0+ 1.0+ 9.5+
Custom data attributes	all all all all all
Custom data attributes - dataset property	none none none none none
File API	3.6+ 4.0+ (partial support) 5.0+
Geolocation	3,5+ (iPhone) 5.0+
History API	4.0+ 5.0+ 5.0+
localStorage	8.0+ 2.0+ 4.0+ 2.0+ 10.5+
sessionStorage	8.0+ 3.5+ 4.0+ 2.0+ 10.5+
Offline Web Applications	3.5+ 4.0+ 1.0+
Online/Offline events	8.0+ 3.5+ 9.5+
Online polling	8.0+ 3.5+ 4.0+ 5.0+ 10.0+
postMessage	8.0+ 3.0+ 4.0+ 1.0+ 9.5+
SVG: - Inline as application/xhtml+xml or text/xml - Included as file via embed, object or iframe elements - Via JavaScript	9.0+ 3.0+ 3.0+ 1.0+ 9.5+
SVG: - Inline in regular HTML documents (text/html)	9.0+ 4.0+ 5.x+ 7.0+
Video	9.0+ 3.5+ 4.0+ 1.0+ 10.5+ Notable here is that the main concern is codecs. Firefox and Opera support Ogg Theora WebM, IE and Safari support H.264 and Google Chrome supports all three formats.
Web Workers	3.5+ 4.0+ 5.0+

Figura 3: Compatibilidad HTML5 - Navegadores Web

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IV. I. Mejoras HTML5

Las principales mejoras que introduce esta versión quinta de HTML con respecto a las anteriores son las siguientes:

- *Estructura del cuerpo:* La mayoría de las webs tienen un formato común (cabecera, pie, navegadores...), y HTML5 permite que todas estas partes representen mediante etiquetas cada una de las partes típicas de una página o aplicación web.

- *Etiquetas específicas:* Se utilizan etiquetas específicas para los contenidos enriquecidos de cada web, como lo son el audio y el video, elementos que hasta ahora se representaban todos con la misma etiqueta.

- *Introducción de Canvas:* Es un elemento que, mediante las funciones de una API (del inglés *Application Programming Interface* o Interfaz de Programación de Aplicaciones, conjunto de procedimientos o funciones que nos ofrece una biblioteca para ser utilizado por otro software) nos permitirá dibujar formas y responder a interacción del usuario, sin necesidad de tener instalado ningún plugin.

- *Aplicaciones Web offline*: Mediante un API es posible desarrollar Web Apps que funcionen sin estar conectados a Internet.

- *Geolocación:* Se pueden localizar geográficamente usuarios de las páginas web (por medio de un API también).

- *Nuevas interfaces de usuario:* Hay nuevos temas muy solicitados que se incorporan a HTML5 para su utilización.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- *Eliminación de etiquetas:* Hay etiquetas en versiones anteriores de HTML que desaparecen ya que su uso carece de provecho. Un ejemplo de esto serían las etiquetas para modificar la presentación del documento, que en HTML5 no se utilizará, y esto correrá a cargo sólamente de CSS (más adelante profundizaremos más en este aspecto).

- *Mejoras en los formularios*: Posibilidad incorporación de nuevos tipos de datos sin tener que utilizar obligatoriamente JavaScript.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IV. II. Nueva estructura HTML5

En esta nueva versión de HTML5, como se ha mencionado al principio del apartado IV, ha mejorado la estructura respecto a sus versiones anteriores con unos cambios de etiqueta más funcionales. Uno de los principales cambios es la desaparición de las etiquetas , que en las anteriores versiones se utilizaba prácticamente en todas las etiquetas, creando una confusión importante para el programador entre otras cosas. Muchos de los problemas de compatibilidad con los navegadores suelen darse por la eliminación de las etiquetas como se explicará más adelante en el apartado del desarrollo del proyecto (apartado IX). A continuación se explicarán los principales cambios uno por uno detallando lo que cada etiqueta proporciona, para conseguir un mejor entendimiento en la materia.

- <section></section> : Representan una sección "general" dentro de un documento o aplicación. Puede contener subsecciones y si lo acompañamos de h1-h6 (formatos de títulos o textos) podemos estructurar mejor toda la página creando jerarquías de contenido, algo muy favorable para el buen posicionamiento web.

- <article></article> : El elemento de artículo representa un componente de una página que consiste en una composición autónoma en un documento o aplicación con la intención de que pueda ser repetido. Se puede utilizar en los artículos de los foros, una revista o el artículo de periódico, una entrada de un blog... es decir, en cualquier artículo independiente de contenido.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- <aside></aside> - Representa una sección de la página que abarca un contenido poco relacionado con el contenido que lo rodea, por lo que se le puede considerar un contenido independiente. Este elemento puede utilizarse para efectos tipográficos, barras laterales, elementos publicitarios, es decir, contenidos que se consideren separados del contenido principal de la página.

- <header></header> : Este elemento representa un grupo de artículos introductorios o de navegación. Básicamente una cabecera para las páginas.

- <nav></nav> : El elemento <nav> representa una sección de una página que es un link a páginas externas o a partes dentro de la página.

- <footer></footer> : Este recurso sirve para representar el pié de una sección, con información acerca de la página/sección que poco tiene que ver con el contenido de la página, como el autor, el copyright, año...

- <audio> y <video> : Estos dos nuevos elementos mencionados antes permitirán insertar un contenido multimedia de sonido o de vídeo. Es una de las novedades más importantes e interesantes en HTML5, ya que permite reproducir y controlar vídeos y audio sin necesidad de plugins. Además, su comportamiento será como el de cualquier elemento nativo, con lo que podremos insertar en un video, enlaces o imágenes.

.

- <embed> : Se emplea para contenido insertado que necesita plugins como el Flash. Es un elemento que ya reconocen todos los navegadores.

- <canvas> : Como ya hemos explicado antes, es un elemento complejo que permite que se generen gráficos al hacer dibujos en su interior. Es utilizado en Google Maps (con lo que para nuestro caso será muy importante).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En la *figura 4* se muestra un la estructura de las etiquetas nuevas en HTML5 explicadas anteriormente:

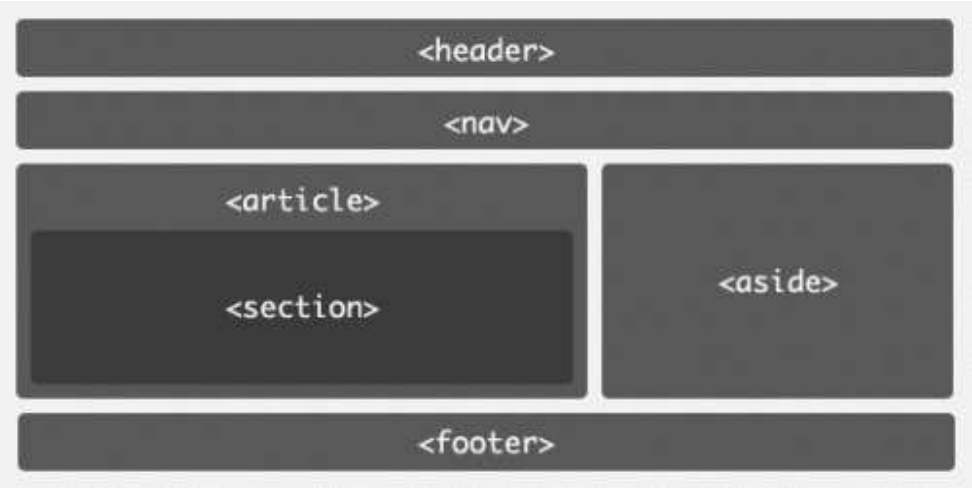


Figura 4: Estructura etiquetas HTML5.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Para finalizar en la *figura 5* se mostrará un pequeño ejemplo de código en HTML5.

Estructura.html

```
<!DOCTYPE html>
<html lang="es">
    <head>
        <meta charset="utf-8" />
    </head>
    <body>
        <header>
            <h1>Titulo tu sitio</h1>
        <nav>
            <ul>
                <li>Enlace 1</li>
                <li>Enlace 2</li>
                <li>Enlace 3</li>
                <li>Enlace 4</li>
                <li>Enlace 5</li>
            </ul>
        </nav>
        </header>
        <section>
            Aquí va todo el contenido de la Web</br>
            <article>
                <h2>Titulo del articulo</h2>
                <p>Contenido</p>
            </article>
            <aside>
                <h3>Contenido Irrelevante</h3>
                <p>Texto</p>
            </aside>
        </section>
        <footer>
            Pie de pagina, copyright, etc.
        </footer>
</html>
```

Figura 5: Ejemplo código HTML5.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

V. ¿Qué es CSS3?

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

CSS, Cascading Style Sheets, o en castellano hoja de estilos en cascada, es un lenguaje que se utiliza para definir la presentación de un documento HTML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas. Actualmente se encuentra en su tercera versión, por lo tanto se conoce como CSS3.

El objetivo CSS consiste en separar el contenido de la forma, y se cumplió ya con las primeras especificaciones del lenguaje. Sin embargo, el objetivo de ofrecer un control total a los diseñadores sobre los elementos de la página ha sido más difícil de cumplir. Las especificaciones anteriores del lenguaje tenían muchas utilidades para aplicar estilos a las páginas o aplicaciones web, pero los desarrolladores todavía continúan usando diferentes trucos para conseguir efectos tan comunes y deseados como los bordes redondeados o el sombreado de elementos en la página.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

V. I. Evolución de CSS

CSS1 : Publicado en el año 1996, este lenguaje incluye un soporte para propiedades de tipo letra, colores de texto, imágenes de fondo, bordes y relleno.

CSS2 : Publicado en el año 1998, este lenguaje añadió a todo lo anterior nuevas propiedades como el posicionamiento absoluto y relativo de los elementos fijos, los tipos de medios y el z-index.

CSS2.1 : Publicado en el año 2005, este lenguaje fue introducido para corregir errores y para eliminar características que el navegador no soportaba.

CSS3: Este lenguaje de programación, que está en desarrollo desde el año 1999, ha experimentado un constante crecimiento hasta llegar a convertirse en una enorme especificación separada en 43 módulos, y que lo hace diferente a las versiones anteriores.

Así, la novedad más importante que aporta CSS3 consiste en la incorporación de nuevos mecanismos para tener mayor control sobre el estilo con el que se muestran los elementos sin tener modificar el código fuente de páginas o aplicaciones web.

En la figura 6 se muestra un ejemplo de código CSS3:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
1  body {
2      font-family: Arial, Helvetica, sans-se
3  }* {
4  margin: 0;
5  padding: 0;     border: 0;
6  color: #000;    font-size: 11px;} #contain
7      margin: 0 auto;}
8  #header h1 a {
9      width: 118px;
10     height: 41px; display: block;position:
11 top: 56px;
12     left: 0;
13     text-indent: -1000px;
14 } a {
15     text-decoration: none;
16 } #content p {   line-height: 1.5em; margin-
17     height: 120px;
18     overflow: hidden;
19     position: relative;
20     border-bottom: 1px solid #918F90;
21 }
22 #nav {
23 position: absolute;
24 top: 90px;
25 right: 0px;
26 }
27 #nav ul {
28 list-style: none;
29 }
30 #nav li {
31 display: inline;
32 padding-left: 22px;
33 }
34 #nav li a {
35 text-transform: uppercase;
```

Figura 6: Ejemplo código en CSS3

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

V. II. Mejoras

Las mejoras que introduce CSS3 respecto a sus versiones anteriores también suponen un problema al igual que ocurre con HTML5 (anteriormente explicado), ya que algunos navegadores son incompatibles y no tienen capacidad de mostrarlas, y por lo tanto el código utilizado sería inservible.

En la *figura 7* se muestra la compatibilidad de CSS3 respecto a los navegadores web principales:

Figura 7: Compatibilidad CSS3-Navegadores Web

Hay una infinidad de mejoras en CSS3, que además de variar el estilo y hacerlo más sencillo e intuitivo, es una herramienta mejor para aquellas personas que quieren desarrollar una página o aplicación web. Vamos a analizar algunas de las más importantes de esta tercera versión de CSS:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Respecto a bordes: Se pueden crear tanto bordes en las imágenes que insertamos como modificar la curvatura, e incluso difuminarlos y crear sombras en ellos. Mención aparte tiene el color, que aunque antes ya se podía modificar, ahora tenemos una mayor gama de colores.

- Respecto a fondos: Anteriormente si insertábamos una imagen de fondo, esta se repetía de borde a borde sin tener en cuenta lo que el usuario quisiese, pero esto con CSS3 ya no ocurre, ya que uno mismo indica cuántos píxeles quiere que ocupe dicha imagen. También se pueden superponer mediante capas distintos fondos.

- Respecto a color: Como ya hemos dicho antes, la gama de colores es mucho más amplia que en versiones anteriores de las hojas de estilos en cascada, pero además, en esta versión 3 se nos permite modificar la opacidad de los elementos que queramos.

- Respecto a texto: Podemos asignar todas las distintas fuentes que encontremos y crear sombras en las letras.

- Otros: La interfaz se puede modificar en su tamaño, se pueden crear distintos tipos de degradados, introducir columnas de texto, crear animaciones...

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VI. ¿Qué es DOM?

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Document Object Model, o en castellano Modelo de Objetos del Documento (DOM) es una interfaz de programación de aplicaciones (API) para documentos HTML y XML. Define la estructura lógica de los documentos y el modo en que se accede y manipula el contenido.

La utilidad de DOM consiste en poder construir documentos, navegar por su estructura, añadir, modificar o eliminar elementos y contenido. Se puede acceder a cualquier cosa que se encuentre en un documento HTML o XML, y se puede modificar, eliminar o añadir usando el DOM.

La creación de DOM surgió debido a que el nacimiento y desarrollo de navegadores suponía que había que cambiar las páginas para cada uno de ellos, por lo que W3C (World Wide Web Consortium) introdujo un modelo de objetos único, con lo que se pretendía favorecer a los desarrolladores de páginas web.

DOM es la estructura de objetos que genera el navegador cuando se carga un documento y que se puede alterar mediante el uso JavaScript para cambiar los contenidos y aspecto de la página. Los objetos del DOM dan forma a la ventana del navegador, historial o página web, y todos los elementos que pueda haber dentro la propia página, como párrafos, tablas, formularios y sus campos...

Mediante DOM se puede acceder, utilizando Javascript, a cualquiera de todos los objetos de estos elementos para alterar sus propiedades y llamar a cada uno de sus métodos o propiedades.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VI. I. Estructura y propiedades

Los documentos del DOM tienen una estructura en forma de árbol, sin embargo, en ningún lugar se especifica que los documentos tengas que ser implementados de esta manera.

En la figura 8 se muestra la estructura en forma de árbol del DOM:

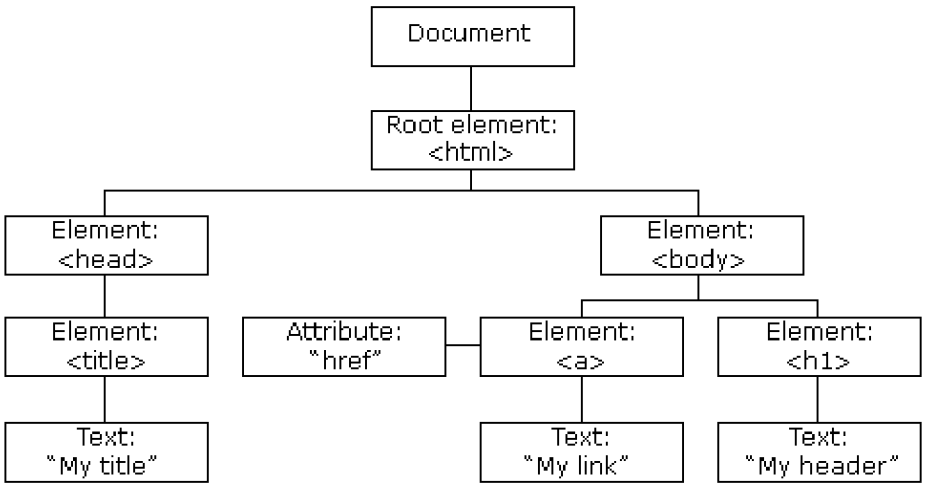


Figura 8: Estructura árbol DOM

Como se puede observar en la figura 8 se pueden distinguir cuatro diferentes tipos de nodos:

- Documento (*Document*): Es el nodo raíz de todos los documentos HTML y XML. Todos los demás nodos derivan de este.

- Elemento (*Element*): Representa los objetos definidos entre dos etiquetas de apertura y cierre (<etiqueta>..Elemento..</etiqueta>).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Atributo (*Attribute*): hace posible que podamos asignar a los objetos cierta característica o valor (color, fuente...) y comentario si deseamos mostrar algo únicamente en nuestro código fuente, sin que aparezca escrito en nuestra aplicación.

- Texto (*Text*): Almacena el contenido de texto que se encuentra entre dos etiquetas.

Dejando a un lado los nodos existentes, hay unas propiedades útiles para todos ellos, de las cuales los más importantes son:

- GetElementById: Devuelve el nodo Elemento con el identificador especificado.

- GetElementsByTagName: Devuelve una lista ordenada de todos los nodos Elemento que tengan como nombre de etiqueta Name.

- CreateElement: Crea un nodo Elemento del tipo que se especifique.

- GetAttribute y SetAttribute: Devuelve o añade el valor del atributo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En la *figura 9* se observa un ejemplo de uso de una de las propiedades mencionadas:

```
function div1ParaElems()
{
  var div1 = document.getElementById("div1")
  var div1Paras = div1.getElementsByTagName("p");

  var num = div1Paras.length;

  alert("Hay " + num + " <p> elementos en el elemento div1");
}
```

Figura 9: Ejemplo de uso de GetElementsByTagName

Otras propiedades específicas de HTML, con las que DOM está directamente relacionado, algunos de los más utilizados son:

- *images:* Documentos de imagen aparecen en el documento.
- *elements:* Conjunto de elementos de un formulario.
- *value:* Valor del contenido en el formulario correspondiente.
- *src:* Lugar de origen de la imagen que deseamos insertar.
- *width* o *height:* anchura o altura del elemento a mostrar.
- *Align*: Permite controlar la alineación de una imagen

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En la figura 10 encontramos un ejemplo de estas propiedades de HTML.

```
<HTML>
<HEAD>
<TITLE>Ejemplo 12</TITLE>
</HEAD>
<BODY>

<H1>Im&aacute;genes</H1>

   <IMG SRC="/graficos/bebe1.jpg" WIDTH=140 HEIGHT=210 BORDER=0 ALT="Un
beb&eacute;" ALIGN="RIGHT">
   <IMG SRC="/graficos/bebe1.jpg" WIDTH=140 HEIGHT=210 BORDER=3 ALT="Un
beb&eacute;" ALIGN="LEFT">
Un texto cualquiera.
</BODY>
</HTML>
```

Figura 10: Ejemplo de uso de propiedades img, src, width, height y align.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. Estudio y elección de los Frameworks

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Primero se debe entender cual es el significado de la palabra Framework: Es una estructura tecnológica, un conjunto de bibliotecas orientadas a la reutilización de componentes software para el desarrollo aplicaciones. Orientado hacia aplicaciones móviles, se podría especificar un poco más como una estructura diseñada con componentes personalizables e intercambiables para favorecer el desarrollo de aplicaciones, sitios y servicios web. Para ello proporcionan bibliotecas para acceder a bases de datos y facilitan la reutilización de código, para así economizar tiempo y reducir la redundancia.

La utilización de Framework aporta una gran ventaja a la hora de desarrollo de aplicaciones, a continuación se detallarán las 3 principales:

- Desarrollo rápido de aplicaciones. Los componentes incluidos en un framework constituyen una capa que libera al programador de la escritura de código de bajo nivel.

- Reutilización de componentes software.

- Un framework orientado a objetos logra que los componentes sean clases que pertenezcan a una gran jerarquía de clases, lo que resulta en bibliotecas más fáciles de aprender a usar.

Para la realización del proyecto se ha decidido hacer el estudio previo, uno por uno, de cinco tipos de frameworks para la realización de aplicaciones web, que son los siguientes:

- jQuery Mobile
- Sencha Touch
- Lungojs
- PhoneGap.
- Titanium Appcelerator

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. I. jQuery Mobile

El primer framework a estudiar está basado en jQuery, creado en 2006 por John Resig bajo el lema "Write less, do more", o lo que es lo mismo, escribir menos y hacer más. Con este lema consiguió resumir en cuatro palabras en qué consiste un Framework, que como anteriormente se ha dicho, el objetivo principal de un framework es no tener que lidiar con las particularidades entre los navegadores y además escribir menos código fuente para hacer cosas más espectaculares.

Perp jQuery va un paso más lejos en la tecnología móvil con jQuery Mobile, ya que debido los dispositivos móviles se han multiplicado los navegadores y plataformas disponibles para el usuario. Existen muchos fabricantes de muchos dispositivos con características distintas, como tamaños de pantallas, sistemas operativos y navegadores personalizados para cada uno de ellos. Es decir, que si antes con los navegadores para ordenadores había problemas de compatibilidad ,siendo estos un número bastante reducido, ahora con los dispositivos móviles estos problemas han aumentado considerablemente. Por esta razón, se hace más necesario el uso de un framework, para realizar el desarrollo una sola vez y ejecutar de manera compatible en todos los dispositivos y navegadores.

Otra pregunta habitual en los Frameworks, es como puede servir de utilidad en el desarrollo de una aplicación web. En el caso de jQuery Mobile, al incluirlo en tu código HTML básico será optimizado para realizar diversos comportamientos dinámicos en los navegadores móviles, de manera automática. Además, muchas cosas del propio framework las vas a poder configurar directamente a través de atributos HTML. Un gran fallo sería pensar que con Jquery Mobile se escribe código HTML.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El CSS también funciona de una manera similar con jQuery Mobile, ya que igual que en HTML, tampoco hay que escribir el código CSS desde jQuery Mobile. Este framework dispone de diversas herramientas CSS, también de manera automática, que podrás utilizar en tus desarrollos. Además, se dispondrá de un pequeño framework CSS para poder hacer cosas como la maquetación a partir de una rejilla, un avance muy importante.

Por otra parte, vamos a hablar de la manipulación de la DOM en jQuery Mobile: Este framework gestiona automáticamente el uso de la DOM. Carga las páginas en la DOM del navegador pero al pasar a la siguiente la página se elimina, por lo que al volver a cargarla (si hacemos un "back" o un enlace externo) la página utiliza la caché del navegador si está disponible, o la pide al servidor en caso contrario.

Pero el DOM lo podemos manipular también mediante el atributo "data-prefetch":

* ... *

Con esta función se cargan las páginas en la DOM antes de visitarlas, es decir, si en la página A, hay un enlace apuntando a la página B con este atributo, la página B se cargará en el DOM cuando se cargue la página A, pero no se mostrará hasta que se pulse el enlace.

Por último, ya hemos hablado antes que la caché podría estar no disponible, por lo que existe un truco para tenerla siempre activada, y sería una función tan sencilla como la siguiente:

$.mobile.page.prototype.options.domCache = true;

En el caso de querer tener desactivada la caché, y activarla solamente para algunas páginas determinadas, con utilizar el atributo "data-dom-cache" en el "page" es suficiente. Así quedaría el page de una página a cachear.

* *

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. I. I. Características

La utilización de librerías JavaScript es una de las principales características de este framework por lo que, al estar basado en JavaScript jQuery, no es un framework desarrollado desde cero, por lo que aquellas personas que ya conocen el framework JavaScript jQuery Mobile en su versión para ordenadores, no van a tener que aprender nada nuevo, sino aplicar sus conocimientos y desarrollar fácilmente aplicaciones para móviles. Esto significa una gran ventaja para aquellas personas que utilizan jQuery, que pueden pasarse sin prácticamente esfuerzo al desarrollo de dispositivos móviles.

Otras de las características de jQuery Mobile se detallan a continuación:

- *Arquitectura de jQueryUI* : Los propios creadores de jQuery usaron su experiencia para desarrollar el framework para móviles y además de implementar la arquitectura diseñada para las librerías de interfaces de usuario jQueryUI.

- *Simplificar el proceso:* jQuery Mobile no se trata simplemente de una capa para realizar código JavaScript que funcione en todos los navegadores, sino un conjunto de herramientas que simplifica el proceso de crear páginas para móviles, desde la escritura en HTML, la maquetación con CSS y la creación de efectos dinámicos con JavaScript.

- *Está desarrollado para trabajar con HTML5:* jQuery Mobile prácticamente obliga a hacer páginas HTML5 para aprovechar todas las características del framework. Esta característica toma importancia debido a los problemas de compatibilidad entre los navegadores y HTML5, por la falta de desarrollo de estos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- *Repleto de automatismos:* Es muy sencillo trabajar con Ajax en jQuery Mobile. De hecho, si el framework capta que puede hacer una conexión de esta manera en lugar de una convencional, lo hace automáticamente. También son automáticas las transiciones entre páginas, la personalización del aspecto de la página, etc.

- *Preparado para dispositivos táctiles:* Los dispositivos táctiles tienen cambios en la gestión de eventos y jQuery Mobile nos facilita la labor de adaptarnos a ellos.

- *Personalización de temas:* En jQuery Mobile podemos elegir entre elegir temas predefinido o crear varios temas personalizados para aplicar al aspecto de nuestra página.

VII. I. II. Compatibilidad

Tenemos que tener en cuenta que además de sus grandes características y facilidad para usarlo, jQuery Mobile es compatible con la mayoría de plataformas que actualmente existen en el mercado. En la *figura 11* podemos ver los logos de los sistemas operativos que jQuery Mobile soporta:

Figura 11: Compatibilidad jQuery Mobile

No obstante, cabe señalar existen diversos grados de compatibilidad para cada sistema, o mejor dicho, para cada navegador dentro de cada familia de dispositivos. El grado de compatibilidad está dividido en tres niveles distintos:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Las especificaciones de cada grado de compatibilidad son las siguientes:

- Grado-A: Sistema completo con animaciones entre transiciones basado en Ajax.
    - Están la mayoría navegadores para iOS y Android, así como BlackBerry , Palm WebOS, los navegadores de ordenadores de escritorio, etc.

- Grado-B: Sistema completo menos Ajax.
    - Encontramos a Symbian, Opera Mini 5.0 y 6.0 para iOS o Balckberry 5.0

- Grado-C: Sistema básico, para smartphones antiguos.
    - El resto de los smartphones, entre los que se encuentra Windows Mobile, Bada o Blackberry 4

Como conclusión podemos sacar que ahora mismo no hay ningún dispositivo móvil que no tenga compatibilidad con jQuery Mobile.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. I. III. Ejemplo

Para entender mejor la aplicación de jQuery Mobile sobre HTML5 se va a seguir un ejemplo con la explicación de este. En la *figura 12* se observa el código HTML5 sobre el que esta aplicado el Framework:

```
<!DOCTYPE html>
<html>
    <head>
    <title>Ejemplo jQuery Mobile</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0b2/jquery.mobile-1.0b2.min.css" />
    <script type="text/javascript" src="http://code.jquery.com/jquery-1.6.2.min.js"></script>
    <script type="text/javascript" src="http://code.jquery.com/mobile/1.0b2/jquery.mobile-1.0b2.min.js"></script>
</head>
<body>
<section data-role="page" id="home" data-title="Home">
    <header data-role="header" data-position="inline" data-theme="a">
        <a href="http://www.facultia.com/blog" data-icon="arrow-u" data-theme="b" rel="external">Blog</a>
        <h1>Grupos</h1>
        <a href="#login" data-icon="check" data-theme="b">Login</a>
    </header><!-- /header -->
    <section data-role="content">
        <ul data-role="listview" data-theme="a" data-dividertheme="b" data-inset="true">
            <li data-role="list-divider">Internacionales</li>
            <li><a href="#">The Beatles</a></li>
            <li><a href="#">Bob Dylan</a></li>
            <li><a href="#">Nick Drake</a></li>
            <li data-role="list-divider">Nacionales</li>
            <li><a href="#">Los Planetas</a></li>
            <li><a href="#">La Casa Azul</a></li>
            <li><a href="#">La Buena Vida</a></li>
        </ul>
    </section><!-- /content -->
    <footer data-role="footer" data-position="fixed">
        <section data-role="navbar">
            <ul>
                <li><a href="#" class="ui-btn-active" data-icon="refresh" data-theme="b">Actualizar</a></li>
                <li><a href="#" data-icon="search" data-theme="b">Buscar</a></li>
                <li><a href="#" data-icon="gear" data-theme="b">Ajustes</a></li>
            </ul>
        </section><!-- /navbar -->
    </footer><!-- /footer -->
</section><!-- /page home -->
</body>
</html>
```

Figura 12: Ejemplo de código sencillo con jQuery Mobile

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El resultado del ejemplo de código de la *figura 12* tiene como resultado en los navegadores la *figura 13*:

Figura 13: Ejemplo de la figura 12 visto en navegador

El ejemplo mostrado en las *figuras 12 y 13* tienen las siguientes características:

- El DOCTYPE indica que es una página HTML versión 5.

- En la sección <head> de la página HTML importamos las librerías de JQuery Mobile, así como la hoja de estilos de JQuery Mobile.

- La página HTML está compuesta por tres páginas lógicas, las cuales están delimitadas por las etiquetas <section id= ”paginaID” data-role= ”page”> y </section>

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Cada página (section) está compuesta por los elementos:
    - <header data-role="header">: Cabecera de la página.
    - <section data-role="content">: Cuerpo de la página.
    - <footer data-role="footer">: Pie de la página.

- El atributo data-role=”page” indica a JQuery Mobile lo que es una página, una cabecera, cuerpo o pie, no es ni el header ni el section ni el footer.

- Cuándo se carga el HTML, JQuery Mobile analizará el HTML (con JavaScript), lo modificará internamente en base a los atributos que reconoce y mostrará la primera página que se encuentre.

- Para navegar de una página a otra, simplemente debemos crear un enlace y establecer el identificador de la página destino en su atributo href, o en ese href escribir el nombre de la página a la que quieres acceder..

- Para aplicar un efecto a la transición entre páginas debemos poner el atributo data-transition="....." (slide, slidedown, slideup, pop o flip).

- Si queremos crear un enlace de un botón debemos escribir el atributo data-role="button".

- Si deseamos decorar un botón con una imagen debemos especificar en el enlace el atributo data-icon="..."(home, back, star,...).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. II. Sencha Touch

La empresa Sencha se creo que resultó de la unión de ExtJs, jQTouch y Raphaël. Al igual que jQuery Mobile, se basa también en librerías JavaScript. Es una framework que, permite desarrollar aplicaciones web para dispositivos móviles táctiles, dando prácticamente la sensación de que son aplicaciones nativas de los sistemas operativos de los dispositivos.

Sencha Touch fue el primer framework construido a partir de estándares web, y diseñado para aprovechar específicamente la potencia, flexibilidad y optimización del lenguaje HTML5, CSS y JavaScript. La utilización de HTML5 le ha dotado la posibilidad de ofrecer componentes como el audio o el video, además de un proxy local para almacenar información incluso estando sin conexión a la red.

Respecto a su hoja de estilos, el CSS, este framework ha implementado una capa de estilos independiente de la resolución que tengan los dispositivos móviles. Gracias a una combinación de tamaños CSS3, ha logrado adaptar los componentes de una interfaz de usuarios a la resolución de cada dispositivo (deben ser compatibles, por supuesto). Además, se aprovecha de una tecnología diferente de los otros frameworks (SASS) que se basa en implementar una capa de CSS en la que se añaden variables y funciones, con lo que se puede cambiar la presentación de una aplicación totalmente. Al igual que el antes explicado jQuery Mobile, este framework proporciona una librería con múltiples widgets como pestañas, formularios, listas... siempre pudiendo utilizar temas generados por el usuario.

En cuanto a la manipulación del DOM, es destacable que su librería muy grande, con colecciones, paquetes y clases para la reutilización de código, al igual que jQuery Mobile. Todo se basa en heredar y codificar nuestros propios objetos, diseños y componentes con códigos que ya existen. A continuación, se hará referencia a las clases principales del Framework siguiendo su jerarquía de herencia o el árbol DOM.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Mediante el siguiente script se consigue llamar al DOM:

```
<script type="text/javascript">

Ext.onReady(function() { //se usa para inicializar los componentes en el momento adecuado
  Ext.BLANK_IMAGE_URL = "common/ext/resources/images/default/s.gif";
  Ext.QuickTips.init();

});

</script>
```

"Ext.onReady" es la función más importante, sirve para inicializar los componentes justo cuando el DOM ya ha sido cargado lo cual significa que cuando el DOM esté listo ejecutará la acción. Otra función interesante es "renderTo", que recibe un elemento DOM para renderizar el componente dentro de éste elemento.

Las funciones relacionadas con HTML5 son prácticamente las mismas que jQuery Mobile , como es lógico, ya que son propias del lenguaje HTML. La diferencia esta en el lenguaje diferente que usan dentro de Javascript.

VII. II. I. Características

Tiene grandes características para decantarnos por este framework: Para desarrollar aplicaciones dinámicas, Sencha Touch presenta un conjunto de datos muy potente y robusto que permite solicitar y obtener datos a través de diferentes fuentes como Ajax (como ya lo hacía jQuery Mobile). Además puede unir esos datos a plantillas o a listas HTML.

A continuación se mostrarán las ventajas de Sencha Touch:

- Nos permite crear aplicaciones complejas utilizando componentes predefinidos, como también es el caso de jQuery Mobile

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Evita el problema de tener que validar el código para que funcione bien en cada uno de los navegadores (Firefox, Chrome, Internet Explorer, Safari, Opera etc.).

- Una ventaja que resalta, es su funcionamiento de las ventanas flotantes, que en este aspecto es superior a cualquier otro framework.

- Relación entre Cliente-Servidor balanceado: Se distribuye la carga de procesamiento entre ellos, permitiendo que el servidor pueda atender más clientes al mismo tiempo, es decir, que se repartan el uso de recursos entre esas dos partes.

- Eficiencia de la red: Disminuye el tráfico en la red pues las aplicaciones cuentan con las posibilidad de elegir qué datos desea transmitir al servidor y viceversa.

- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a una acción del usuario, es decir, se puede cargar información sin que el cliente se de cuenta.

- Al ser "herencia" de ExtJs, este framework estaba desarrollado en su mayor parte, por lo que no se ha requerido de demasiados cambios para hacerlo funcional, a la vez de evitar muchos de los fallos que ocurren al iniciar un proyecto desde cero.

- Hay una gran cantidad de componentes y muy diversos, por lo que podemos crear el diseño de una Web App de muchas formas diferentes con un código similar.

- Al haber una empresa como Sencha detrás de este framework, su desarrollo se vuelve mucho más sencillo de realizar.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Soporte para geolocalización y mapas, incluyendo un componente que implementa el API de Google Maps.

Pero no todo son ventajas, tiene alguna desventaja de bastante peso para el usuario a la hora de elegir framework:

- Al ser el primer framework que apareció, los errores o bugs que había debían ser corregidos sin poder apoyarse en otros.

- Es uno de los frameworks más pesado, es decir, que los recursos de sus aplicaciones ocupan más que los de la competencia, detalle por el cuál muchos usuarios pueden dejar de lado a Sencha Touch y decantarse por otro.

- El hecho de ser aplicaciones más pesadas hace que muchas de las apps sean lentas y haya sobrecarga en ellas, lo cual vuelve a ser un problema de peso para algunos usuarios.

VII. II. II. Compatibilidad

Otra de las grandes desventajas que tiene este framework es su compatibilidad con los dispositivos móviles. Se ha creado la versión 2 de Sencha Touch, que aunque tiene muchas variantes y mejoras que su primera versión, todavía está desarrollada en HTML5 para muy pocos sistemas operativos de móviles. Únicamente es compatible con Iphone (todos sus productos), Android (versiones desde 2.3 hacia adelante) y algunos dispositivos de BlackBerry (un grupo muy reducido). A pesar de cubrir una gran parte del mercado ya que cobre Iphone y Android, está un nivel por debajo de jQuery Mobile en este aspecto.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. II. III. Ejemplo

Al igual que en apartado de jQuery Mobile, se expondrá un ejemplo de cómo funciona Sencha Touch, con su código y el ejemplo de navegador, detallando las partes más importantes de este.

En las *figuras 14* se puede ver un ejemplo de código HTML5 para Sencha Touch, mientras que en la *figura 15* se observa la utilización de JavaScript en este Framework.

```
view plain  copy to clipboard  print  ?
<!DOCTYPE html>
<html>
<head>
    <title>Iniciando con el desarrollo en Sencha Touch</title>
    <link rel="stylesheet" href="touch/resources/css/sencha-touch.css" type="text/css">
    <script type="text/javascript" src="touch/sencha-touch-all.js"></script>
    <script type="text/javascript" src="app.js"></script>
</head>
<body>
</html>
```

Figura 14: Ejemplo de HTML5 con Sencha Touch

Para el código HTML5 lo único que se necesita son dos archivos y una copia de la carpeta de Sencha Touch. Los archivos son:

- *app.js* : Archivo donde se definen las configuraciones de tu aplicación.
- *Touch* : La copia de la carpeta descargada de Sencha Touch.
- *index.html* : una página inicial de HTML que incluye el archivo principal de Sencha Touch.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Tiene la misma forma que framework jQuery Mobile pero las librerías de Javascript son diferentes. La forma de crear los interfaces también es totalmente diferente, aprovechándose de la utilización del archivo *app.js* comentado anteriormente.

En la figura 15, como se ha mencionado antes, se nota la gran diferencia respecto a jQuery Mobile, observando la función TabPanel, que crea una interfaz de paneles agrupados en tabs, siendo en este ejemplo una imagen con el logo de Sencha lo insertado.

```
Ext.application({
    name: 'Sencha',

    launch: function() {
        Ext.create("Ext.TabPanel", {
            fullscreen: true,
            tabBarPosition: 'bottom',

            items: [
                {
                    title: 'Home',
                    iconCls: 'home',
                    html: [
                        '<img src="http://staging.sencha.com/img/sencha.png">',
                        '
<h1><cufon style="width: 153px; height: 40px;" alt="Welcome " class="cufon cufon-canvas">
<canvas style="width: 194px; height: 42px; top: -1px; left: -5px;" height="42" width="194"></canvas>
<cufontext>Welcome </cufontext></cufon><cufon style="width: 38px; height: 40px;" alt="to " class="cufon cufon-canvas">
<canvas style="width: 79px; height: 42px; top: -1px; left: -5px;" height="42" width="79"></canvas><cufontext>to </cufontext>
</cufon><cufon style="width: 122px; height: 40px;" alt="Sencha " class="cufon cufon-canvas">
<canvas style="width: 163px; height: 42px; top: -1px; left: -5px;" height="42" width="163"></canvas><cufontext>Sencha </cufontext>
</cufon><cufon style="width: 90px; height: 40px;" alt="Touch" class="cufon cufon-canvas">
<canvas style="width: 121px; height: 42px; top: -1px; left: -5px;" height="42" width="121"></canvas><cufontext>Touch</cufontext>
</cufon></h1>

',
                        "

You're creating the Getting Started app. This demonstrates how ",
                        "to use tabs, lists and forms to create a simple app
                    ].join("")
                }
            ]
        });
    }
});
```

Figura 15: Utilización de JavaScript para Sencha Touch

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El resultado de el código de los ejemplos visto en un navegador compatible con Sencha Touch sería lo que se ve en la *figura 16*. Donde podemos observar el logo de Sencha (en el código *img src:*) y el icono de 'home' con su nombre (mediante *title* e *iconCls*).

Figura 16: Ejemplo en navegador del código de las figuras 14 y 15

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. III. LungoJS

LungoJs es otro de los frameworks que se basan en HTML5, CSS3 y JavaScript para desarrollar aplicaciones web. Es uno de los últimos frameworks JavaScript que se han creado pero sus grandes capacidades y herramientas lo colocan a la altura de jQuery Mobile o Sencha Touch.

Aunque LungoJs parece una copia de jQuery Mobile, dado que la mayoría de las características y recursos que este último posee, las utiliza el framework que se estudia, de manera que prácticamente no hay diferencia entre ellos.

Una cosa que sí hay que tener en cuenta, que así como jQuery Mobile necesitaba de dos scripts para adaptar la aplicación móvil, LungoJs necesita de cuatro (además del que asigna a un documento HTML el uso de este framework). Se explicará detalladamente cuáles son y para qué sirven cada uno de ellos:

- App.js : Se encarga de iniciar LungoJs y la aplicación web. En este se pueden indicar el nombre, versión del proyecto, secciones... Si se quiere que algo se arranque al principio también se debe hacer en este fichero. Hay que tener en cuenta que este script es el primero que se lanza y el orden, en este framework es muy importante (si no se respeta puede llevar a tener muchas confusiones).

- Events.js : Los eventos JavaScript son bastante importantes para una aplicación web ya que llevarán el peso de toda la lógica del programa. LungoJs provee eventos táctiles como ya hemos dicho, algunos de los cuales son el tap (pulsar pantalla), swipe (arrastrar la pantalla)...

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- View.js : Nos sirve para mostrar las vistas, listas, barras de navegaciones, transiciones... es decir, todo lo que tenga que ver con lo visual y deba ser dinámico debe estar aquí.

- Data.js : Este script sirve para crear una base de datos si es que es requerido. Aparte de esto, es muy importante debido a que la caché se almacena aquí, por lo que si lo eliminamos se podrían perder los datos de la aplicación.

Aunque estos son los cuatro imprescindibles, también hay más ficheros JavaScript que se podrían introducir, aunque no necesarios para su funcionamiento, como son services.js (para datos externos), o DOM.js (para manipular el DOM como ya lo hacían los dos anteriores frameworks estudiados).

VII. III. I. Características

La mejor de las características de LungoJS es el "easy prototype", es decir lo fácil que es crear prototipos de dichas aplicaciones sin introducir ninguna línea de código JavaScript. únicamente con la semántica del documento HTML5 y con la aplicación de comportamientos en los elementos podemos crear lo que puede ser la estructura general de nuestra aplicación web.

Las demás características del Framework, no exentas de importancia, se explican en la siguiente lista:

- Diseño y creación de aplicaciones para iOS, Android, Blackberry y Windows Phone 7.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Diseñado para aprovechar las características de los dispositivos móviles actuales.

- Captura de eventos como Swipe, tap, double-Tap (efectos táctiles en los dispositivos móviles).

- Se puede distribuir las aplicaciones en "Mobile Stores"(Market, App Store..) o en páginas web.

- Implementa capacidades de HTML5 como geolocalización, orientación del dispositivo...

- Implementa una interfaz con un estilo de aplicación nativa (en iPad y iPhone).

- Totalmente personalizable.

VII. III. II. Compatibilidad

A la hora del desarrollo del Framework no se le dio demasiada importancia a la compatibilidad de este con los dispositivos móviles, dando compatibilidad y testeando solamente los sistemas operativos con mayor mercado en el momento. Con esto consigue acaparar la mayoría del mercado, pero se encontrarían desajuste y fallos en los demás sistemas operativos debido a la falta de esta compatibilidad. Estos sistemas operativos son:

- iOS
- Android
- Blackberry
- WebOs

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Es probablemente la mayor desventaja que puede tener respecto a jQuery Mobile. En cuanto a la comparación con Sencha Touch se podría afirmar que la compatibilidad es parecida, aunque LungoJS lo supera mínimamente.

VII. III. III. Ejemplo

Se enseñara un ejemplo del código mediante LungoJS para encontrar las diferencias con los frameworks anteriormente explicados. En la *figura 17* se observa el *app.js* de LungoJS, un ejemplo de como usa JavaScript este framework.

```
var App = (function(lng, undefined) {

    //Define your LungoJS Application Instance
    lng.App.init({
        name: 'BuscaTweets',
        version: '1.1'
    });

    return {

    };

})(LUNGO);
```

Figura 17: Ejemplo codigo app.js

En la próxima figura, la *figura 18*, se puede observar un código sencillo en HTML5 de LungoJS, sobre todo debemos fijarnos en la parte final del archivo donde carga todo lo referente a este Framework:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
<!doctype html>
<html manifest="index.appcache">
<head>
    <meta charset="utf-8">
    <title>LungoJS</title>
    <meta name="description" content="">
    <meta name="author" content="Javier Jiménez Villar (@soyjavi)">
    <!-- Mobile viewport optimization http://goo.gl/b9SaQ -->
    <meta name="HandheldFriendly" content="True">
    <meta name="MobileOptimized" content="320">
    <meta http-equiv="cleartype" content="on">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0;">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <!-- For iPhone 4 with high-resolution Retina display: -->
    <link rel="apple-touch-icon-precomposed" sizes="114x114" href="assets/images/icon@2x.png">
    <link rel="apple-touch-icon-precomposed" sizes="72x72" href="assets/images/icon-72.png">
    <link rel="apple-touch-icon-precomposed" href="assets/images/icon.png">
    <link rel="apple-touch-startup-image" href="assets/images/default.png">
    <!-- Main Stylesheet -->
    <link rel="stylesheet" href="./lungo.js/lungo-1.1.2.min.css">
    <link rel="stylesheet" href="./lungo.js/lungo.theme.default.css">
</head>

<body class="app">
    <!--
        First, you have to do is create a LungoJS Application instance in the file app.js.
        ...and use a Webkit browser as Chrome or Safari.
    -->
    <section id="hello_world">
        <header data-title="Hello world!">
            <a href="#" data-target="aside" data-icon="files" class="button"></a>
        </header>

        <footer></footer>

        <aside></aside>

        <article></article>
    </section>

    <!-- LungoJS (Production mode) -->
    <script src="lungo.js/lungo-1.1.2.packed.js"></script>
    <!-- LungoJS - Sandbox App -->
    <script src="app/app.js"></script>
    <script src="app/data.js"></script>
    <script src="app/events.js"></script>
    <script src="app/services.js"></script>
    <script src="app/view.js"></script>
</body>
</html>
```

Figura 18: ejemplo código HTML5 usando LungoJS

Se observa un section con un header, un footer, un aside y un article. Estos elementos se van a visualizar nada más descomentar las líneas del app.js, teniendo el aspecto de la *figura 19* en los dispositivos móviles:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

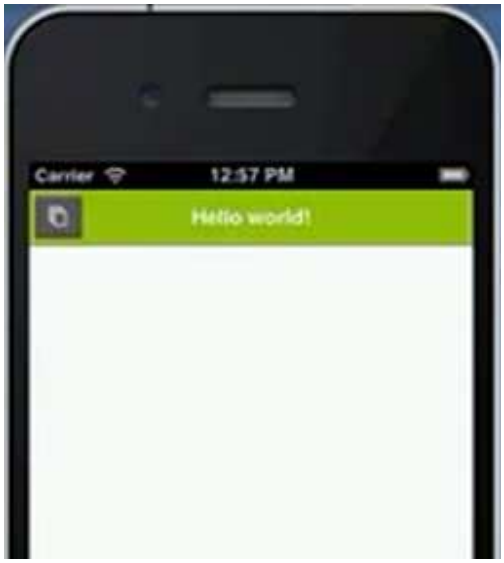


Figura 19: Ejemplo en navegador de LungoJS

En el ejemplo se encuentra sencilla aplicación en el cual se tiene un 'Hello World' con un interfaz totalmente neutro, pudiendo el usuario extender dicho interfaz a través de la creación de nuevo temas con CSS3.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. IV. PhoneGap

PhoneGap, junto a otro Framework que se explicara en el siguiente punto (Titanium Appcelerator), tiene la diferencia respecto a los 3 anteriores de que sirve para convertir una aplicación web en una nativa, con los beneficios e inconvenientes que esto conlleva. Por ello, las aplicaciones deben ser ejecutadas dentro en un componente del móvil.

PhoneGap es un sistema para crear aplicaciones usando exclusivamente HTML5, CSS3 y Javascript, al igual que en los tres frameworks estudiados antes.

Tiene una serie de librerías Javascript desarrolladas en el lenguaje específico de cada plataforma que nos permiten acceder a las características del móvil como GPS, acelerómetro, cámara, contactos... Al ser en principio una página web, tenemos acceso al DOM y podemos usar frameworks para complementar la página. Requiere diseñar tu aplicación web con los componentes visuales típicos del HTML o usar un framework como jQuery Mobile, Sencha Touch o LungoJs, es decir, más que un framework propiamente dicho, es un complemento a los tres anteriormente estudiados.

Con PhoneGap se crean una serie de páginas web que están almacenadas y empaquetadas dentro de una aplicación móvil visualizadas con un navegador web. Además, con acceso a la mayoría de APIs del móvil (cámara, geolocalización, acelerómetro...), lo cual lo convierte en una alternativa muy sencilla para crear aplicaciones, aunque en el modo que se desea usar para el proyecto funciona más como un transformador a aplicaciones nativas que como un framework.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. IV. I. Características

Una vez hecho una explicación del funcionamiento de PhoneGap, se van a detallar las características de PhoneGap basándose en las ventajas y desventajas que tiene:

Ventajas:

- Es compatible con un gran número de plataformas móviles soporta, algo que en este aspecto le hace estar por delante de Sencha Touch o LungoJs pero no de jQuery Mobile. Además de Iphone/Ipad y Android, funciona también en Palm, Symbian, WebOS, Windows 7 y BlackBerry.

- Es muy fácil de desarrollar y proporciona una gran libertad a los que tienen conocimientos de HTML y Javascript (igual que los 3 anteriores frameworks).

- Hay mucha y muy buena documentación y bastantes ejemplos sobre el framework.

Desventajas:

- Cada plataforma necesita de diferente programación para crear la aplicación web, con el inconveniente claro de no saber utilizar los programas para todas ellas.

- La aplicación no es más que una página web, por lo que el aspecto dependerá del framework web utilizado, por lo anteriormente explicado se depende de frameworks específicos tipo jQuery Mobile, Sencha Touch o LungoJS para que una web app en PhoneGap sea vistosa.

- Es una mezcla entre aplicación nativa y web app, detalle importante a la hora de su elección debido al interés por un framework específico para aplicaciones web..

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. IV. II. Compatibilidad

Otro punto para desechar la opción de PhoneGap es que cada plataforma posee una forma distinta de desarrollo, es decir, no es lo mismo crear una aplicación web para Iphone o Ipad, utilizando Xcode, que para un sistema Android, utilizando Eclipse). Esto supone que la creación sea más costosa o perder la compatibilidad con alguno de los sistemas operativos.

Como se ha adelantado brevemente en el punto de las características, es compatible con la mayoría de las plataformas móviles, ganando esta batalla a los anteriormente explicados Sencha Touch o LungoJs. En este aspecto jQuery Mobile sigue siendo superior, ya que es compatible con todas las plataformas presentes actualmente en el mercado. Como se puede observar en la figura 20, es compatible en mayor o menor medida con con las características de iOS, Android, Palm, Symbian, WebOS, Windows 7 y BlackBerry.

Figura 20: Compatibilidad de PhoneGap

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. V. Titanium Appcelerator

Titanium Appcelerator, al igual que PhoneGap explicado en el punto anterior, tiene la diferencia respecto jQuery Mobile, Sencha Touch y LungoJS de que sirve para convertir una aplicación web en una nativa. Es una plataforma para desarrollar aplicaciones móviles y de escritorio utilizando tecnologías web (HTML, CSS, JavaScript...). Traduce las habilidades aprendidas en aplicaciones nativas que se ven y operan tal como si hubiesen sido escritas en Objective-C o Java. Es prácticamente un transformador de aplicaciones web a nativas, al igual que PhoneGap.

Con Titanium es posible crear aplicaciones para Android, Iphone y, además, de escritorio, usando exclusivamente Javascript. Para programar con este framework se utiliza Titanium Studio, con el que se crean los proyectos y editan los ficheros Javascript y el resto de recursos.

Mediante Titanium, gracias a Javascript, se crean manualmente todos los controles, usando para ello una librería que hace de puente entre tu aplicación Javascript y los controles del sistema. Esto significa que, los controles visuales (botones, listas, menús, etc) son nativos: Al insertar un botón, se crea un botón del sistema y se añade a la vista, lo que lo hace que la respuesta del usuario sea más rápida. A diferencia de PhoneGap, en Titanium no hay DOM, por lo que no se pueden usar frameworks tipo jQuery Mobile, Sencha Touch o LungoJS para manipularlo, ya que no se ejecuta dentro de un documento HTML.

El paso de aplicación web a nativa, es decir el empaquetamiento, produce que el código JavaScript sea compilado y transformado. De esta manera es necesario un motor JavaScript en los dispositivos móviles para que sea ejecutado. La ventaja es que esto produce una respuesta más rápida en comparación con PhoneGap.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. V. I. Características

Para hacerse una idea de las características que tiene comparandolo con otros frameworks se detallan sus principales ventajas y desventajas.:

Ventajas

- Multiplataforma móvil y también de escritorio.
- Aspecto y controles nativos. Rendimiento muy rápido.
- Muchos ejemplos y muy buenos para la programacion
- Es gratuito

Desventajas:

- Requiere Mac y Xcode para empaquetar aplicaciones IOS.
- Definición de componentes visuales y controles manualmente (al contrario que PhoneGap, que es HTML)
- Cada aplicación se debe ejecutar con el sistema operativo con el que ha sido desarrollada, es decir una aplicación creada para iPhone no funciona para Android, y viceversa.
- Escasa compatibilidad con los sistemas operativos

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. V. II. Compatibilidad

En Titanium Appcelerator algo que se desarrolla para Iphone no funciona para Android, y viceversa. Los desarrollos de las librerías Javascript evolucionan por separado y hay que mirar bien qué se puede hacer y cómo. A diferencia de PhoneGap, que solo tiene una librería Javascript para acceder a las características especiales del sistema, Appcelerator necesita además librerías para manejar los controles nativos y su disposición en la pantalla, por lo que el desarrollo en general es más costoso.

Sobre el soporte para Iphone/Ipad, es necesario tener Mac con Xcode 4 instalado, el entorno de desarrollo oficial de Apple para aplicaciones Mac y IOS. En cambio para el soporte Android, tanto para probar en el simulador como para empaquetar la aplicación, solo hay que tener el SDK de Android.

Todo esto hace que su compatibilidad comparado con otros frameworks sea bastante mala en cuanto a dispositivos móviles, a pesar de cubrir iOS y Android, que ocupan la mayoría del mercado. El hecho de que algo desarrollado para iOS no sea compatible con Android y viceversa le hace inferior sobre otros frameworks en cuanto a aplicaciones móviles.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Para finalizar con la comparación, se expone una tabla con las diferencias entre PhoneGap y Titanium Appcelerator como se puede observar el la *figura 21:*

	Appcelerator			PhoneGap		
	Android	IOS	Win/Mac/Linux	Android	IOS	BlackBerry
Requiere Mac/Xcode	-	Si	-	-	Si	-
Lenguaje	Javascript			Javascript		
Layout	-			HTML5/CSS3		
Componentes	Nativos		HTML + framework externos	HTML + frameworks externos como Sencha Touch, jQuery Mobile, etc		
IDE	Titanium Studio (basado en Eclipse)			Eclipse	Xcode	No (BB SDK + Ant, JDK)
Rendimiento	Muy bueno			Medio (corre en un navegador)		
Comunidad, documentación, fiabilidad	Regular			Regular		
Precio	Gratis			Gratis		
Motor físico, soporte para juegos	No			No		
Se distribuye con el código fuente	No		Si	Si		
Compatbilidad	Todos los dispositivos			Si		

Figura 21: Comparativa Titanium Appcelerator - PhoneGap

La conclusión de esta tabla será la elección de uno o otro dependiendo del objetivo de la aplicación:

- Publicarla en el App Store/Android Market con un precio por descarga (aplicación de pago): No se debe usar PhoneGap porque podrá ser "pirateada" con extrema facilidad,

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

al incluir el código fuente. Se debería de elegir Appcelerator.

- Publicarla en el App Store/Android Market como aplicación gratuita: Se puede usar PhoneGap, siempre que al usuario no le importe que el mundo entero vea su código. Con Appcelerator no se tendría el problema del código.

- Aplicación a medida: Se puede usar PhoneGap, el cliente quiere que se desarrolle una aplicación a medida, no robar el código. En este caso no seria ningun problema la elección entre PhoneGap y Appcelerator, siempre teniendo en cuenta las limitaciones de cada uno.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VII. VI. Justificación de elección del framework

Una vez vistos y estudiados detalladamente cada uno de los framework que había al alcance para crear y desarrollar la aplicación móvil, se ha llegado a la conclusión de que dos de los estudiados no nos sirven para la creación de la aplicación web. Estos dos frameworks son PhoneGap y Titanium Appcelerator, que como se ha explicado en el estudio de los frameworks, sirven para empaquetar aplicaciones web y convertirlas en nativas, cosa que para el proyecto a desarrollar no es aplicable. Por lo tanto la elección del framework debe de ser entre jQuery Mobile, Sencha Touch y LungoJS.

Estudiando los tres frameworks se ha visto que LungoJS es prácticamente igual que jQuery Mobile, por lo tanto primero se elegirá el mejor de estos dos para seguidamente compararlo con Sencha Touch que es diferente.

Entre jQuery Mobile y LungoJS se llega a la conclusión de que el segundo es prácticamente una copia del primero, y además, por el hecho de que el segundo haya salido hace muy poco, lo hace estar poco desarrollado aún. Además el código de LungoJS es algo más complejo en cuanto a JavaScript, y la compatibilidad respecto a los sistemas operativos lo cubre en menor manera que jQuery Mobile. Por lo tanto entre estos dos framework el elegido es jQuery Mobile.

Una vez descartado LungoJS, quedan jQuery Mobile y Sencha Touch como los dos frameworks más importantes para la creación de la aplicación web. Observando las características de cada uno se llega a la conclusión de que ninguno de los dos es superior al otro, ya que los 2 cubren todo lo deseado para la creación del proyecto. Es destacable que el procedimiento para hacer la web app sea más laborioso con Sencha Touch, ya que tiene mucha más programación con JavaScript, lenguaje mucho más costoso, aunque como con todo lenguaje, una vez estudiado y aprendido, no supondría un problema su desarrollo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Una vez dicho todo esto, lo que ha desequilibrado la decisión entre los dos, además de que sea más costoso el desarrollo con Sencha Touch, ha sido la compatibilidad de cada uno en cuanto a los sistemas operativos de los dispositivos móviles. Como se ha mencionado y explicado en el estudio de cada uno de los frameworks, jQuery Mobile soporta todas las plataformas que actualmente hay en el mercado, mientras que Sencha Touch solo se ha centrado en las más importantes, siendo únicamente compatible con Iphone (todos sus productos), Android (versiones desde 2.3 hacia adelante) y algunos dispositivos de BlackBerry (un grupo muy reducido).

Partiendo de que la elección de Sencha Touch como framework final no hubiera sido una mala decisión, ni mucho menos, se ha creído que jQuery Mobile es algo superior para el proyecto a desarrollar. El argumento principal para la elección de jQuery Mobile es la compatibilidad, apoyado por el desarrollo menos costoso.

Elección final: jQuery Mobile.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

VIII. Tecnologías

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Una vez explicada detalladamente toda la teoría necesaria para el entendimiento del proyecto a desarrollar, y desarrollado el estudio previo a la elección del framework, se explicará brevemente cada una de las tecnologías utilizadas una por una, para que luego en el desarrollo se detalle donde se ha utilizado. Las tecnologías se nombren en el siguiente listado:

- HTML5
- jQuery Mobile
- CSS3
- JavaScript
- Theme Roller
- Google Maps
- Photowise
- Dreamweaver
- Ajax
- Chrome inspector
- XAMPP

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

HTML5:

Es la quinta versión del lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. Esta explicado detalladamente en los conceptos iniciales de la memoria (apartado IV).

jQuery Mobile:

jQuery Mobile es un framework para desarrollo de aplicaciones y sitios web optimizados para smartphones y tablets. Se puede desarrollar una sola aplicación que funcione en las plataformas más populares de smartphones y tablets, en vez de tener que escribir aplicaciones nativas para cada dispositivo móvil o sistema operativo, siempre teniendo en cuenta las limitaciones del navegador utilizado. En el estudio de los frameworks esta explicada la tecnología ampliamente (apartado VII.I)

CSS3:

Mediante este lenguaje definimos la forma, tamaño, colores, etc de cada una de las partes de la aplicación web. Es imprescindible la utilización de CSS si se quiere crear algún elemento vistoso o agradable a la vista. Además mejora considerablemente la usabilidad de la aplicación web porque se cargue más rápido al tener el mismo estilo todos los

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

documentos. Se ha explicado detalladamente en el apartado V.

JavaScript:

Javascript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Tiene la ventaja de ser incorporado en cualquier página web y puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado. Se puede decir que Javascript es un lenguaje interpretado, basado en prototipos.

Theme Roller:

ThemeRoller es una herramienta que encontramos en jQuery UI y que permite ajustar y definir colores, tipografías, etc. de los componentes para interfaz o widgets que ofrece esta biblioteca. Una vez creado el tema deseado, proporciona un archivo (nombre.css o nombre.min.css) que al insertarlo en el código de la aplicación proporciona ese tema a las páginas dejándolo vistoso para el usuario.

Google Maps:

Es un servidor de aplicaciones de mapas en la Web. Ofrece imágenes de mapas desplazables, así como satelitales del mundo e incluso la ruta entre diferentes ubicaciones. En el proyecto se utilizará para crear una apartado útil a la vez que vistoso, el que más adelante se explicará.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Photowise:

PhotoSwipe, otra tecnología importante, está desarrollada con HTML, CSS y Javascript. Esta galería soporta los gestos comunes en dispositivos táctiles, y también detecta la orientación del dispositivo. Funciona en la mayoría de los navegadores móviles y de escritorio, e escala las imágenes automáticamente cuando el dispositivo cambia de posición

Dreamweaver:

Es uno de los programas más comunes en el sector del diseño y la programación web por sus funcionalidades. Permite crear sitios de forma totalmente gráfica, y dispone de funciones para acceder al código HTML generado. Además, para que los usuarios puedan acceder a la aplicación web, Dreamweaver permite la conexión a un servidor, PHP, Javascript, cliente FTP integrado, etc.

Ajax:

Ajax, es una técnica de desarrollo web para crear aplicaciones interactivas . Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. En el proyecto no ha sido prácticamente utilizado pero parece necesaria su explicación ya que puede ser usada en líneas futuras.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Chrome inspector:

El inspector web de Google Chrome es una herramienta para encontrar posibles errores o avisos que vamos a utilizar a la hora de desarrollar nuestra aplicación web. Es un recurso muy importante debido a que informa de los fallos y lo más importante, en qué lugar está el mencionado fallo. Además, enseña la estructura del árbol DOM, concepto explicado anteriormente. Aunque sea muy importante, hay que mencionar que sólo funciona con Google Chrome, lo que hace que todo el test del desarrollo de la página lo hacemos con dicho navegador.

XAMPP:

XAMPP es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. Una de las ventajas de XAMPP es que de una forma muy sencilla y rápida se puede crear en el ordenador un servidor local e independiente para testear la aplicación web antes que esta sea subida a un host en la red.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. Desarrollo de App Web

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En este apartado del desarrollo de la aplicación web se explica detalladamente como se ha ido creando el proyecto, explicando una por una las funcionalidades que se han introducido mediante las tecnologías anteriormente comentadas, junto a los problemas que estas han provocado en su creación.

El primer paso para la explicación del desarrollo consiste en la estructuración de la aplicación web, es decir, hacerse una idea de como va a estar organizada la aplicación. Después se explicarán y se mostrarán mediante ejemplos las diferentes funcionalidades de la aplicación, explicando las características de cada una estas. Seguidamente se detallará la utilización de otras tecnologías utilizadas pero que no son visibles en la aplicación final, aunque sin ellas hubiese sido complicado la creación de dicha aplicación, como lo son por ejemplo Dreamweaver o los servidores locales y web.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. I. Estructura de la Aplicación Web

IX. I. I. Estructura organizativa

El primer paso para un correcto desarrollo consiste en la estructuración de la aplicación web, es decir, hacerse una idea de como va a estar organizada la página. Para ello, basándose en las necesidades y organización del Club Deportivo Ibararte, se decidió, después de varias plantillas e ideas, la creación de 7 apartados principales, siendo 6 de ellos la base de la aplicación junto a una página de inicio que lleva a todas las demás.

Dejando a un lado la página de inicio, se hará una pequeña introducción de cada apartado de la aplicación web uno por uno:

- Club: Es la descripción de cómo se creó el Club Deportivo Ibararte basándose en la unión de dos escuelas deportivas, así como la distribución de sus integrantes y el objetivo del club.

- Ubicación: El objetivo de este apartado es conseguir, mediante la utilización de dispositivos móviles, llevar a la persona que lo desee desde el lugar en el que se encuentra hasta las instalaciones del club. Para ello se debe permitir que la página encuentre la ubicación del dispositivo y, con la utilización de Google Maps, indicara el camino hacia las instalaciones mediante un mapa y explicaciones de cada paso a seguir.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Fútbol Sala: Está formado por 6 subapartados que representan cada categoría y equipo del club. Los apartados son: Pre-Benjamín, Benjamín, Infantil B, Infantil A, Senior y Senior Especial. Cada subapartado tendrá calendario, señalamientos y clasificación del equipo, la lista de los integrantes y una foto de estos.

- Voleibol: Al igual que en fútbol sala, estará formado por 6 subapartados de los diferentes equipos que son los siguientes: Pre-Benjamín, Benjamín,Alevín, Infantil, Cadete-Juvenil y Selección Navarra. Cada subapartado tendrá los datos de competición, la lista de las integrantes y una foto de estas.

- Galería: Se basa en una lista de imágenes del club en formato reducido, que al seleccionar una de ellas aparece a pantalla completa para una mejor visión de estas. En el formato de pantalla completa tendremos la opción de ir visionando las fotos una por una simplemente con desplazarlas con un dedo.

- Patrocinadores: Es un listado de todos los que patrocinan al club, como gesto de agradecimiento, porque gracias a ellos sería improbable que el club pudiese seguir adelante. Cada patrocinador contiene, aparte de ser nombrado, una foto de su logo y un enlace a su página web, siempre que tengan esto último.

El esquema de la organización de la estructura se muestra en la *figura 22*, teniendo en cuenta que en cada página hay una barra de navegación para poder ir en todo momento de una página a otra:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

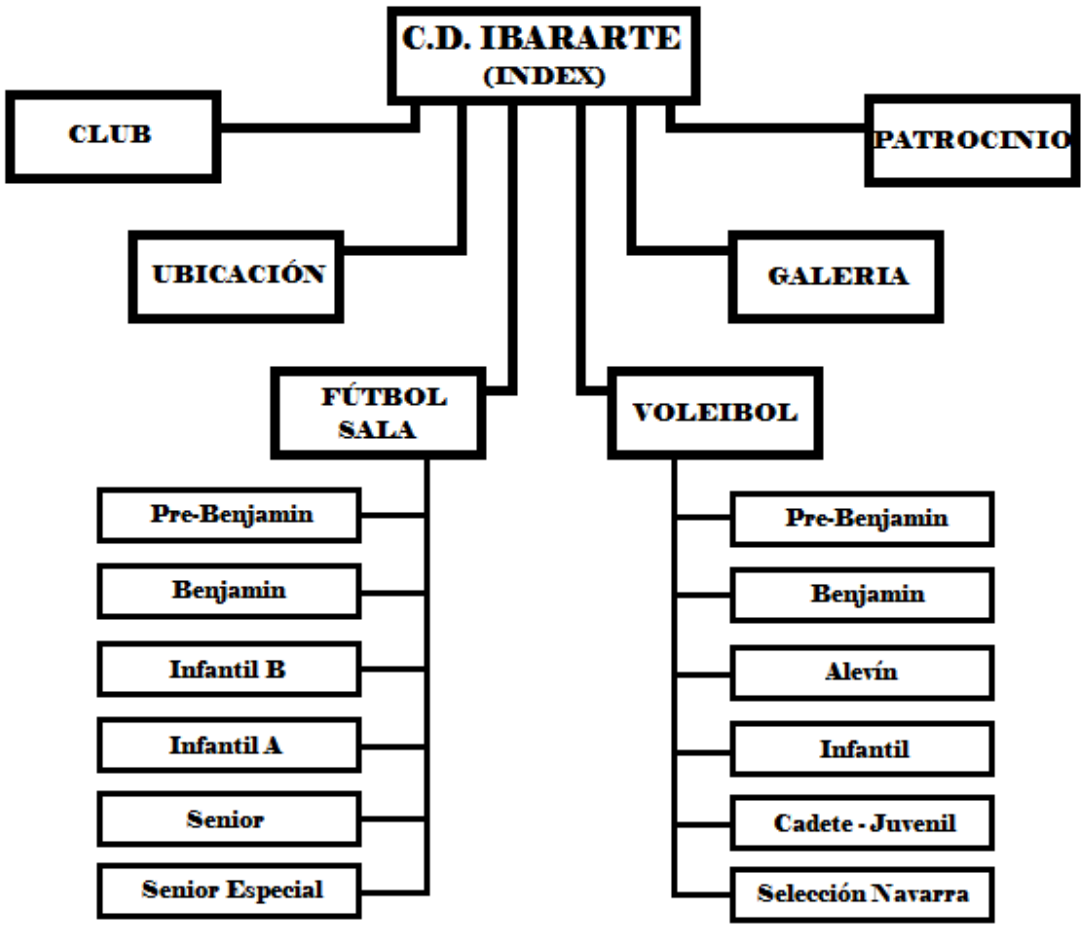


Figura 22: Esquema organización de la aplicación web

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. I. II. Estructura página base

A la hora de iniciar una aplicación web lo primero que se debe de realizar es la estructura de una página base, para luego a partir de esa cambiar el contenido de las demas y asi conseguir que la estructura de toda la aplicación web sea similar. Esta estructura de la página base que se basará resumidamente en una cabecera, una barra de navegación y un pie de página, dejando el contenido a la elección de cada página. Los colores de la aplicación están basados en el rojo y el negro del Club Deportivo Ibararte.

La cabecera estará completada por el nombre del C.D.Ibararte junto a los dos escudos que los miembros de este club llevan a cada lado de la camiseta con la que juegan los campeonatos, cuyos escudos son los de los valles donde comenzó su proyecto deportivo.

La barra de navegación está formada por 6 pestañas, las cuales llevan a cada apartado de la aplicación. Estas barras están presentes en todas las paginas, consiguiendo asi una facil navegacion de unas paginas a otras. La distribución de las pestañas es tres encima de otras tres como se ve en la *figura 23*, ya que jQuery Mobile no permite poner más de 5 en una sola fila, además de que quedarían muy apretadas y serían muy pequeñas.

El pie de página incluye dos botones, uno de ellos lleva a la página inicial y el otro retrocede a la página en de la que llegas. La página inicial es la única que no tiene estos botones ya que no tendrían utilidad

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Todo esta estructura base se puede ver en la *figura 23* en la cual el contenido cambiará, pero la base siempre será la misma:

Figura 23: Ejemplo de estructura base de una página

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. II. Funcionalidades

Primero se explicará brevemente las funcionalidades más importantes de la aplicación web, para luego explicarlas una por una detalladamente para su mejor entendimiento.

- Menús y navegación: Se verá la utilización de distintos recursos de jQuery Mobile y HTML5/ CSS3 escogidos para las páginas que componen la aplicación web.

- Transiciones: Al cambiar de una página a otra crea una transición vistosa y agradable para el usuario

- Localización: Un importante fichero JavaScript que permite la geolocalización y da instrucciones sobre cómo llegar desde el punto en el que se encuentra el usuario hasta el lugar indicado .

- Multimedia:
    - Imagenes: Desarrollado mediante Photowise, una tecnología que está adaptada a los dispositivos móviles (independiente de jQuery Mobile y de HTML5 pero que se realiza con JavaScript).
    - Vídeos: Introducción de videos mediante Youtube.

- Detector de plataforma: script por el cual detecta si se ha entrado la aplicación por móvil, no utilizado directamente pero importante para líneas futuras.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. II. I. Menús y navegación

Las páginas han sido creadas mediante HTML5, con la ayuda del framework jQuery Mobile, y con la utilización de los CSS para conseguir un diseño elegante y funcional.

Estos CSS se pueden encontrar de dos maneras al crearlos mediante ThemeRoller, normal (.css) y minimizado (.min.css), siendo exactamente iguales en su contenido, aunque el minimizado suprime los saltos de línea, espacios en blanco y comentarios para ocupar menos, lo que supone una carga más rápida de la web app pero tiene el inconveniente de ver el código mucho más desordenado y difícil de modificar manualmente.

En la *figura 24* se puede ver el tema creado mediante ThemeRoller:

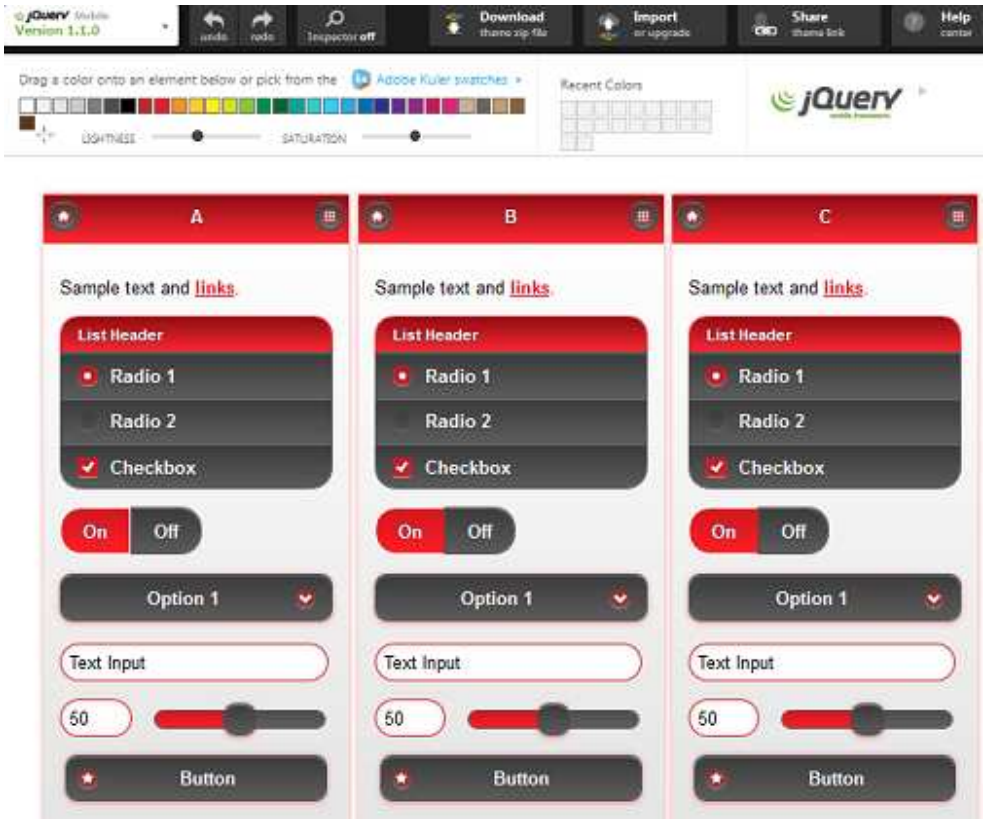
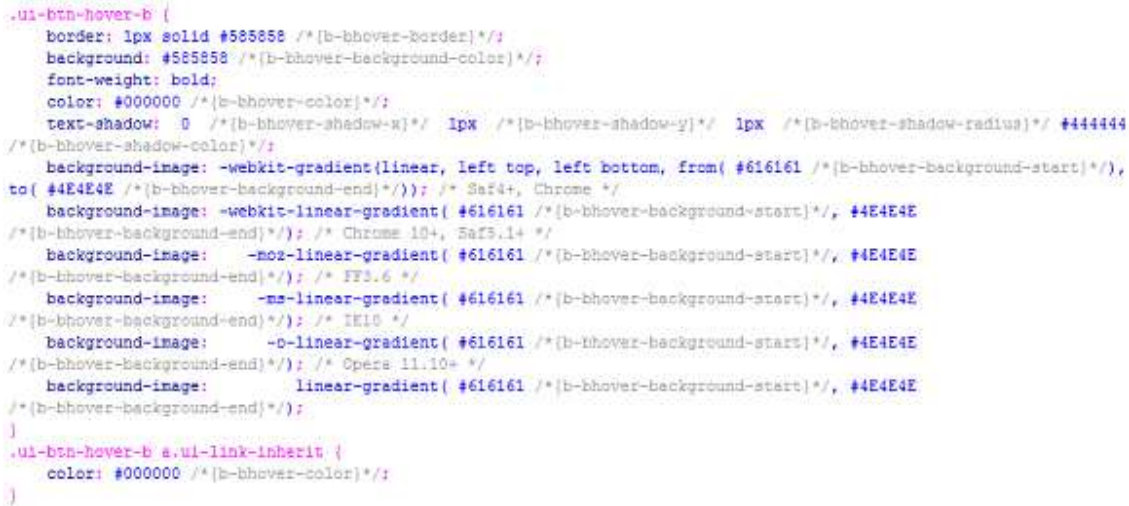


Figura 24: Tema del proyecto mediante ThemeRoller

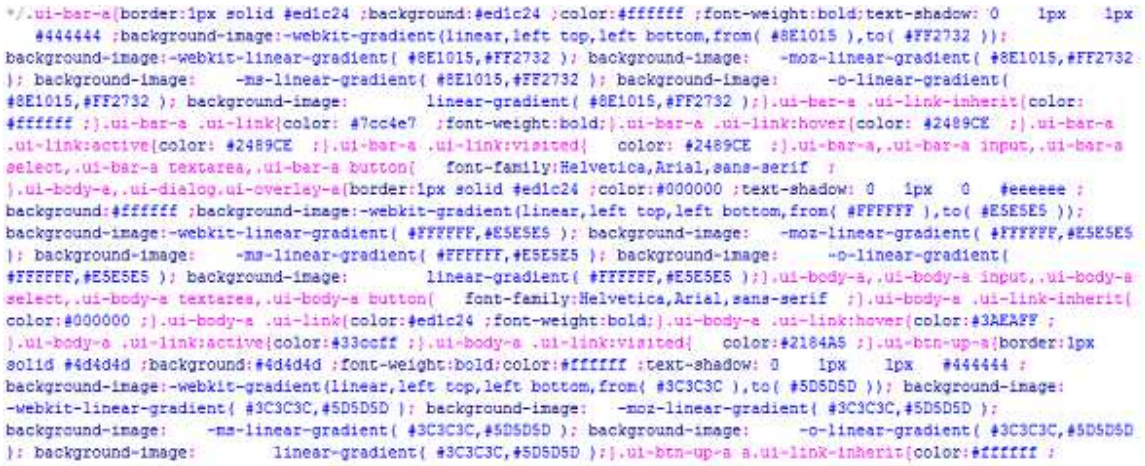
upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Este tema creado, como se ha explicado anteriormente, crea unos archivos con extensión *.css* (*figura 25*) o con extension *.min.css* (*figura 26*) que se observa su código en las siguientes imágenes:



```
.ui-btn-hover-b {
    border: 1px solid #585858 /*{b-bhover-border}*/;
    background: #585858 /*{b-bhover-background-color}*/;
    font-weight: bold;
    color: #000000 /*{b-bhover-color}*/;
    text-shadow:  0  /*{b-bhover-shadow-x}*/  1px  /*{b-bhover-shadow-y}*/  1px  /*{b-bhover-shadow-radius}*/ #444444
/*{b-bhover-shadow-color}*/;
    background-image: -webkit-gradient(linear, left top, left bottom, from( #616161 /*{b-bhover-background-start}*/),
to( #4E4E4E /*{b-bhover-background-end}*/)); /* Saf4+, Chrome */
    background-image: -webkit-linear-gradient( #616161 /*{b-bhover-background-start}*/, #4E4E4E
/*{b-bhover-background-end}*/); /* Chrome 10+, Saf5.1+ */
    background-image:    -moz-linear-gradient( #616161 /*{b-bhover-background-start}*/, #4E4E4E
/*{b-bhover-background-end}*/); /* FF3.6 */
    background-image:     -ms-linear-gradient( #616161 /*{b-bhover-background-start}*/, #4E4E4E
/*{b-bhover-background-end}*/); /* IE10 */
    background-image:      -o-linear-gradient( #616161 /*{b-bhover-background-start}*/, #4E4E4E
/*{b-bhover-background-end}*/); /* Opera 11.10+ */
    background-image:         linear-gradient( #616161 /*{b-bhover-background-start}*/, #4E4E4E
/*{b-bhover-background-end}*/);
}
.ui-btn-hover-b a.ui-link-inherit {
    color: #000000 /*{b-bhover-color}*/;
}
```

Figura 25: Ejemplo theme.css de la aplicación web



```
*/.ui-bar-a{border:1px solid #ed1c24 ;background:#ed1c24 ;color:#ffffff ;font-weight:bold;text-shadow: 0    1px    1px
   #444444 ;background-image:-webkit-gradient(linear,left top,left bottom,from( #8E1015 ),to( #FF2732 ));
background-image:-webkit-linear-gradient( #8E1015,#FF2732 ); background-image:   -moz-linear-gradient( #8E1015,#FF2732
); background-image:    -ms-linear-gradient( #8E1015,#FF2732 ); background-image:     -o-linear-gradient(
#8E1015,#FF2732 ); background-image:        linear-gradient( #8E1015,#FF2732 );}.ui-bar-a .ui-link-inherit{color:
#ffffff ;}.ui-bar-a .ui-link{color: #7cc4e7  ;font-weight:bold;}.ui-bar-a .ui-link:hover{color: #2489CE  ;}.ui-bar-a
.ui-link:active{color: #2489CE  ;}.ui-bar-a .ui-link:visited{   color: #2489CE  ;}.ui-bar-a,.ui-bar-a input,.ui-bar-a
select,.ui-bar-a textarea,.ui-bar-a button(   font-family:Helvetica,Arial,sans-serif  ;
}.ui-body-a,.ui-dialog.ui-overlay-a{border:1px solid #ed1c24 ;color:#000000 ;text-shadow: 0   1px   0   #eeeeee ;
background:#ffffff ;background-image:-webkit-gradient(linear,left top,left bottom,from( #FFFFFF ),to( #E5E5E5 ));
background-image:-webkit-linear-gradient( #FFFFFF,#E5E5E5 ); background-image:   -moz-linear-gradient( #FFFFFF,#E5E5E5
); background-image:    -ms-linear-gradient( #FFFFFF,#E5E5E5 ); background-image:     -o-linear-gradient(
#FFFFFF,#E5E5E5 ); background-image:        linear-gradient( #FFFFFF,#E5E5E5 );}.ui-body-a,.ui-body-a input,.ui-body-a
select,.ui-body-a textarea,.ui-body-a button(   font-family:Helvetica,Arial,sans-serif  ;}.ui-body-a .ui-link-inherit{
color:#000000 ;}.ui-body-a .ui-link{color:#ed1c24 ;font-weight:bold;}.ui-body-a .ui-link:hover{color:#3AEAFF ;
}.ui-body-a .ui-link:active{color:#33ccff ;}.ui-body-a .ui-link:visited{   color:#2184A5 ;}.ui-btn-up-a{border:1px
solid #4d4d4d ;background:#4d4d4d ;font-weight:bold;color:#ffffff ;text-shadow: 0    1px    1px   #444444 ;
background-image:-webkit-gradient(linear,left top,left bottom,from( #3C3C3C ),to( #5D5D5D )); background-image:
-webkit-linear-gradient( #3C3C3C,#5D5D5D ); background-image:   -moz-linear-gradient( #3C3C3C,#5D5D5D );
background-image:    -ms-linear-gradient( #3C3C3C,#5D5D5D ); background-image:     -o-linear-gradient( #3C3C3C,#5D5D5D
); background-image:        linear-gradient( #3C3C3C,#5D5D5D );}.ui-btn-up-a a.ui-link-inherit{color:#ffffff ;
```

Figura 26: Ejemplo theme.min.css de la aplicación web

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Como se observa en las figuras *25* y *26* los temas se encargan de definir todos los elementos posibles de la aplicación, para así, con poner simplemente en el código HTML en la sección *class* un tipo de clase, tendrá la forma y color predefinido anteriormente mediante ThemeRoller. Esto supone una gran ayuda, y si se desea cambiar algo de los temas siempre se podrá hacer manualmente modificando el código de los archivos *.css*.

Para realizar la explicación de los menús y los navegadores, se detalla cada parte principal de estos poniendo un ejemplo de la aplicación para su mejor entendimiento.

- Cabecera: (<header>..</header>)Se debe poner el atributo data-role="header" para que los CSS se encarguen de darle la forma elegida. También se han colocado dos imágenes, una a cada lado. Se puede observar el código y su resultado gráfico en las *figuras 27 y 28:*

```
  <header data-role="header">
<section class="ui-grid-b" style="margin-bottom:-5px">
      <section class="ui-block-a" > <img src="images/erro.jpg" width="100" height="150" alt="Foto"/>
</section>
      <section class="ui-block-b" >
    <h1 align="center" >C. D. Ibararte</h1>
      </section>
      <section class="ui-block-b" > <img src="images/esteribar.jpg" width="100" height="150" alt=
"Foto" align="right"/> </section>
</section>
  </header>
```

Figura 27: Código cabecera

Figura 28: Resultado en navegador cabecera

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Barra de navegación (<nav>..</nav>): Estas etiquetas hacen posible una lista de enlaces a otras páginas para tener siempre disponible una zona de navegación. Además, dentro de ellas están <li>.. y <a>.. para hacer referencia a las mencionadas páginas. El listado está separado en dos apartados de 3 links cada uno porque jQuery Mobile no permite más de 5 links en la misma fila, además de que de esta manera es mucho más visible y táctil para el usuario. Mediante el código *class="ui-btn-active"* se consigue que esté seleccionado cuando nos encontramos en dicha pestaña. Se muestra el código y su resultado gráfico en las *figuras 29 y 30:*

```
<nav data-role="navbar">
  <ul>
    <li><a href="club.html">Club</a></li>
    <li><a href="ubicacion.html">Ubicación</a></li>
    <li><a href="sala.html"  class="ui-btn-active">Sala</a></li>
  </ul>
  <ul>
    <li><a href="voleibol.html">Voleibol</a></li>
    <li><a href="galeria.html">Galeria</a></li>
    <li><a href="patrocinador.html">Patrocinador</a></li>
  </ul>
</nav>
```

Figura 29: Código barra de navegación

Figura 30: Resultado en navegador de barra de navegación

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- Pie de página (<footer>..</footer>) : Se encuentra al final de la página, se le han insertado dos botones con el objetivo de que uno sea para retroceder a la página anterior y otro para volver a la página de inicio. Mediante el código *<a href="javascript:window.history.back();* se consigue retroceder a la página anterior y con el código ** se redirecciona a la página de inicio. Se muestra el código y su resultado gráfico en las *figuras 31 y 32:*

```
<div data-role="footer"  align="center"  style="padding-top:5px">
<a href="javascript:window.history.back();" data-role="button" data-icon="back">ATRAS</a>
<a href="index.html" data-role="button" data-icon="home">INICIO</a>
</div>
```

Figura 31: Código pie de página

Figura 32: Resultado en navegador de pie de página

- Texto (.. , <h2>..</h2>...) : Estas etiquetas nos sirven para introducir un texto, dependiendo de la etiqueta se consigue un tamaño o otro de letra. Se muestra el código y su resultado gráfico en las *figuras 33 y 34:*

```
<h2><strong>Senior</strong></h2>
<p> Son de la categoria juevenil, pero compiten en division 1 de la liga senior.</p>
<p> En la temporada 2011-2012 quedaron quintos en la liga y segundos en la copa</p>
```

Figura 33: Código de ejemplo de texto

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

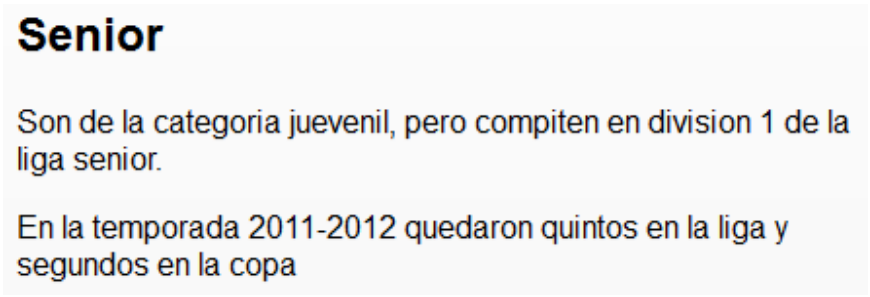


Figura 34: Resultado en navegador de ejemplo de texto

- Botones (data-role="button"): jQuery Mobile ofrece una manera simple de crear botones, a partir de una serie de propiedades o atributos que vienen predeterminados en el framework. Lo más destacable es que, para crear un botón, solo se necesitan pocas líneas de código y con ayuda de algo de JavaScript se convierte en un bonito botón. Se muestra el código y su resultado gráfico en las *figuras 35 y 36*:

```
      <section data-role="content">
    <section class="ui-grid-a">
          <section class="ui-block-a" > <a href="club.html" data-role="button"
data-transition="slide"> <img src="images/escudo.jpg" width="100" height="100" >
            <h3 style="margin:-2px">Club </h3>
            </a> </section>
          <section class="ui-block-b" ><a href="ubicacion.html" data-role="button"
data-transition="flip"> <img src="images/buscar.jpg" width="100" height="100" >
            <h3  style="margin:-2px" >Ubicacion </h3>
            </a></section>
        </section>
    <!-- /grid-a -->

    <section class="ui-grid-a">
          <section class="ui-block-a" ><a href="sala.html" data-role="button"
data-transition="slideup"> <img src="images/balon.jpg" width="100" height="100" >
            <h3 style="margin:-2px" >Futbol Sala </h3>
            </a> </section>
          <section class="ui-block-b" ><a href="voleibol.html" data-role="button"
data-transition="slidedown"> <img src="images/voleibol.jpg" width="100" height="100" >
            <h3 style="margin:-2px" >Voleibol </h3>
            </a></section>
        </section>
```

Figura 35: Código de ejemplo de texto

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Figura 36: Resultado en navegador de botones

- Listas (data-role="listview",
..): Al igual que con los botones, para crear un listado, solo se necesitan pocas líneas de código y con ayuda de algo de JavaScript se vuelve a mostrar unas vistosas listas de lo que se desee. En la aplicación, se ha utilizado para varios objetivos, como lo es por ejemplo el listado de la barra de tareas anteriormente explicado (*figuras 29 y 30*), o en la lista de patrocinadores, con una lista más vistosa incluyendo la foto del patrocinador. Se muestra el código y su resultado gráfico de la lista de patrocinadores en las *figuras 37 y 38*:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
  <section data-role="content">
    <div class="content-primary">
      <ul data-role="listview">
        <li><a href="http://www.magnesitasnavarras.es/"> <img src="images/publi/magna.jpg"
/>
          <h3>Magnesitas Navarra</h3>
          </a></li>
        <li><a href="http://www.erroibar.net/es/"> <img src="images/publi/erro.jpg" />
          <h3>Ayuntamiento de Erro</h3>
          </a></li>
        <li><a href="http://www.esteribar.org/"> <img src="images/publi/esteribar.jpg" />
          <h3>Ayuntamiento de Esteribar</h3>
          </a></li>
        <li><a href="http://www.luzaide-valcarlos.net/"> <img src=
"images/publi/luzaide.jpg" />
          <h3>Ayuntamiento de Luzaide</h3>
          </a></li>
        <li><a href="http://www.burguete.es/es/"> <img src="images/publi/auritz.jpg" />
          <h3>Ayuntamiento de Auritz</h3>
          </a></li>
```

Figura 37: Código de listas de patrocinadores

Figura 38: Resultado en navegador de listas de patrocinadores

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. II. II. Transiciones

Las transiciones son efectos que hacen que al cambiar desde una página a otra, se produzca un determinado efecto diferente según la elección, dándole una elegancia de cara al usuario. jQuery Mobile ,en la versión utilizada para esta aplicación, permite seis diferentes tipos de transiciones, con algo tan sencillo como la utilización del codigo *data-transition="..."* (escribiendo el nombre de la transición entre las comillas). Al utilizar el boton que te lleva a la pagina anterior del pie de pagina se produce la transición inversa. Los efectos que jQuery Mobile realiza son los siguientes:

- Slide : La página actual se desplaza hacia la izquierda a la vez que entra desde la derecha la página nueva. Al no poner ningun tipo de transición realiza esta por defecto.

- Pop : La página nueva se abre sobre la antigua en forma de pop-up pero ocupando todo el espacio.

- Slideup : La página a la que se accede aparece desde abajo, desplazándose hasta ocupar toda la pantalla.

- Slidedown : Igual que Slideup pero apareciendo desde arriba y desplazándose hacia abajo ocupando la pantalla.

- Fade : La pantalla nueva se monta sobre la que está en pantalla con una transición de cambio de opacidad.

- Flip : Esta transición entre pantallas simula un giro de 180º de la página, simulando que la nueva es el dorso de la que teníamos. Es la que más puede sorprender pero también la que más problemas produce debido a muchos dispositivos móviles no tienen la capacidad de realizar el giro rápidamente.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En la *figura 39* se muestra el código para hacer las transiciones, mientras que el la *figura 40* se muestra un ejemplo en 6 pasos de la transición Flip, pasando de la página de inicio a la página Ubicación.

```
    <section data-role="content">
   <section class="ui-grid-a">
         <section class="ui-block-a" > <a href="club.html" data-role="button"
data-transition="slide"> <img src="images/escudo.jpg" width="100" height="100" >
          <h3 style="margin:-2px">Club </h3>
          </a> </section>
         <section class="ui-block-b" ><a href="ubicacion.html" data-role="button"
data-transition="flip"> <img src="images/buscar.jpg" width="100" height="100" >
          <h3  style="margin:-2px" >Ubicacion </h3>
          </a></section>
       </section>
```

Figura 39: Código ejemplo transición Flip

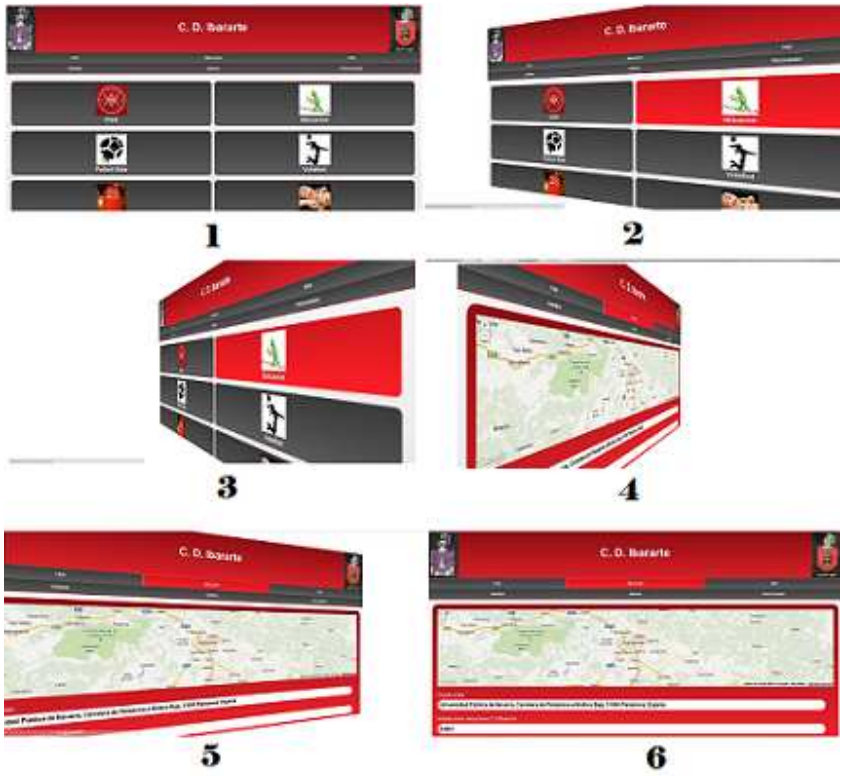


Figura 40: Resultado del ejemplo de transición Flip

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. II. III. Localización

Mediante un script creado por Google es posible que el dispositivo móvil sea localizado en la posición en la que se encuentra, siempre que el usuario acepte ser localizado, y además, dar las instrucciones necesarias sobre cómo llegar al lugar deseado. Esto puede ser muy útil en la aplicación ya que permite a cualquier persona o club, este en el lugar que este, guiarse hasta las instalaciones deportivas del Club Deportivo Ibararte.

Dependiendo del navegador y del dispositivo que disponga el usuario, vamos a poder obtener la información de su ubicación geográfica pero el trabajo de obtenerla lo realizará el navegador, por lo que en ciertos casos podemos tener problemas, ya que, aunque la mayoría soportan HTML5 y jQuery Mobile, hay algunos que dan fallos por problemas de compatibilidad.

Se puede observar en la *figura 41* como la aplicación localiza donde esta el usuario, en este caso concreto la Universidad Pública de Navarra, y mediante el mapa indica el camino a seguir hasta Zubiri, lugar donde se encuentran las instalaciones deportivas del Club. Debajo del mapa encontramos un en botón en el que pone "Te guiamos", haciendo click encima de él, aparecerán las instrucciones de como llegar hasta el lugar deseado como se ve en la *figura 42*.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Figura 41: Ejemplo de geolocalización

Figura 42: Ejemplo de guía a instalaciones desde UPNA

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Para que el servicio que facilita Google Maps sean utilizables en la aplicación, es necesario insertar algunos scripts en el codigo, ademas de descargar los plugins necesarios. Estos plugins de jquery ui son:

- *jquery.ui.map.js :* Muestra el mapa.
- *jquery.ui.map.services.js :* Aplica Google Maps.
- *jquery.ui.map.extensions.js :* Introduce marcadores.
- *demo.js :* Hace que funcionen los tres anteriores.
- *maps.google.com/maps/api/js?sensor=true :* Activa el sensor de geolocalización.

Una vez incluidos estos scripts se debe añadir otro, con un código extenso, que utiliza todos los anteriore para llamar a la función de geolocalización en los distintos navegadores. Se puede observar dicho script en la *figura 43*:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
<script type="text/javascript">
        $(function() {
            var mobileDemo = { 'center': '42.9333,-1.5', 'zoom': 10 };
            $('#directions_map').live('pagecreate', function() {
                demo.add('directions_map', function() {
                    $('#map_canvas_1').gmap({'center': mobileDemo.center, 'zoom':
mobileDemo.zoom,  'zoomControl':true, 'disableDefaultUI':true, 'callback': function()
                        var zoom = $('#map_canvas').gmap('option', 'zoom');
                        var self = this;
                        self.set('getCurrentPosition', function() {
                            self.refresh();
                            self.getCurrentPosition( function(position, status) {
                                if ( status === 'OK' ) {
                                    var latlng = new google.maps.LatLng(position.
coords.latitude, position.coords.longitude)
                                    self.get('map').panTo(latlng);
                                    self.search({ 'location': latlng }, function(
results, status) {
                                        if ( status === 'OK' ) {
                                            $('#from').val(results[0].
formatted_address);
                                        }
                                    });
                                } else {
                                    alert('Unable to get current position');
                                }
                            });
                        });
                        $('#submit').click(function() {
                            self.displayDirections({ 'origin': $('#from').val(),
'destination': $('#to').val(), 'travelMode': google.maps.DirectionsTravelMode.DRIVING }
, { 'panel': document.getElementById('directions')}, function(response, status) {
                                ( status === 'OK' ) ? $('#results').show() : $(
'#results').hide();
                            });
                            return false;
                        });
                    });
                }).load('directions_map');
            });
            $('#directions_map').live('pageshow', function() {
                demo.add('directions_map', $('#map_canvas_1').gmap('get',
'getCurrentPosition')).load('directions_map');
            });
        });
    </script>
```

Figura 43: Script necesario para Google Maps

El único cambio desarrollado en el script, una vez entendido,es el cambio de coordenadas para que encuentre el lugar donde se inicia el mapa en el caso de que no permita el usuario la geolocalización. En el caso de esta aplicación web solo ha hecho falta buscar las cordenadas de Zubiri, y insertarlas en la tercera fila del código, para que el mapa así comienza en el pueblo donde se encuentran las instalaciones deportivas del Club Deportivo Ibararte.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A la hora de crear esta página surgió un problema mediante el cual no se conseguía la visualización del mapa, aunque si que funcionase todos los demás apartados. Esto llevó a varios quebraderos de cabeza ya que todo el código estaba tal y como estaba estipulado en apartado de donde se aprendió a aplicar lo que nos facilita Google Maps. Se miraron en varios foros y múltiples páginas web de internet en busca de información pero no se conseguía corregirlo, hasta que se uso la consola de Google Inspector. Según esta consola el fallo estaba en que no se encontraba un archivo llamado *Modernizr.js,* y buscando dicho archivo en los plugins jquery ui descargados se certificó su ausencia.

En ese momento se decidió descargar otra versión de dichos plugins, ya que la versión Alpha no incluía el archivo *Modernizr.js*, y fue un acierto ya que al descargar la versión Beta se encontraba el archivo deseado. Se cambiaron los plugins y se volvió a probar su funcionamiento, esta vez con una grata sorpresa debido a su perfecto funcionamiento.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. II. IV. Multimedia

El apartado multimedia está basado en una galería fotográfica y la inserción de video en la aplicación web. Para ello, en la galería fotográfica se ha utilizado la tecnología Photowise, ya que después de estudiar las tecnologías disponibles para crear las galerías de fotos de jQuery Mobile, esta es con mucha diferencia las más elegante y funcional, ya que las fotos se pueden arrastrar tactilmente para pasar de una a otra.

Además Photowise consigue tiene la función de que al girar la pantalla del móvil reescala la imagen para la pantalla completa, consiguiendo así que el usuario siempre vea la imagen con el mayor tamaño posible dentro de lo que permite el dispositivo móvil.

En cuanto al código que debemos insertar en nuestro código fuente, son solamente tres scripts, dos de ellos se deben descargarlos y otro más extenso, el cual solo debemos ponerlo en el código de la página inicial. En la *figura 44* se puede observar como se han insertado estos scripts en el código.

```
    <script type="text/javascript" src="plugins/klass.min.js"></script>
    <script type="text/javascript" src="plugins/code.photoswipe.jquery-3.0.4.js">
</script>
    <script type="text/javascript">
        (function(window, $, PhotoSwipe){
            $(document).ready(function(){
                $('div.gallery-page')
                    .live('pageshow', function(e){
                        var
                            currentPage = $(e.target),
                            options = {},
                            photoSwipeInstance = $("ul.gallery a", e.target).photoSwipe
(options,  currentPage.attr('id'));
                        return true;
                    })
                    .live('pagehide', function(e){
                        var
                            currentPage = $(e.target),
                            photoSwipeInstance = PhotoSwipe.getInstance(currentPage.
attr('id'));
                        if (typeof photoSwipeInstance != "undefined" &&
photoSwipeInstance != null) {
                            PhotoSwipe.detatch(photoSwipeInstance);
                        }
                        return true;
                    });
            });
        }(window, window.jQuery, window.Code.PhotoSwipe));
    </script>
```

Figura 44: Código scripts Photowise

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En el apartado de la galería fotográfica nos encontramos con un problema, ya que las fotos que se ven antes de seleccionar alguna de ellas, deberían de salir con el formato que Photowise expone en su código. Despues de mas de una semana estudiando este problema, entrado en foros buscando la opinión de otros desarrolladores y probando prácticamente lo improbable, se ha llegado a la conclusión de que el problema viene del hecho de que HTML5 y jQuery Mobile estan todavia sin acabar de tener una compatibilidad completa. A esta conclusión se ha llegado ya que que probandolo en otra pequeña aplicación creada para intentar solucionar el problema, pero en este caso usando HTML en vez de HTML5, su funcionamiento ha sido correcto, pero al intentar introducir HTML en la aplicación web del proyecto, perdía la mayoría de las virtudes que tiene jQuery Mobile, ya que se supone que esta diseñado para HTML5, aunque se tenga el problema de la galería con Photowise. Una vez seleccionada la imagen el funcionamiento es correcto como se puede ver en la *figura 45*:

Figura 45: Ejemplo Photowise foto

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Para insertar un video en la aplicación web solo tenemos que utilizar un sencillo código como el que se ve en la *figura 46*, ya que aprovechándose de Youtube, podemos insertar todos los videos deseados siempre que vengan de esta plataforma. Como se ve en la figura 46, las etiquetas para la inserción de video son <iframe>..</iframe>, asi HTML toma las dirección que tenga el video en la página web de donde proviene.

```
<iframe class="youtube-player" content="width=device-width, initial-scale=1"
src="http://www.youtube.com/embed/4Ga6zw7AR6k" frameborder="0">
        <param name="allowFullScreen" value="true">
        </iframe>
```

Figura 46: Ejemplo de inserción de video

En la *figura 47* se observa como queda el video en la aplicación:

Figura 47: Video de la aplicación web

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. II. V. Detección de dispositivos móviles

Para que detecte cuando un usuario entra desde un dispositivo móvil o un ordenador es necesario un script adicional a los anteriormente explicados. Este script hará que, al acceder a la página, detecte si se está accediendo mediante uno de los tres diferentes sistemas operativos que aparecen en la *figura 48*, y en ese caso, se redireccionará directamente a la página que se puede observar en "document.location", que aunque actualmente no sea útil ya que no tiene el C.D. Ibararte página web oficial, será una de las líneas futuras a mencionas, por lo que ya estará preparado para el momento en que se haga oficial la página web del club. En la *figura 48* se observa el script:

```
<script type="text/javascript">
var navegador = navigator.userAgent.toLowerCase();
if( navegador.search(/iphone|ipod|android/) > -1 ){
document.location ="http://ibararte.freeiz.com";
}
</script>
```

Figura 48: Script para detección de móvil

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. III. Aplicación de otras tecnologías

IX. III. I. Servidor local y web para test

En el apartado de tecnologías del uso se ha mencionado una herramienta para crear un servidor local para testear la aplicación web. Como antes se ha mencionado, XAMPP es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. Una de las ventajas de XAMPP es que de una forma muy sencilla y rápida se puede crear en el ordenador un servidor local e independiente para testear la aplicación web antes que esta sea subida a un host en la red.

Una vez iniciado y situado en el panel de control del programa, lo único que tenemos que hacer es ejecutar los componentes Apache y MySql, como se muestra en la figura 49, además de introducir en la carpeta *"htdocs"* todos los archivos de la aplicación web, para que se puedan ejecutar como servidor local.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

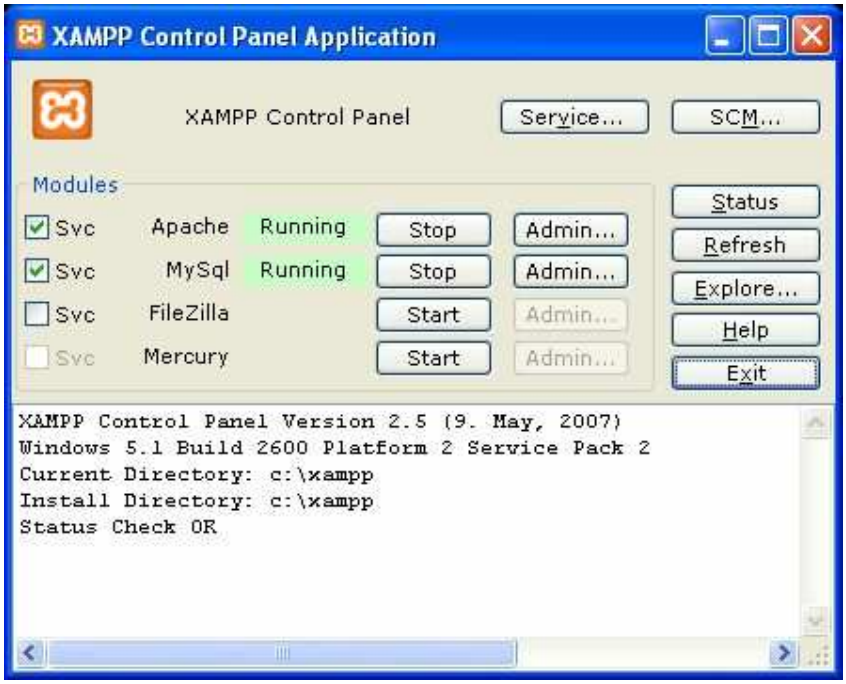


Figura 49: Panel de control XAMPP

Para poder acceder al servidor local, siempre teniendo XAMPP en funcionamiento y todos los archivos en la carpeta *"htdocs"* del servidor, debemos de escribir en el navegador lo siguiente: *"localhost/nombrecarpetaproyecto"* .

Una vez hecho todo lo anterior solo queda probar su funcionamiento desde el servidor local, lo que permite comprobar su funcionamiento en la totalidad, cosa que sin subirlo al servidor local no permitía. Un ejemplo de lo nombrado son las transiciones, ya que sin testarlo con el servidor local no se visualizaban correctamente, y desde el servidor local si.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Para subir la pagina a un servidor web, lo primero que se necesita es un dominio y un host para alojar la web. Para ello se ha decidido hacerlo mediante *www.000webhost.com* ya que es totalmente libre de publicidad y no cambia el codigo de la pagina en absoluto, ademas permite 1500 Mb de espacio y se pueden crear 5 páginas adicionales y 5 subdominios, tambien tiene soporte php y mysql. Se puede crear la página web poniendo el nombre que quieras, aunque te añaden otro nombre del servidor gratuito que lo facilita, en este caso Freeiz, dejando la direccion de la aplicación web en: *http://www.ibararte.freeiz.com*

Una vez obtenido el dominio y el host, habrá que subir la aplicación a la red mediante Dreamweaver, ya que es uno de los programas que lo permiten como se puede observar en la *figura 50*. Se explicará algo mas detallado el uso de este programa en el siguiente apartado.

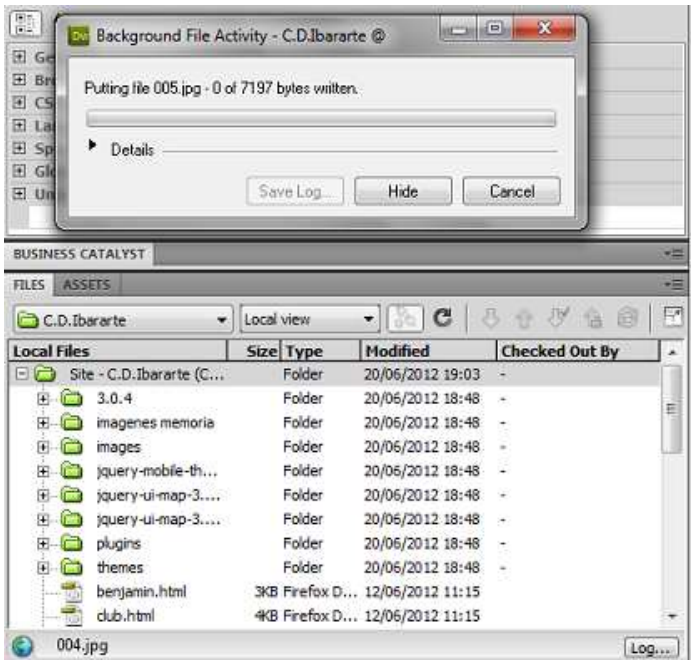


Figura 50: Subir aplicacion web con Dreamweaver

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. III. II. Dreamweaver

Como se ha explicado en el apartado de las tecnologías Dreamweaver es uno de los programas más comunes en el sector del diseño y la programación web por sus funcionalidades. Permite crear sitios de forma totalmente gráfica, y dispone de funciones para acceder al código HTML generado.

Según los expertos en el tema Dreamweaver es la herramienta de diseño de páginas web más avanzada. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrarán en este programa razones para utilizarlo, sobre todo en lo que a productividad se refiere.

Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas
- Javascript para crear efectos e interactividades
- Inserción de archivos multimedia...

La única pega consiste en que al ser tan avanzado, puede resultar un poco difícil su manejo para personas menos experimentadas en el diseño de webs.

En la *Figura 51* se puede observar como la imagen de la pantalla principal de Dreamweaver, donde a la izquierda se tiene el código de la aplicación a desarrollar, en el centro la vista de la aplicación que se esta creando (para poder hacer cambios y observarlos al instante), y a la derecha menús donde se puede modificar la aplicación.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Figura 51: Ejemplo programa Dreamweaver

El menú de abajo a la derecha de la *figura 51*, con el nombre FILES, es el encargado de subir la aplicación a un servidor web como se ha explicado en el apartado anterior. Mediante el primer icono (dos enchufes) se conecta al servidor, mientras que mediante la flecha hacia arriba se sube la aplicacion al servidor web. Si se cambia la opcion de *Local view* por *Remote server*, todos los cambios que se vayan haciendo se irán cambiando simultáneamente en la aplicación colgada en la red, opción que no interesa actualmente ya que es mejor hacer los cambios y una vez finalizado volver a subir la aplicación. Estos detalles se observan mejor en la *figura 52:*

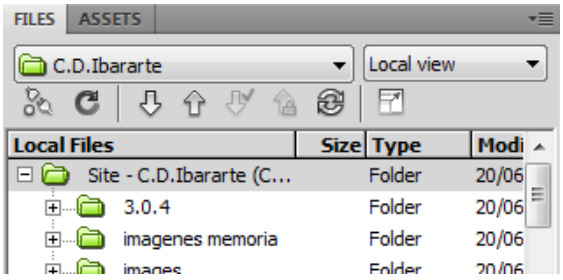


Figura 52: Opciones Files de Dreamweaver

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

IX. III. III. Chrome Inspector

Chrome Inspector fue creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar (revisar velocidad de carga, estructura DOM), editar, monitorizar y depurar el código fuente, CSS, HTML y JavaScript de una página web de manera instantánea y online.

No es un simple inspector como DOM Inspector, además edita y permite guardar los cambios, su atractiva e intuitiva interfaz, con solapas específicas para el análisis de cada tipo de elemento (consola, HTML, CSS, Script, DOM y red), permite al usuario un manejo fácil y rápido.

En el proyecto de la aplicación web, la función que más se ha utilizado a sido la consola, para encontrar posibles errores o avisos en el desarrollo. Es un recurso muy importante debido a que informa de los fallos y lo más importante, en qué lugar está esos fallo, así ahorrando varios horas de búsqueda de fallos a los desarrolladores. Una de las desventajas que tiene es que sólo funciona con Google Chrome, lo que hace que todo el test del desarrollo de la página se haga con dicho navegador. En la *Figura 53* se puede observar un ejemplo de la consola de Chrome Inspector.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Figura 53: Ejemplo de Chrome Inspector

En la *figura 53* se puede observar como no encuentra la imagen de patrocinadores, y mediante la consola del elemento de inspección de Google Chrome se ve como encuentra el fallo y nos indica que no puede cargar la imagen al no estar esa imagen en el directorio seleccionado. Por lo tanto se tendrá que cambiar el directorio o buscar la imagen *publicidad.jpg* y dejarla en la carpeta *images*.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

X. Conclusiones

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- La conclusión más importante durante todo este proceso es la inestabilidad de jQuery Mobile al utilizar codigo HTML5. Durante el desarrollo de esta aplicación web se han sufrido varios problemas de compenetración entre el código HTML5 y jQuery Mobile, ya que por ejemplo, al utilizar la etiqueta <section>, nueva etiqueta de HTML5, no reconoce parte del código de jQuery Mobile que si reconoce con la etiqueta . En un intento de arreglarlo dicho problema con la utilización de la etiqueta de HTML en partes del proyecto, se pierden la mayoría de los recursos que facilita jQuery Mobile, ya que esta creado para que funcione con lenguaje HTML5, aunque no siempre sea cierto.

- Una de las primeras conclusiones es la sencillez del desarrollo de las aplicaciones web gracias al Framework jQuery Mobile. El uso de scripts y código HTML facilita mucho la creación del proyecto ya que relativamente con poco código se ha creado una aplicación elegante y funcional. Esta conclusión afirma la correcta elección del framework, ya que uno de los argumentos para dicha selección era la sencillez de desarrollo.

- Respecto a la compatibilidad se llega a la conclusión que aunque jQuery Mobil teóricamente sea compatible con todos los sistemas operativos, y en la práctica funcione con todas las plataformas en mayor o menor medida, no acaba de ser del todo cierta esta compatibilidad, ya que en distintos navegadores se encuentran diferencias y fallos en la forma, posición o ejecución.

- El script para detectar el dispositivo móvil con el que se accede a la aplicación tiene el inconveniente de sólo ser válido para Android y dispositivos de la marca Apple, lo cual hace que al acceder desde cualquier sistema operativo tipo Blackberry o Windows Mobile, no servirá de nada dicho script.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

XI. Líneas Futuras

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Las líneas futuras son las mejoras que se le desean introducir a la aplicación web en un futuro, mostrándose a continuación el listado de ellas:

- Debido a la inestabilidad entre HTML5 y jQuery Mobile explicada en las conclusiones, se espera que en relativamente poco tiempo se puedan solucionar los errores que se producen como consecuencia del avance y mejoras del framework jQuery Mobile y del lenguaje HTML5.

- Una línea de futuro próxima y sencilla sería utilizar PhoneGap o Titanium Appcelerator, es decir, uno de los frameworks explicados en el estudio de estos, para la creación de la aplicación nativa a partir de la aplicación web realizada.

- Otra posible mejora de la aplicación sería insertar un formulario en el cual el usuario podría pedir ser parte del club, creando una base de datos que recoja estos datos y los almacene, para así al iniciar cada temporada poder ver las solicitudes de inserción en el club y estudiarlas.

- Una vez creada la página web oficial del Club Deportivo Ibararte, se insertará en ella el código para la detección de dispositivos móviles ya preparado en este proyecto. Así, al entrar en la página oficial desde un dispositivo móvil, detectará que es desde este y accede directamente a la aplicación web, siempre poniendo en esta última la opción de poder ver la versión completa de la página web.

- También se podría crear un código QR para ponerlo en los lugares donde haga referencia al club, y así poder acceder directamente sin necesidad de entrar en el navegador y escribir la dirección de la aplicación web.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

XII. Bibliografía

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- HTML5 : UP AND RUNNING. Mark Pilgrim. O'Reilly, 2010.

- JQUERY REFERENCE GUIDE: A COMPREHENSIVE EXPLORATION OF THE POPULAR JAVASCRIPT LIBRARY. Jonathan Chaffer. Mumbai [India] : Packt Publishing, 2007

- BUSCADOR DE GOOGLE.

- ADOBE.COM/PRODUCTS/DREAMWEAVER. Descarga del programa.

- JQUERYMOBILE.COM . Página oficial de jQuery Mobile

- SENCHA.COM/PRODUCTS/TOUCH . Página oficial de Sencha Touch

- LUNGOJS.COM . Página oficial de LungoJs

- APPCELERATOR.COM . Página oficial de Titanium Appcelerator

- PHONEGAP.COM . Página oficial de PhoneGap

- Manuales HTML5 y JavaScript. Diversos autores

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira