



Audioguía turística del R.M. de San Lorenzo de
El Escorial

Desarrollo de aplicaciones móviles con HTML5

Jesús Villegas Guerra

Ingeniero Técnico en Informática de Gestión



Consultor: Albert Grau Perisé
Curso: 2013-2014-2
Fecha: 07-06-2014



INDICE

1. INTRODUCCIÓN	11
1.1. APP CON HTML5.....	11
1.2. TIPOS DE APPS	12
1.3. VENTAJAS E INCONVENIENTES DE USAR APLICACIONES HÍBRIDAS.....	13
1.4. HARDWARE Y SOFTWARE	15
1.5. TECNOLOGÍAS UTILIZADAS	15
1.6. OBJETIVOS DEL TRABAJO FIN DE CARRERA	16
2. REQUISITOS DE LA APLICACIÓN	17
2.1. DESCRIPCIÓN DEL PROYECTO	17
2.2. REQUISITOS FUNCIONALES MÍNIMOS A DESARROLLAR	18
2.3. REQUISITOS NO FUNCIONALES.....	19
2.4. METODOLOGÍA DCU	20
2.5. RIESGOS DEL PROYECTO	21
3. PLANIFICACIÓN Y EJECUCIÓN DEL PROYECTO	22
3.1. TAREAS PRINCIPALES	22
3.2. PLANIFICACIÓN: DIAGRAMA DE GANTT	23
4. INVESTIGACIÓN	24
4.1. PERFILES DE USUARIO	26
4.2. CONTEXTO DE USO.....	30
4.3. ANÁLISIS DE TAREAS.....	30
4.4. TEST DE USABILIDAD: ESTUDIO DE LA COMPETENCIA.....	31
4.4.1. CONCLUSIONES DEL TEST: DATOS ÚTILES PARA EL DISEÑO DE LA APP.....	38
5. DISEÑO DE ESCENARIOS, FLUJOS DE INTERACCIÓN Y CASOS DE USO	39
5.1. ESCENARIOS.....	39
5.1.1. LEER UN TEXTO SOBRE ALGÚN TEMA CONCRETO DE LA APLICACIÓN.....	39
5.1.2. IR AL INICIO DE LA APLICACIÓN DESDE LA PANTALLA DONDE SE ENCUENTRE	39
5.1.3. ACCEDER AL TEXTO DE PANTEÓN DE LAS INFANTAS	40
5.1.4. ESCUCHAR UNA AUDICIÓN Y PARARLA.	40
5.1.5. BUSCAR EL TELÉFONO DEL MONASTERIO Y LLAMAR.	41
5.2. FLUJOS DE INTERACCIÓN.	41

5.2.1.	LEER UN TEXTO SOBRE ALGÚN TEMA CONCRETO DE LA APLICACIÓN	41
5.2.2.	ACCEDER LOS DISTINTOS LISTADOS DE LA APLICACIÓN.	42
5.2.3.	ESCUCHAR UNA AUDICIÓN.	43
5.3.	CASOS DE USO: ESPECIFICACIÓN TEXTUAL Y DIAGRAMA.....	44
5.3.1.	IR AL INICIO	46
5.3.2.	LEER UN TEXTO	46
5.3.3.	VER UNA IMAGEN	47
5.3.4.	LLAMAR POR TELÉFONO	48
5.3.5.	ACCEDER AL MENÚ.....	49
5.3.6.	UTILIZAR EL CONTROL DE AUDIO	50
5.3.7.	ENTRAR EN UNA PÁGINA.....	51
6.	PROTOTIPO	51
7.	EVALUACIÓN	57
7.1.	CONCLUSIONES A LA EVALUACIÓN.....	60
8.	IMPLEMENTACIÓN	62
8.1.	CREACIÓN DEL PROYECTO	62
8.2.	ESTRUCTURA INICIAL DE LA PÁGINA INDEX.HTML.....	64
8.2.1.	HEAD	64
8.2.2.	BODY : PÁGINA.....	66
8.2.3.	BODY: SCRIPTS.....	67
8.3.	ANATOMÍA DE LA APLICACIÓN REALIZADA EN ECLIPSE.....	67
8.4.	ASPECTO VISUAL: MULTIDISPOSITIVO.	69
8.4.1.	APP.CSS: MEDIA QUERIES (ETIQUETA @MEDIA)	71
8.5.	FUNCIONALIDADES.....	74
8.5.1.	PANTALLA SPLASH-SCREEN	76
8.5.2.	IR A INICIO	78
8.5.3.	LEER UN TEXTO	81
8.5.4.	VER UNA IMAGEN	84
8.5.5.	LLAMAR POR TELÉFONO	85
8.5.6.	ACCEDER AL MENÚ.....	86
8.5.7.	UTILIZAR EL CONTROL DE AUDIO	90
8.5.8.	ENTRAR EN UNA PÁGINA.....	93
8.5.9.	VISUALIZAR GOOGLE-MAPS.....	94
8.5.10.	ACCEDER A PÁGINA EXTERNA.....	99
8.6.	PHONEGAP BUILD: DESARROLLO MULTIPLATAFORMA.....	100
8.7.	GOOGLE PLAY: DISTRIBUCIÓN DE LA APLICACIÓN.	106

9. CONCLUSIONES	107
10. GLOSARIO	109
11. ANEXOS	111
11.1. CÓMO PREPARAR WINDOWS PARA TRABAJAR CON ANDROID+PHONEGAP	111
11.2. COPIA DE LA ENTREVISTA A LOS USUARIOS	115
11.3. HOJA PRESENTADA A LOS USUARIOS: TAREA DEL TEST A LA COMPETENCIA.	116
11.4. HOJA PRESENTADO A LOS USUARIOS: TAREA EVALUACIÓN DEL PROTOTIPO.	116
11.5. EVENTOS DE JQUERY MOBILE	117
11.6. PANTALLAS DE LA APLICACIÓN.	118
11.6.1. PANTALLAS DE INICIO	118
11.6.2. PANTALLAS PRINCIPALES	118
11.6.3. PANTALLAS PANTEONES	119
11.6.4. PANTALLAS DE INFORMACIÓN	119
12. BIBLIOGRAFIA	120

INDICE DE FIGURAS

FIGURA 1: APP CON HTML5.....	11
FIGURA 2: CARACTERÍSTICAS DE APLICACIONES NATIVAS, WEB E HÍBRIDAS.....	13
FIGURA 3: VENTAJAS Y DESVENTAJAS DE LAS APPS HIBRIDAS.....	14
FIGURA 4: TECNOLOGÍAS WEB.....	15
FIGURA 5: MODELO CLÁSICO DE DESARROLLO DE SOFTWARE.....	20
FIGURA 6: MODELO DCU DE DESARROLLO DE SOFTWARE.....	21
FIGURA 7: TABLA DE PLANIFICACIÓN TEMPORAL.....	23
FIGURA 8: PLANIFICACIÓN TEMPORAL. DIAGRAMA DE GANTT.....	24
FIGURA 9: DURACIÓN ESTANCIA EN R.M. EL ESCORIAL EN %.....	26
FIGURA 10: CONSULTAS EN OFICINA INFORMACIÓN TURÍSTICA DE EL ESCORIAL (MESES Y PROCEDENCIA).....	26
FIGURA 11: EDAD MEDIA DEL TURISTA.....	27
FIGURA 12: ACOMPAÑANTE DEL TURISTA.....	27
FIGURA 13: LUGARES VISITADOS EN MADRID EN %.....	28
FIGURA 14: TABLA DE PERFIL USUARIO 1.....	29
FIGURA 15: TABLA DE PERFIL DE USUARIO 2.....	29
FIGURA 16: TABLA DE PERFIL DE USUARIO 3.....	30
FIGURA 17: TABLA USUARIO 1. TEST USABILIDAD COMPETENCIA APP “EL ESCORIAL”.....	32
FIGURA 18: TABLA USUARIO 2. TEST USABILIDAD COMPETENCIA APP “EL ESCORIAL”.....	33
FIGURA 19: TABLA USUARIO 3. TEST USABILIDAD COMPETENCIA APP “EL ESCORIAL”.....	33
FIGURA 20: TABLA USUARIO 1. TEST USABILIDAD COMPETENCIA APP “NATIONAL MUSEUM OF KOREA”.....	36
FIGURA 21: TABLA USUARIO 2. TEST USABILIDAD COMPETENCIA APP “NATIONAL MUSEUM OF KOREA”.....	36
FIGURA 22: TABLA USUARIO 2. TEST USABILIDAD COMPETENCIA APP “NATIONAL MUSEUM OF KOREA”.....	37
FIGURA 23: FLUJO DE INTERACCIÓN. DIAGRAMA LEER UN TEXTO.....	42
FIGURA 24: FLUJO DE INTERACCIÓN. ACCEDER A LISTADOS.....	43
FIGURA 25: FLUJO DE INTERACCIÓN. ESCUCHAR AUDICIÓN.....	44
FIGURA 26: DIAGRAMA: CASOS DE USO.....	45
FIGURA 27: TABLA CASO DE USO. IR A INICIO.....	46
FIGURA 28: TABLA CASO DE USO. LEER UN TEXTO.....	47
FIGURA 29: TABLA CASO DE USO. VER UNA IMAGEN.....	48
FIGURA 30: CASO DE USO. LLAMAR POR TELÉFONO.....	49
FIGURA 31: CASO DE USO. ACCEDER AL MENÚ.....	50
FIGURA 32: CASO DE USO. UTILIZAR CONTROL DE AUDIO.....	50
FIGURA 33: CASO DE USO. ENTRAR EN UNA PÁGINA.....	51
FIGURA 34: PROTOTIPO. MAPA DE LA APLICACIÓN. AXURE RP.....	52
FIGURA 35: PROTOTIPO. PÁGINA INICIO.....	53
FIGURA 36: PROTOTIPO. PULSAR “COMENZAR”.POP UP.....	54
FIGURA 37: PROTOTIPO. PANTEONES.....	54
FIGURA 38: PROTOTIPO. MÁS INFORMACIÓN.....	55
FIGURA 39: PROTOTIPO. BIOGRAFÍAS.....	55

Alumno: Jesús Villegas Guerra

FIGURA 40: PROTOTIPO. MISTERIOS.....	56
FIGURA 41: PROTOTIPO. PLANOS.....	56
FIGURA 42: PROTOTIPO. LLAMAR POR TELÉFONO.....	57
FIGURA 43: TABLA USUARIO 1. EVALUACIÓN PROTOTIPO.....	58
FIGURA 44: TABLA USUARIO 2. EVALUACIÓN PROTOTIPO.....	59
FIGURA 45: TABLA USUARIO 3. EVALUACIÓN PROTOTIPO.....	60
FIGURA 46: CREACIÓN PROYECTO. CARPETA BIN PHONEGAP.....	62
FIGURA 47: CREACIÓN PROYECTO. PROYECTO CREADO.....	62
FIGURA 48: CREACIÓN PROYECTO. ECLIPSE.....	63
FIGURA 49: CREACIÓN PROYECTO. CARPETA "ASSETS\WWW".....	63
FIGURA 50: CREACIÓN PROYECTO. ESTRUCTURA INICIAL "INDEX.HTML".....	64
FIGURA 51: ESTRUCTURA INDEX.HTML. DOCTYPE.....	64
FIGURA 52: ESTRUCTURA INDEX.HTML. ETIQUETA VIEWPORT.....	65
FIGURA 53: ESTRUCTURA INDEX.HTML. CSS.....	65
FIGURA 54: ESTRUCTURA INDEX.HTML. BODY-PAGINA JQUERY MOBILE.....	66
FIGURA 55: ANATOMÍA DE LA APLICACIÓN. ANDROID.....	68
FIGURA 56: ASPECTO VISUAL. CSS UTILIZADOS.....	70
FIGURA 57: ASPECTO VISUAL. THEMEROLLER.....	70
FIGURA 58: ASPECTO VISUAL. CSS. MEDIA QUERIE.....	71
FIGURA 59: ASPECTO VISUAL. CSS. MEDIA QUERIE II.....	72
FIGURA 60: ASPECTO VISUAL. RIPPLE. 320x480PX.....	73
FIGURA 61: ASPECTO VISUAL. RIPPLE. 480x800.....	73
FIGURA 62: ASPECTO VISUAL. RIPPLE. TABLET 7".....	74
FIGURA 63: FUNCIONALIDADES. TABLA DE FUNCIONALIDADES.....	76
FIGURA 64: FUNCIONALIDADES: PANTALLA SPLASH-SCREEN.....	77
FIGURA 65: FUNCIONALIDADES: SPLAS-SCREEN. ARCHIVO "SRC".....	77
FIGURA 66: FUNCIONALIDADES. SPLASH-SCREEN. CÓDIGO.....	77
FIGURA 67: FUNCIONALIDADES. SPLASH-SCREEN. ARCHIVO "RES".....	78
FIGURA 68: FUNCIONALIDADES. IR A INICIO.....	79
FIGURA 69: FUNCIONALIDADES. BOTÓN INICIO Y ATRÁS.....	79
FIGURA 70: FUNCIONALIDADES. CÓDIGO BOTÓN IR A INICIO Y ATRÁS.....	80
FIGURA 71: FUNCIONALIDADES. TRANSICIONES. APPCONFIG.JS.....	80
FIGURA 72: FUNCIONALIDADES. LEER UN TEXTO. MENSAJE POP-UP.....	81
FIGURA 73: FUNCIONALIDADES. LEER UN TEXTO. SCRIPT POP-UP.....	82
FIGURA 74: FUNCIONALIDADES. LEER UN TEXTO. PANTALLA.....	83
FIGURA 75: FUNCIONALIDADES. LEER UN TEXTO. CÓDIGO CSS.....	83
FIGURA 76: FUNCIONALIDADES. VER UNA IMAGEN. PANTALLA.....	84
FIGURA 77: FUNCIONALIDADES. VER UNA IMAGEN. CÓDIGO CSS Y MEDIA QUERIE.....	85
FIGURA 78: FUNCIONALIDADES. LLAMAR POR TELÉFONO. CÓDIGO.....	85
FIGURA 79: FUNCIONALIDADES. LLAMAR POR TELÉFONO. PANTALLA.....	86
FIGURA 80: FUNCIONALIDADES. ACCEDER AL MENÚ. PANTALLAS.....	87

FIGURA 81: FUNCIONALIDADES. ACCEDER AL MENÚ. CÓDIGO 1.....	87
FIGURA 82: FUNCIONALIDADES. ACCEDER AL MENÚ. CÓDIGO 2.....	88
FIGURA 83: FUNCIONALIDADES. ACCEDER AL MENÚ. CSS. ICONOS PERSONALIZADOS.....	89
FIGURA 84: FUNCIONALIDADES. ACCEDER AL MENÚ. CSS. QUITAR ICONOS REDONDEADOS.....	89
FIGURA 85: FUNCIONALIDADES. UTILIZAR CONTROL DE AUDIO. PANTALLA.....	90
FIGURA 86: FUNCIONALIDADES. UTILIZAR CONTROL DE AUDIO. CÓDIGO HTML5+ JQUERY MOBILE.....	90
FIGURA 87: FUNCIONALIDADES. UTILIZAR CONTROL DE AUDIO. CÓDIGO PARAR AUDICIÓN.....	91
FIGURA 88: FUNCIONALIDADES. UTILIZAR CONTROL DE AUDIO. EVENTO.....	92
FIGURA 89: FUNCIONALIDADES. UTILIZAR CONTROL DE AUDIO. AUDIO EXTERNO A LA APP.....	92
FIGURA 90: FUNCIONALIDADES. LEER UN TEXTO. NAVEGACIÓN AJAX.....	93
FIGURA 91: FUNCIONALIDADES. LEER TEXTO. DESACTIVAR AJAX.....	94
FIGURA 92: FUNCIONALIDADES. VISUALIZAR GOOGLE-MAPS. PANTALLAS.....	95
FIGURA 93: FUNCIONALIDADES. VISUALIZAR GOOGLE-MAPS. ESTILOS CSS.....	95
FIGURA 94: FUNCIONALIDADES. VISUALIZAR GOOGLE-MAPS. SCRIPTS.....	96
FIGURA 95: FUNCIONALIDADES. VISUALIZAR GOOGLE-MAPS.JS/APP.JS. VER MAPA.....	96
FIGURA 96: FUNCIONALIDADES. VER GOOGLE-MAPS. CODIGO CHECK_NETWORK ().....	97
FIGURA 97: FUNCIONALIDADES. VER GOOGLE-MAPS. OBJECT.....	98
FIGURA 98: FUNCIONALIDADES. VER GOOGLE-MAPS. CSS. OBJECT.....	99
FIGURA 99: FUNCIONALIDADES. ACCEDER PÁGINA EXTERNA. PANTALLAS.....	99
FIGURA 100: FUNCIONALIDADES. ACCEDER PÁGINA EXTERNA. CÓDIGO.....	99
FIGURA 101: PHONEGAP BUILD. IMAGEN DE COMPILACIÓN.....	100
FIGURA 102: PHONEGAP BUILD. PLANES DEL SERVICIO.....	101
FIGURA 103: PHONEGAP BUILD. REPOSITORIO GITHUB.....	102
FIGURA 104: PHONEGAP BUILD. APP EMPAQUETADA.....	102
FIGURA 105: PHONEGAP BUILD. CONFIG.XML. NAME-DESCRIPTION.....	103
FIGURA 106: PHONEGAP BUILD. CONFIG.XML. LAUNCHER, SPLASH-SCREENS.....	103
FIGURA 107: PHONEGAP BUILD. CONFIG.XML. VERSIONES-ORIENTACION-INSTALACIÓN.....	104
FIGURA 108: PHONEGAP BUILD. CONFIG.XML. FUNCIONALIDADES REQUERIDAS.....	105
FIGURA 109: PHONEGAP BUILD. APLICACIÓN GITHUB INSTALADA EN WINDOWS 7.....	105
FIGURA 110: GOOGLE PLAY. KEYSTORE.....	106
FIGURA 111: GOOGLE PLAY. APK COMPILADA CON CERTIFICADO.....	107

DEDICATORIA

Dedico este trabajo y con ello todo lo que significa el haber logrado una meta llena de sacrificios, a mi querida esposa e hijo, Elena y Gonzalo. Mi familia ha sido mi apoyo en estos años de Carrera, al igual que lo son en mi vida. Sobre todo a mi esposa por la paciencia, comprensión, ánimos y cariño que me ha ofrecido durante todo este recorrido y a mi hijo por aquellos momentos que no he podido compartir con él.

A mis padres por haberme enseñado que con esfuerzo, constancia y perseverancia se pueden lograr las metas que uno desee.

Y a Dios por haberme dado las fuerzas necesarias para conseguirlo.



RESUMEN

Este trabajo ha tenido como objetivo la realización de una aplicación para móviles desarrollada en HTML5 llamada "Audioguía Turística de El Real Monasterio de San Lorenzo de El Escorial".

Se han utilizado tecnologías como HTML5, javascript, jQuery Mobile (framework de javascript), CSS, ThemRoller, y Phonegap (ahora apache Cordoba). También se ha usado Eclipse como IDE para el desarrollo con el pluging de Aptana.

Se ha logrado una aplicación instalable en cualquier dispositivo móvil (Smartphone) o Tablet de cualquier tamaño, obteniéndose una app multidispositivo de tipo híbrida que puede usar y disponer de los recursos que posee el teléfono móvil como hacer llamadas, geolocalización, etc. Aparte de todo ello, para alcanzar una app multiplataforma se ha usado Phonegap Build, proporcionándonos con la compilación en la nube varios archivos instalables en distintas plataformas como Android, Windows-Phone, etc.

Por tanto, se ha obtenido el desarrollo de una aplicación con HTML5 multidispositivo y multiplataforma.

Palabras clave: App móvil, multidispositivo, multiplataforma, HTML5, CSS, jQuery Mobile, javascript, Phonegap, Phonegap Build, desarrollo en HTML5.

RESUM

Aquest treball ha tingut com a objectiu la realització d'una aplicació per a mòbils desenvolupada en HTML5 anomenada " Audioguia Turística del Reial Monestir de San Lorenzo de l'Escorial " .

S'han utilitzat tecnologies com HTML5 , javascript , jQuery Mobile (framework de javascript) , CSS , ThemRoller , i PhoneGap (ara apache Cordoba) . També s'ha usat Eclipse com IDE per al desenvolupament amb el pluging de Aptana .

Alumno: Jesús Villegas Guerra

S'ha aconseguit una aplicació que es pot instal·lar en qualsevol dispositiu mòbil (Smartphone) o Tablet de qualsevol mida, obtenint-se una app multidispositiu de tipus híbrida que pot usar i disposar dels recursos que té el telèfon mòbil com fer trucades , geolocalització, etc . A part de tot això , per assolir una app multiplataforma s'ha fet servir PhoneGap Build , proporcionant amb la compilació en el núvol diversos arxius instal·lables en diferents plataformes com Android , Windows- Phone , etc .

Per tant , s'ha obtingut el desenvolupament d'una aplicació desenvolupada amb HTML5 multidispositiu i multiplataforma .

Paraules clau : App mòbil , multidispositiu , multiplataforma , HTML5 , CSS , jQuery Mobile, javascript , PhoneGap , PhoneGap Build , desenvolupament en HTML5 .

ABSTRACT

This work has been aimed at the realization of a mobile application developed in HTML5 called " Tourist Audio Guide The Royal Monastery of San Lorenzo de El Escorial" .

We used technologies such as HTML5, javascript, jQuery Mobile (javascript framework) , CSS, ThemRoller and Phonegap (now apache Cordoba) . Eclipse has also been used as IDE for development with Aptana plugging.

There has been an installable application on any mobile device (Smartphone) or tablet of any size, obtaining a multi-device app hybrid type that you can use and dispose of resources held by the mobile phone as making calls, geolocation , etc. . Beyond that, to achieve cross-platform app Phonegap Build has been used, providing the cloud compile several installable files on different platforms such as Android, Windows -Phone, etc.

Therefore, we have got the development of a multi-device and cross-platform application with HTML5.

Keywords : Mobile App, multi-device , multi-platform , HTML5 , CSS , jQuery Mobile, Javascript , Phonegap , Phonegap Build , HTML5 development .

1. INTRODUCCIÓN

Hoy en día, y cada vez más, las personas tendemos a aprovechar el poco tiempo que tenemos y con una de las cosas con la que lo conseguimos es con el móvil. Se acerca a las capacidades de un ordenador, posee un sistema operativo muy funcional, podemos guardar datos, extraerlos, comunicarnos y además a todo ello nos ayuda la iCloud (la nube).

Las estadísticas no nos mienten (www.abc.es, tecnología “Las once estadísticas increíbles del marketing de móviles”, 31/05/2012), el 91% de los usuarios de smartphones tienen el teléfono siempre al alcance, se tardan 90 segundos en mandar un mensaje de texto (sms, whatsapp, line), hay 7.000 millones de personas en la Tierra y 5.100 tienen teléfono móvil.

Precisamente hace poco tiempo, hablando de hacer un blog para una asociación con WordPress me comentaban los asociados que eso sería fenomenal ya que con una web se centralizarían las noticias y sugerí que se hiciera también la web adaptada al móvil, en lo que alguno de ellos dijeron que si ya estaba bien tener una web, que la tuvieran en el móvil, sería maravilloso porque lo podrían consultar en cualquier momento. Por ello, ya vemos la importancia del desarrollo de app para móviles.

A continuación expondré una pequeña introducción de porqué realizar app con HTML5, que tipos hay, sus ventajas e inconvenientes y para terminar la introducción los objetivos que busco al realizar con este trabajo fin de carrera.

1.1. App con HTML5

¿Por qué Apps con HTML5? La siguiente figura lo dice todo:



Figura 1: App con HTML5.

Fuente <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>

Alumno: Jesús Villegas Guerra

HTML5 no es nuevo, proviene del antiguo HTML4 con etiquetado de markup. Sin embargo, hay algunos aspectos que hacen que HTML5 sea diferente y atractivo. Apple, Opera y Mozilla en 2004 crean WHATWG para desarrollar HTML5 y más adelante W3C comienza a trabajar con WHATWG. Además hoy en día Google, Nokia y muchos otros tienen su apoyo, así como hace un mes se ha iniciado por parte de Firefox su nuevo sistema operativo Firefox OS con base de desarrollo en HTML5 para plataformas móviles.

Algunas de las novedades más interesantes de HTML5 son:

- Nuevas etiquetas para dar más significado a las páginas (web semántica).
- Más control en los formularios.
- Almacenamiento offline.
- Soporte para audio y video.
- Elemento canvas para crear dibujos y animaciones.
- Sockets web.
- Doctype sencillo.
- XHTML5.

A lo que se añaden algunas reglas para su desarrollo como:

- Las nuevas características deben basarse en HTML, CSS, DOM y JavaScript.
- Reducir la necesidad de plugins externos.
- Mejor manejo de errores.
- Más etiquetas para reemplazar scripting.
- Debe ser independiente del dispositivo.
- El proceso de desarrollo es visible para el público.

1.2. Tipos de Apps

Podemos encontrar 3 tipos de apps:

- **Aplicaciones nativas:** Que son desarrolladas en el propio lenguaje del terminal que se use.

- **Aplicaciones web:** Una web a la que se accede a través de una URL en el navegador del dispositivo (Safari, Chrome u otro) y se adapta al formato de la pantalla del terminal donde se esté visualizando para que tenga aspecto de navegación App.
- **Aplicaciones híbridas:** Aúnan lo mejor de los dos anteriores modelos. Este tipo de aplicaciones permite el uso de tecnologías multiplataforma como HTML, Javascript y CSS pero permiten acceder a buena parte de los dispositivos y sensores del teléfono.

Buena parte de la infraestructura es tipo web y la comunicación con los elementos del teléfono se hace mediante comunicadores tales como Phonegap (actualmente Apache Córdoba).

1.3. Ventajas e inconvenientes de usar aplicaciones híbridas

A continuación muestro un gráfico con las características de cada tipo de aplicaciones.

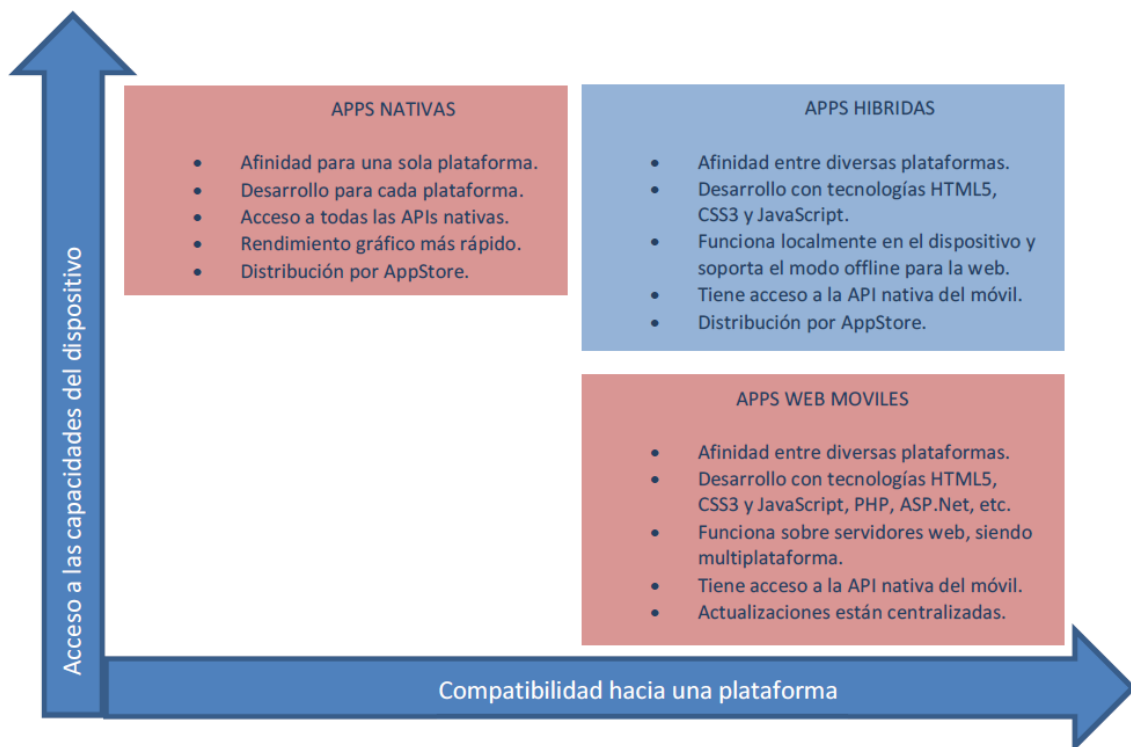


Figura 2: Características de aplicaciones nativas, web e híbridas

Elaboración propia. Traducido de la página <http://blogthinkbig.com/wp-content/uploads/Native-apps-vs-hybrid-apps-vs-web-apps.png>

Alumno: Jesús Villegas Guerra

Las ventajas e inconvenientes de las aplicaciones híbridas pueden ser las siguientes:

VENTAJAS	INCONVENIENTES
Son multiplataforma, permiten que el código fuente se pueda ejecutar en diferentes plataformas.	Las actualizaciones son manuales, el usuario debe acceder al canal de compra de aplicaciones para conseguir la actualización.
Poder acceder a todos los recursos del móvil (cámara, gps, llamadas, agenda).	En algunos casos se han de pasar validaciones de control para poder ser llevadas al market place.
No es necesaria la conexión a internet a excepción de aquellas partes que lo requieran.	La parte nativa a veces ha de ser desarrollada específicamente para cada plataforma.
Aúna lo mejor de las aplicaciones web y las nativas.	El rendimiento de la aplicación puede ser más lento que con la aplicación nativa.
No es necesario un lenguaje específico, la mayor parte utiliza HTML5.	
El desarrollo de esta aplicación es más barato, al requerir menor tiempo de implementación y más rápido.	
Posibilidad de subir la aplicación al mercado como google play, amazon store, app store, etc.	
La aplicación puede ser instalada en el dispositivo.	

Figura 3: Ventajas y desventajas de las apps híbridas.

Elaboración propia.

Por tanto, el realizar un tipo u otro de aplicación (nativa, web o híbrida) va a depender de muchos factores, entre ellos la complejidad, la actualización, la necesidad de rendimiento, el coste, la rapidez de desarrollo, etc. Es por lo que según nuestras necesidades se deberá tomar un camino u otro, aunque están tomando mucho auge las aplicaciones híbridas.

1.4. Hardware y software

En un principio utilizaré el hardware y software que mostraré a continuación , aunque dependiendo de cómo se desarrolle el proyecto podré utilizar otros nuevos.

Para el **hardware** cuento con:

- Ordenador HP TouchSmart con dos procesadores.
- Teléfono móvil Smartphone- Android marca Cubot 7+ de 3,5".

El **software** que utilizaré será:

- Microsoft Office.
- Microsoft Project.
- Microsoft Visio.
- Photoshop
- Axure RP para el prototipado y pruebas iniciales con usuarios.
- Navegadores Chrome (con plugin Ripple) y Firefox para debugger (depurador).
- IDE: Eclipse con instalación de Aptana y el SDK para Android.

1.5. Tecnologías utilizadas

Con respecto a las tecnologías que utilizaré, según me he estado documentado serán las siguientes, aunque no descarto cambiar alguna de ellas o ampliarlas según se vaya desarrollando el proyecto:

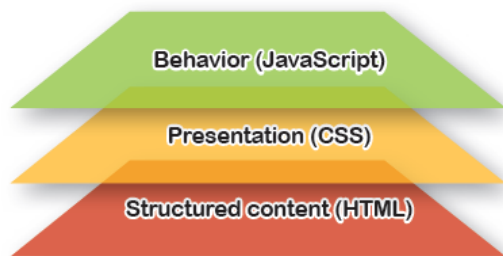


Figura 4: Tecnologías web.

(Fuente: Lyza Danger Gardner & Jason Grigsby, 2012. "Head first mobile web". Editorial O'Really).

-  **HTML5** : Utilizado para estructurar el contenido de la aplicación web.
-  **CSS3**: Da una presentación al contenido realizado con HTML5.
-  **JavaScript**: Añade un comportamiento determinado a los contenidos, con acceso al DOM (Document Object Model).
-  **jQuery Mobile**: Es un framework basado en javascript que nos facilita el desarrollo de aplicaciones móviles de una forma más rápida con acceso rápido al DOM, manejador de eventos, desarrollo de listados, botones e interacción técnicas de AJAX que además es compatible con otros marcos de aplicaciones móviles y plataformas como Apache Cordoba (Phonegap).
-  **Apache Cordoba (PhoneGap)**: es un framework para el desarrollo de aplicaciones móviles producido por Nitobi, y comprado posteriormente por Adobe Systems. Principalmente, PhoneGap permite a los programadores desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3. Las aplicaciones resultantes son híbridas, es decir que no son realmente aplicaciones nativas al dispositivo.
-  **ThemeRoller** : Permite crear estilos css de una manera sencilla para jQuery Mobile y poder usarlos en el desarrollo de la aplicación.

1.6. Objetivos del trabajo fin de carrera

El objetivo principal que quiero alcanzar con este trabajo fin de carrera (TFC en adelante) es plasmar los conocimientos adquiridos a lo largo de la carrera en la realización de una app que cumpla con los requisitos de las técnicas de desarrollo del software, así como de los principios de la IPO (interacción persona ordenador) , DCU (diseño centrado en el usuario) junto con los nuevos conocimientos que deseo adquirir:

- Mejorar la práctica de HTML con HTML5 y CSS3.
- Aprender JavaScript.

Alumno: Jesús Villegas Guerra

- Conocer el framework jQuery Mobile.
- Estudiar cómo poder trabajar con Apache Cordova (antiguo Phonegap).
- Poder formarme en realizar aplicaciones multiplataforma (Android, iOS, WindowsPhone, Blackberry) con Phonegap Build .
- Aplicar media query de HTML5 para realizar web apps adaptables (responsive web app).
- Realizar la aplicación propuesta en este proyecto que sea multidispositivo, multiplataforma, como comienzo de futuras web apps híbridas o web adaptables, aunque en este caso solo podré trabajar para realizar la aplicación en Android al no disponer de equipo para Mac.
- Que me puedan servir estos conocimientos para mi proyección laboral.
- Exponer que con HTML5 y Phonegap se pueden hacer aplicaciones que usen los recursos internos del Smartphone al igual que lo haría una aplicación nativa.

2. REQUISITOS DE LA APLICACIÓN

2.1. Descripción del proyecto

Se quiere realizar una aplicación que incorpore un sistema de audioguía turística para el Real Monasterio de San Lorenzo de El Escorial (Madrid) para ofrecer una experiencia interactiva basada en el atractivo del monumento.

No será una audioguía al uso, sino que contendrá además imágenes y texto de las distintas estancias del Real Monasterio (en adelante R.M.).

Una vez en la aplicación se tendrá acceso a los menús que configuran la misma, se podrá acceder a las distintas estancias o salas que lo componen de tal forma que el usuario pueda elegir que quiere leer o bien que quiera escuchar mientras pasea por las estancias del R.M. , además de poder observar alguna fotografía que muestre lo que está viendo o quiere ver. Con el aspecto de escuchar un texto hablado se está teniendo en cuenta la diversidad, y el poder tener accesible una aplicación para personas con problemas de visión ya que muchas veces las letras de los móviles son tan pequeñas que no es posible ser leídas por todas las personas.

Alumno: Jesús Villegas Guerra

También se tendrá en cuenta incluir en la aplicación información de carácter general como horarios al público, precios de la entrada en el R.M., biografías y otras informaciones de interés para el usuario.

Consideraremos que la aplicación se podrá adaptar a distintas plataformas de distinto tamaño ya sean móviles de 3,2", 4", 5", o tablets de 7" o 10".

El diseño de la aplicación tendrá en cuenta una buena usabilidad (con interfaz fácil e intuitiva) y diseño sencillo.

2.2. Requisitos funcionales mínimos a desarrollar

Tenemos que tener en cuenta que se presentarán los requisitos mínimos a desarrollar y que estos pueden ir cambiando conforme se va desarrollando la aplicación, ya que pueden surgir nuevos aspectos a tratar o modificar respecto de lo inicial.

Mostrare a continuación los requisitos mínimos que tendrá la aplicación:

- Se mostrará una pantalla inicial con un menú para que el usuario pueda elegir.
- El menú contendrá las distintas salas que conforman el R.M. y que pueden ser visitadas por los usuarios de la aplicación, como pueden ser Biblioteca, Casa del Rey, Panteones, Basílica.
- Contendrá además otro menú de más información.
- En el menú de más información se permitirá acceder a otros menus interesantes para el usuario como biografías, misterios, horarios
- El usuario podrá acceder al menú desde cualquier página en la que esté.
- El usuario en cualquier momento podrá ir a la pantalla "home" desde cualquier página en la que esté.
- El usuario podrá leer texto y ver imágenes como mínimo cuando entre en el menú de Biblioteca, Casa del Rey, Panteones y Basílica. Pudiendo ampliar texto e imagen a otros menús como biografías u horarios.
- El usuario podrá escuchar el texto que está leyendo al acceder a los menús de Biblioteca, Casa del Rey, Panteones y Basílica. Pudiendo ser ampliada la escucha a otros menús.

- Se podrá parar en cualquier momento la audición del texto de la página en la que se encuentre el usuario.
- Se podrá pausar la audición.
- Se podrá continuar la audición después de la pausa.
- Se mostrará el tiempo total de la audición y el tiempo transcurrido.
- Se parará la audición al salir de la página en la que se esté si estaba escuchando una audición.
- Se podrá acceder a llamar por teléfono al R.M. en caso de necesitarlo el usuario desde esta aplicación directamente.

2.3. Requisitos no funcionales

Entre los requisitos no funcionales se encuentran entre otros:

- Que la información se pueda adaptar a distintos tamaños de plataformas, tanto smartphones como tablets, siendo adaptable “responsive”.
- Estudiar la posibilidad de que los archivos de audio e imágenes sean externos a la aplicación para intentar obtener una aplicación lo más liviana posible, aunque esta posibilidad hay que probarla.
- Realizar un diseño sencillo y fácil de utilizar.
- Integrar “pop up” de ayudas donde sea necesario.
- Intentar que las audiciones no tengan una duración superior a los 180 segundos (3 minutos).
- La aplicación debe ser intuitiva y usar símbolos que el usuario conozca, aplicando técnicas de la IPO como la affordance (por ejemplo el símbolo cuadrado para parar la audición, etc).
- Se tendrá en cuenta la diversidad de las personas, en cuanto los textos pueden ser hablados, ayudando a aquellas personas que tienen problemas al no poder apreciar los textos pequeños de los móviles.
- En lo posible se seguirá la metodología DCU, aunque conozco que lleva más tiempo que la metodología clásica de desarrollo de software, por los condicionantes de la planificación temporal usaré esta metodología de forma reducida y en algunas partes del diseño.

2.4. Metodología DCU

Actualmente la mayoría de los modelos de desarrollo de software que la industria utiliza son los basados en los métodos propuestos por la Ingeniería del Software aunque existan estándares internacionales ISO, como la ISO 13407, que describe como un proceso de diseño centrado en el usuario (DCU) puede conseguir sistemas más usables (Granollers, T.; Lorés, J. Perdrix F.; 2002).

A continuación muestro el modelo clásico:

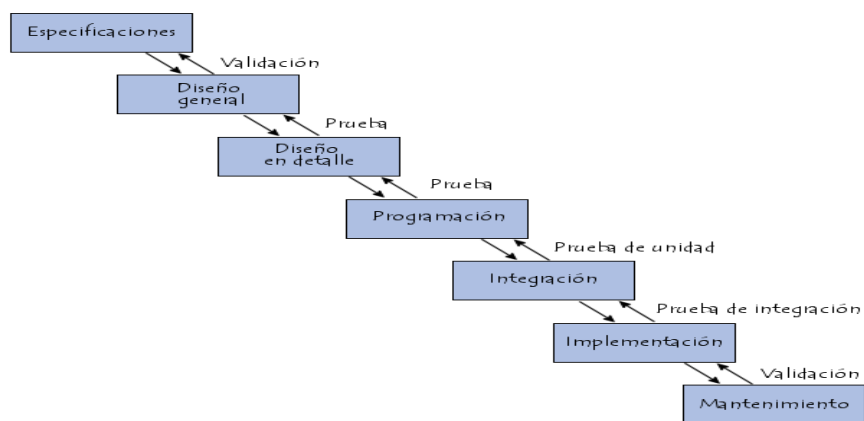


Figura 5: Modelo clásico de desarrollo de software.

Considero que este modelo clásico (figura 5) tiene la desventaja de que si las especificaciones del software no se hacen bien, ocurriendo esto muy a menudo porque una vez terminadas tenemos otras nuevas que no habíamos tomado en cuenta, y tenemos realizada la fase de diseño y programación, entonces debemos volver atrás con el coste que ello supone o dejar sin implementar esa especificación nueva o modificada por el coste añadido que conlleva.

La mejor opción sería por tanto estudiar una pequeña parte de los requisitos con cierta autonomía, y diseñarla, programarla y probarla. Una vez el cliente la haya dado por buena, hacer lo mismo con otra parte. Esto se denomina ciclo de vida iterativo e incremental (iterativo porque se repite dentro un mismo proyecto e incremental porque se realiza por partes). Otros autores lo llaman modelo evolutivo y otros MPlu+a (Granollers,T; y otros, 2002) , que significa Modelo de Proceso de la Ingeniería de la usabilidad y de la accesibilidad mostrado en la figura 6, basados todos ellos en un diseño más orientado hacia el usuario.

En resumen lo que se pretende es llegar a que se mejore la usabilidad del software.

Por ello, intentaré utilizar este sistema, si bien por falta de tiempo puede que en la fase de diseño no pueda realizar un prototipado que pueda utilizar para que algunos usuarios puedan probarlo, dando lugar a que no pueda hacerlo hasta la fase de Implementación.

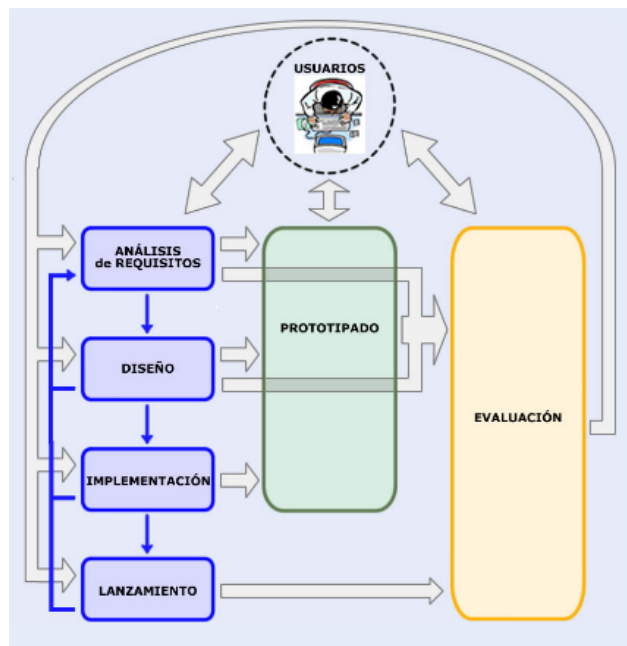


Figura 6: Modelo DCU de desarrollo de software.

Fuente (<http://www.grihohcitolos.udl.cat>)

2.5. Riesgos del proyecto

Desde el inicio del proyecto hay que reflexionar sobre los riesgos para tenerlos en cuenta y minimizarlos.

Por ello, pienso que los riesgos más importantes a tener en cuenta en el momento de diseñar y realizar el proyecto son:

- Falta de experiencia en el conocimiento de aplicaciones basadas en web (web apps), por la necesidad de aprendizaje de nuevos lenguajes y frameworks como JavaScript, jQuery o jQuery Mobile.
- Problemas que pueda tener al traducir las ideas planteadas en los requisitos funcionales y poder aplicarlas a la aplicación.

- El tiempo limitado para la realización del proyecto, teniéndolo que compaginar trabajo con estudios.
- Problemas añadidos que pueda tener con el IDE y tecnologías usadas.
- En la fase de Diseño no pueda usar un prototipo funcional para que pueda ser probado, ya que requiere un tiempo considerable su realización y pruebas con usuarios.

3. PLANIFICACIÓN Y EJECUCIÓN DEL PROYECTO

3.1. Tareas principales

- **Plan del proyecto:** Tiene una duración de 15 días. Se corresponde con la PEC1 de la temporización de UOC. Realizando una investigación inicial del proyecto que se quiere realizar, documentando la definición del proyecto, funcionalidades mínimas a tener en cuenta, objetivos y planificación.
- **Diseño:** Durará 36 días. Se corresponde con la PEC2 de la temporización de UOC. Se efectuará una investigación inicial donde se estudiará el mercado y población a la que va dirigida la aplicación, un diseño de escenarios y flujos de interacción, así como un prototipo en alta definición que nos servirá para la implementación (de aquellos aspectos que puedan ser realizados en maqueta) y probados con usuarios. Finalmente, iniciar una maqueta real con HTML5, CSS3 y jQuery Mobile de la aplicación para comenzar con la implementación.
- **Implementación:** Duración 34 días. Se corresponde con la PEC3 de la temporización de UOC. Se implementará la aplicación siguiendo el prototipo creado en la fase anterior, haciendo pruebas por cada funcionalidad terminada.
- **Finalización del proyecto:** Duración 21 días. Si quedaran funcionalidades o partes de la implementación de la fase anterior, podrán ser terminadas. Se realizará la memoria final y el video para defender el proyecto realizado que contendrá una explicación del proyecto, de forma razonada y una demostración de la aplicación.

Comentar que si se termina con antelación algunas de las fases del proyecto se comenzará la siguiente con un tiempo extra para su realización.

3.2. Planificación: Diagrama de Gantt

A continuación muestro una tabla con la planificación temporal y un diagrama de Gantt donde se puede observar más gráficamente la temporización del proyecto.

Nombre de la tarea	Duración	Comienzo	Fin
PLAN DEL PROYECTO- PEC1	15 días	lun 03/03/14	lun 17/03/14
Investigación inicial	6 días	lun 03/03/14	sáb 08/03/14
Definición del proyecto	4 días	dom 09/03/14	mié 12/03/14
Planificación del proyecto	2 días	jue 13/03/14	vie 14/03/14
Instalación Eclipse, Sdk y pruebas	2 días	sáb 15/03/14	dom 16/03/14
Redacción del proyecto	14 días	lun 03/03/14	dom 16/03/14
Entrega	1 día	lun 17/03/14	lun 17/03/14
DISEÑO- PEC2	36 días	mar 18/03/14	mar 22/04/14
Investigación	4 días	mar 18/03/14	vie 21/03/14
Diseño de escenarios y flujo de interacción	6 días	sáb 22/03/14	jue 27/03/14
Prototipado	16 días	vie 28/03/14	sáb 12/04/14
Evaluación	6 días	dom 13/04/14	vie 18/04/14
Redacción del proyecto	35 días	mar 18/03/14	lun 21/04/14
Pruebas unitarias con Eclipse	36 días	mar 18/03/14	mar 22/04/14
Estudio de tecnologías: Html5, CSS3, jQuery Mobile	36 días	mar 18/03/14	mar 22/04/14
Entrega	1 día	mar 22/04/14	mar 22/04/14
IMPLEMENTACIÓN - PEC3	34 días	mié 23/04/14	lun 26/05/14
Implementación prototipo básico (Html5+jQuery Mobile)	23 días	mié 23/04/14	jue 15/05/14
Configurar imágenes, texto y voz en mp3	4 días	vie 16/05/14	lun 19/05/14
Afinar la implementación	6 días	mar 20/05/14	dom 25/05/14
Prueba	33 días	mié 23/04/14	dom 25/05/14
Redacción del proyecto	33 días	mié 23/04/14	dom 25/05/14
Entrega	1 día	lun 26/05/14	lun 26/05/14
FINALIZACIÓN DEL PROYECTO	21 días	mar 27/05/14	lun 16/06/14
Terminar la implementación	5 días	mar 27/05/14	sáb 31/05/14
Elaborar la memoria	8 días	dom 01/06/14	dom 08/06/14
Elaborar presentación del video del TFC	7 días	lun 09/06/14	dom 15/06/14
Entrega	1 día	lun 16/06/14	lun 16/06/14

Figura 7: Tabla de planificación temporal

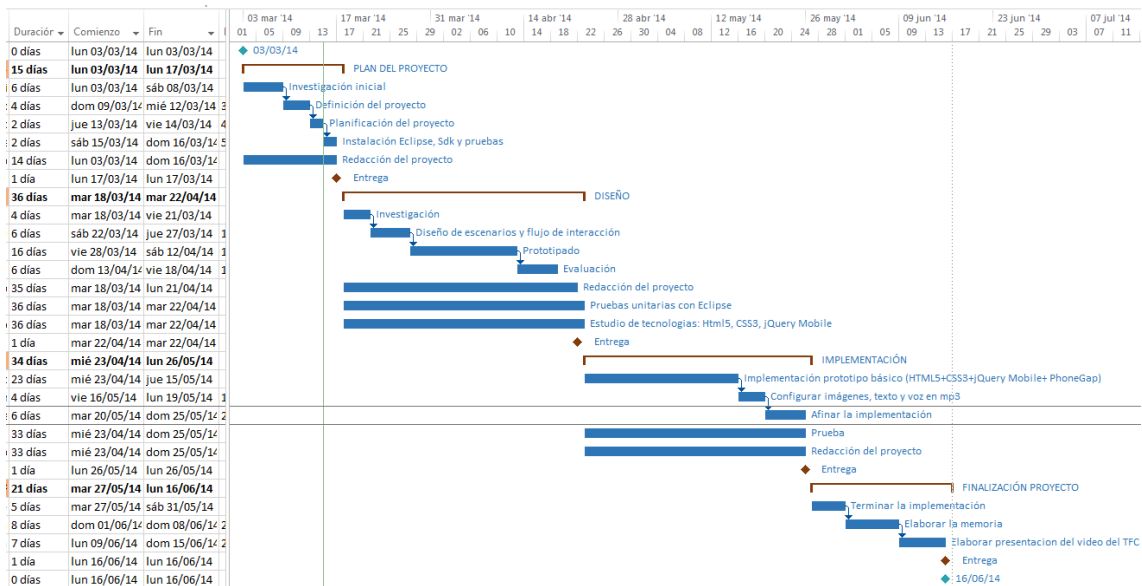


Figura 8: Planificación temporal. Diagrama de Gantt

Fuente: Realización propia con Project de Microsoft

4. INVESTIGACIÓN

La investigación inicial se realiza para conocer a los usuarios, la competencia, y como los usuarios interactúan con los distintos aplicativos existentes en el mercado. A raíz de este punto se podrá conocer más detalladamente que funcionalidades requieren los usuarios y como debe ser la interfaz de usuario para que sea más usable o también nos sirve este estudio si ya se conocen que funcionalidades debe tener una aplicación y queremos mejorarla en cuanto a "User Experience" (UX), es decir, en la experiencia de usuario y usabilidad de la aplicación. En definitiva de lo que se trata es de perfeccionar las aplicaciones existentes y aplicar a nuestra propia aplicación esas mejoras en base a la investigación llevada a término.

Para realizar la investigación se utilizará distintos medios:

- **Investigación contextual:** Sería interesante realizar la prueba en el lugar de uso. En nuestro caso la aplicación puede utilizarse tanto en casa como, en el lugar de turismo o en cualquier otro lugar
- **Entrevista corta:** Para conocer información cualitativa usada en el contexto de uso y escenarios.

Alumno: Jesús Villegas Guerra

- **Análisis competitivo:** Utilizaremos dos aplicaciones para realizar una prueba de usabilidad inicial, en nuestro caso se usarán las aplicaciones tomadas de Google Play “El Escorial” y “National Museum of Korea”, que aunque no sean de la misma temática en alguno de ellos (palacios, monasterios), si que se parecen bastante a la aplicación final deseada como es el caso de la audio guía del National Museum of Korea. Este análisis es una manera de entender realmente aquellos elementos que funcionan y aquellos que no sirven para que el usuario pueda alcanzar sus objetivos.

En los anexos se puede ver una copia de la entrevista y de las hojas que les he presentado a los usuarios del test para que realizasen sus tareas.

En cuanto al **diseño conceptual**, me he basado en la investigación del perfil de usuario para realizar unas tareas determinadas y como es el comportamiento de los usuarios para acercarnos al diseño de la interfaz. Las tareas mínimas que debería realizar para el estudio de la aplicación serían las siguientes:

- Leer un texto sobre un tema concreto en la aplicación.
- Ir al inicio de la aplicación desde la pantalla donde se encuentre.
- Escuchar una audición y pararla (en el caso de tener la aplicación dicha funcionalidad).

Para realizar los **escenarios** me he apoyado en la información recogida en los cuestionarios cortos, así como en un estudio de Planificación y Gestión Turística en destinos patrimoniales: el caso del Real Sitio de San Lorenzo de El Escorial de la autora Minguez , M^a del Carmen (Marzo de 2007).

Los **flujos de interacción** se realizan basándonos en la experiencia obtenida en el uso de las aplicaciones estudiadas en la investigación de la competencia y de ahí extraer un flujo que seguirá nuestra aplicación para poder ser desarrollada.

Finalmente para realizar el **prototipo** he utilizado la herramienta Axure RP que crea un prototipo simulado en el pc y que puede ser probado por los usuarios. Este prototipo junto con las conclusiones de usabilidad por parte de los usuarios es en lo que nos basaremos para el desarrollo real de la aplicación.

4.1. Perfiles de usuario

Para conocer los posibles usuarios de la aplicación sobre todo me he basado en estudios estadísticos sobre las personas que visitan el Monasterio de El Escorial.

Según el estudio realizado por Minguez, M^a del Carmen (2007) podemos extraer dos tablas tomadas de información turística:

	Primavera	Verano	Otoño	Invierno	TOTAL
Unas horas	34,6	49,5	39,8	61,6	46,3
Un día	53	30,2	48,7	31,2	40,7
Pernocta	12,4	20,5	11,5	7,2	13
TOTAL	100	100	100	100	100

Figura 9: Duración estancia en R.M. El Escorial en %

Fuente: Encuestas San Lorenzo de El Escorial (agosto 2003-julio 2004)

	E	F	M	A	M	Jn	Jl	A	S	O	N	D	Total
Local	101	112	216	111	140	102	218	207	179	232	268	458	2.344
C Madrid	2.339	1.226	2.323	1.387	2.487	1.727	2.246	2.871	1.901	1.994	955	1.327	22.783
España	1.055	1.330	3.985	2.349	2.456	2.435	2.102	4.093	2.878	2.769	783	1.305	27.540
U. E	418	762	1.076	1.778	1.695	1.123	974	1.420	1.517	1.331	337	300	12.731
Extranjero	711	673	820	658	819	932	1.040	774	1.242	1.030	320	377	9.396
Total	4.624	4.103	8.420	6.283	7.597	6.319	6.580	9.365	7.717	7.356	2.663	3.767	74.794

Figura 10: Consultas en Oficina información turística de El Escorial (meses y procedencia).

Fuente: Oficina Local de Información de San Lorenzo de El Escorial.

Y basándonos en el "Informe sobre el turista en la Comunidad de Madrid" tenemos que:

Alumno: Jesús Villegas Guerra

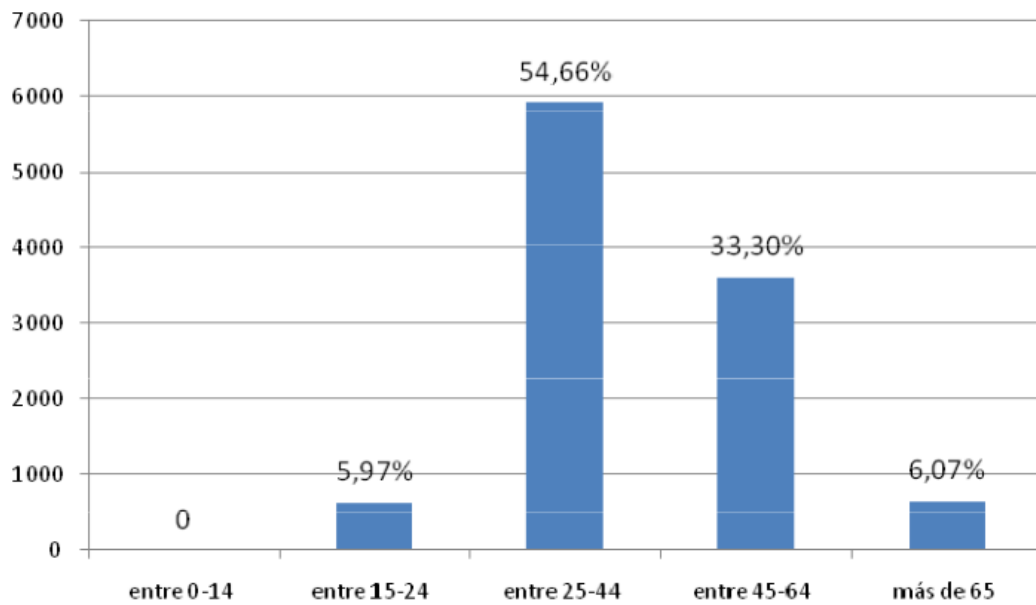


Figura 11: Edad media del turista.

Fuente: Informe sobre el turista en la Comunidad de Madrid

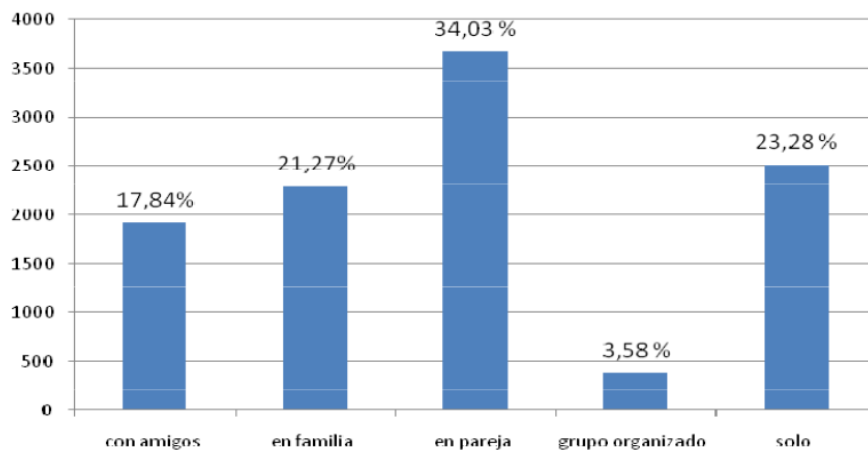


Figura 12: Acompañante del turista.

Fuente: Informe sobre el turista en la Comunidad de Madrid

EL ESCORIAL	44%
ARANJUEZ	27%
ALCALA DE HENARES	11%
CHINCHON	6%
MANZANARES EL REAL	2%
NAVALCARNERO	1%
BUITRAGO DE LOZOYA	1%
OTROS MUNICIPIOS	8%

Figura 13: Lugares visitados en Madrid en %.

Fuente: Informe sobre el turista en la Comunidad de Madrid

Podemos ver que la estancia en el Escorial de los turistas durante unas horas es del 46,3 % de las personas, que los que más visitan el R.M. son los españoles, seguidos de extranjeros Europeos. Además el 54,66% de los visitantes tienen una edad entre 25-44 años que vienen acompañados por su pareja en un 34,03 % de los casos, seguido de familias en un 21,27%. Y finalmente el lugar más visitado en la Comunidad de Madrid es El Escorial con un 44% del total.

Por tanto, el mejor perfil para el estudio de la aplicación sería una persona que su estancia en el Escorial es de horas, español, con una edad comprendida entre 25-44 años y que viene acompañado de su pareja. Otro perfil interesante sería el mismo pero con acompañantes familias y por último también hay un perfil de extranjeros.

De esta investigación se desprende que la aplicación no tiene que ser muy extensa ya que por porcentajes la estancia en el R.M. de El Escorial es de horas y por tanto necesitarían una aplicación que tuviese condensado todos los lugares y estancias que podrán visitar en unas horas sin ser exhaustivos. La población que más la podrá usar sería: Personas entre 25-44 años en parejas y familias. Además sería interesante poder realizar la aplicación con texto en

inglés dado la gran afluencia de público extranjero, sin embargo esta no es una funcionalidad mínima exigida y requiere de un trabajo añadido de traducción y programación.

Los perfiles de usuarios con lo que trabajaré para esta aplicación estará comprendido en su mayoría por el perfil mayoritario. Efectivamente nos dejamos parte de la población sin embargo no por ello será menos representativa la muestra, ya que nos dará un resultado del 65% de los fallos de usabilidad según Jakob Nielsen y atendiendo a Steve Krug (2010) con tres personas es suficiente para un estudio de estas características:

Perfil usuario 1	
Edad	42
Sexo	Mujer
Profesión	Administrativa
Formación	Estudios Superiores
Miembros en la familia	3
Nivel de inglés	Avanzado
Nivel de conocimientos informáticos	Medios. Conocimiento de office.
Uso del móvil	Diario- Smartphone.
Clasificación del usuario	Usuario poco experimentado.

Figura 14: Tabla de Perfil Usuario 1

Perfil usuario 2	
Edad	35
Sexo	Hombre
Profesión	Ingeniero informático
Formación	Estudios Superiores
Miembros en la familia	3
Nivel de inglés	Medio
Nivel de conocimientos informáticos	Avanzados, sistemas.
Uso del móvil	Diario- iPhone.
Clasificación del usuario	Usuario muy experimentado.

Figura 15: Tabla de Perfil de Usuario 2

Perfil usuario 3	
Edad	10
Sexo	Hombre
Profesión	Estudiante
Formación	Estudios básicos
Nivel de inglés	Medio-Avanzado
Dispositivos de uso internet	Ordenador, portátil, móvil (esporádicamente)
Nivel de conocimientos informáticos	Bajos, Office.
Uso del móvil	Esporádicamente.
Clasificación del usuario	Usuario poco experimentado.

Figura 16: Tabla de Perfil de Usuario 3

4.2. Contexto de uso

El contexto de uso elegido es en casa o cualquier lugar como paso previo posible al lugar que se va a visitar, sea un museo, un palacio, monumento, etc. Aunque también como comentamos la aplicación puede ser usada en el lugar de visita y en el momento de la visita.

4.3. Análisis de tareas

Las tareas que se realizarán en los test de usabilidad de las aplicaciones “El Escorial” y “National Museum of Korea” van encaminadas a ver el comportamiento de los usuarios en las aplicaciones de la competencia.

He diseñado las pruebas para que sean genéricas, puedan servirnos posteriormente para nuestra aplicación, y sean acordes a los requisitos que queremos posea la aplicación:

- **Leer un texto sobre algún tema concreto de la aplicación:** Con esta acción queremos que el usuario acceda desde la pantalla inicial a algún texto que le pueda ser de su interés para ver como interacciona con la aplicación.
- **Ir al inicio de la aplicación desde la pantalla donde se encuentre:** Buscamos ver al usuario buscando algún botón y como lo hace para ir al inicio de la aplicación, observar como usa los botones específicos de aplicaciones de móvil.
- **Escuchar una audición y pararla (si la aplicación lo posee):** Queremos ver como usa el dispositivo de audio en los móviles.

4.4. Test de usabilidad: Estudio de la competencia.

Para probar algunas de las funcionalidades requeridas en nuestro diseño me he basado en dos aplicaciones. Las razones han sido porque National Museum of Korea posee muchas características que querría incorporar a la aplicación a desarrollar y la aplicación de El Escorial posee la temática que desarrollaré:

- El Escorial
- National Museum of Korea

Usuario 1

El Escorial

Leer un texto

Le gustó:

- El mapa de Google maps, sin embargo se creía que era un mapa del Monasterio.



Foto 1

- Las fotos en la página.



Foto 2

- Es muy útil la información que dan en texto.

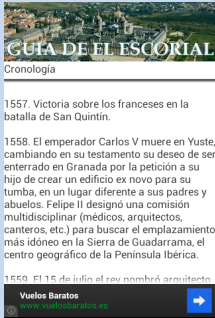
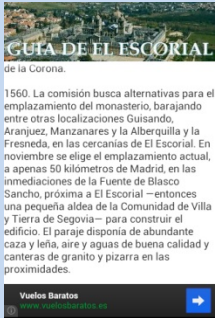
No le gustó:

- Publicidad.
- Echa en falta un plano del edificio.

Alumno: Jesús Villegas Guerra

	<p>Observaciones:</p> <ul style="list-style-type: none"> Accede bien a diferentes textos, utilizando el botón atrás. No le ha costado trabajo, ha sido intuitivo el uso del botón. El usuario como indica en el perfil está acostumbrado a usar el botón atrás de Android.
Ir al inicio	<p>Le gustó: - -</p> <p>No le gustó: Que no indica en ningún sitio donde está el inicio de la aplicación.</p> <p>Observaciones: Ha ido pulsando botón atrás hasta llegar al inicio de la aplicación.</p>
Escuchar audición	No posee.

Figura 17: Tabla Usuario 1. Test usabilidad competencia app "El Escorial"

Usuario 2	
El Escorial	
Leer un texto	<p>Le gustó: Limpieza de la interfaz por el texto negro sobre fondo blanco.</p> <p>No le gustó: Que la situación o el lugar donde nos encontramos no está situado en el encabezado de la página dentro del texto por lo que al pasar la página perdemos donde nos encontramos.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>Foto 3</p> </div> <div style="text-align: center;">  <p>Foto 4</p> </div> </div> <p>Observaciones: --</p>

Alumno: Jesús Villegas Guerra

Ir al inicio	Le gustó: --
	No le gustó: <ul style="list-style-type: none"> No hay botón específico para ir al inicio y hay que ir jugando con el botón “volver” y el botón “atrás” del teléfono Android. No le gusta la publicidad.
	Observaciones: Se nota que es un usuario experimentado.
Escuchar audición	No posee

Figura 18: Tabla Usuario 2. Test usabilidad competencia app “El Escorial”

Usuario 3	
El Escorial	
Leer un texto	Le gustó: Tiene muchos títulos y apartados y es fácil de usar.
	No le gustó: --
	Observaciones: Consigue fácilmente acceder al cualquier parte del texto.
Ir al inicio	Le gustó: --
	No le gustó: --
	Observaciones: Busca el inicio, marca el botón de atrás. Y utiliza el botón volver hasta llegar al inicio. Nos comenta que debería tener más botones para volver.
Escuchar audición	No posee.

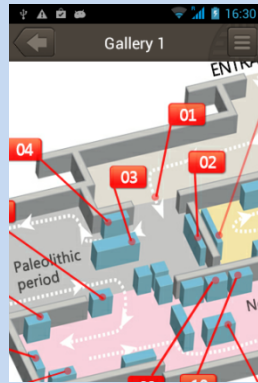
Figura 19: Tabla Usuario 3. Test usabilidad competencia app “El Escorial”

Usuario 1

National Museum of Korea

Leer un texto

Le gustó:



Al ir a encontrar un texto que quisiera leer, ha encontrado una pantalla con un plano en 3D que le ha gustado.

Foto 5

No le gustó:

- En la pantalla de inicio: Quiere avanzar hacia abajo y la pantalla rebota y no sabe si ha cambiado. Lo hace luego más lentamente y ve que no cambia que es el final y que solo hay 2 apartados.

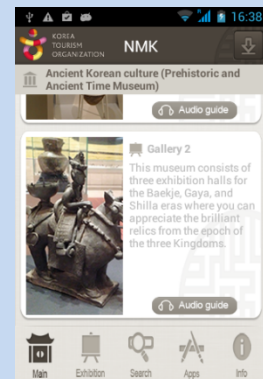


Foto 6

Observaciones:

- No encuentra un texto para leer del listado que mostramos en las siguientes imágenes, solo se va a la imagen y se accede directamente al audio desde un listado (Foto 7) sin posibilidad de leer un texto (Foto 8)

Alumno: Jesús Villegas Guerra

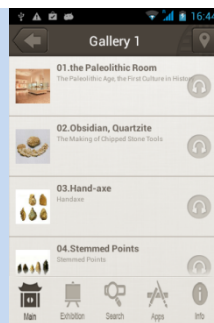


Foto 7

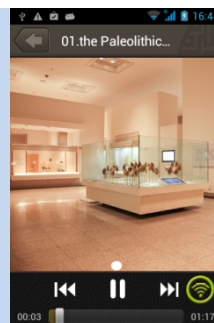


Foto 8

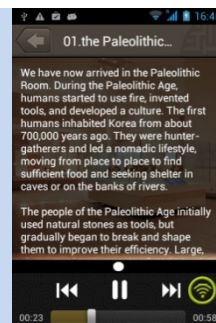


Foto 9

A la tercera vez encuentra que pulsando sobre la imagen de la Foto 8 se accede a un listado y en ese listado al pulsar se accede a una audioguía que automáticamente lee el texto (Foto 9).

Sin embargo, preguntando como ha encontrado el texto nos comenta que no sabe cómo. Le decimos que lo haga otra vez y no lo consigue.

Ir al inicio

Le gustó:

Es fácil ir al inicio ya que hay un icono visible y se accede bien al mismo, Nos dice que claramente se ve el icono de "Main".

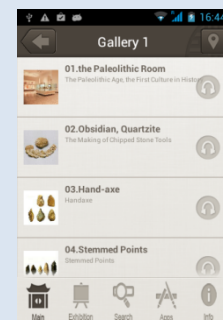


Foto 10

No le gustó: --

Observaciones: --

Escuchar audición

Le gustó:

Se accede automáticamente al audio.

Alumno: Jesús Villegas Guerra

	No le gustó: --
	Observaciones: --

Figura 20: Tabla Usuario 1. Test usabilidad competencia app “National Museum of Korea”

Usuario 2	
National Museum of Korea	
Leer un texto	<p>Le gustó:--</p> <p>No le gustó: El texto no se hace visible hasta que se pulsa sobre la pantalla. Debería poder verse el texto, aunque nos comenta que al ser una audioguía su función está más enfocada al audio que al texto.</p> <p>Observaciones: --</p>
Ir al inicio	<p>Le gustó:</p> <ul style="list-style-type: none"> • Que hay un botón en el encabezado de vuelta atrás. <p>No le gustó:</p> <ul style="list-style-type: none"> • El menú permanente en la parte inferior. • Al llegar a la pantalla de inicio el menú pop up no tiene botón de salida o cerrar. <p>Observaciones: --</p>
Escuchar audición	<p>Le gustó:</p> <ul style="list-style-type: none"> • Los controles de audio están muy bien diseñados. <p>No le gustó: --</p> <p>Observaciones: --</p>

Figura 21: Tabla Usuario 2. Test usabilidad competencia app “National Museum of Korea”

Usuario 3	
National Museum of Korea	
Leer un texto	<p>Le gustó: --</p> <p>No le gustó:--</p> <p>Observaciones:</p> <ul style="list-style-type: none"> • Ha ido probando varios menús de la barra inferior de menus de la Foto 11 hasta encontrar un listado (Foto 13) • Al listado ha accedido desde la pantalla “Gallery” (Foto 12)

Alumno: Jesús Villegas Guerra

	<p>en la que ha encontrado un menú y ha entrado al audio, después de pulsar sobre la pantalla ha leído el texto.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Foto 11</p> </div> <div style="text-align: center;">  <p>Foto 12</p> </div> <div style="text-align: center;">  <p>Foto 13</p> </div> </div>
<p>Ir al inicio</p>	<p>Le gustó: --</p> <p>No le gustó: --</p> <p>Observaciones:</p> <p>Le ha sido fácil, desde la pantalla de audio, ha dado botón atrás en el menú de navegación de la parte superior (Foto 14) y de ahí ha encontrado el icono de "Main" en la Foto 15, sin problemas.</p> <div style="display: flex; justify-content: space-around; margin-top: 20px;"> <div style="text-align: center;">  <p>Foto 14</p> </div> <div style="text-align: center;">  <p>Foto 15</p> </div> </div>
<p>Escuchar audición</p>	<p>Le gustó:</p> <p>Sin problemas accede al main y desde allí pulsa "Audio guide", accediendo al listado y de ahí al audio de alguna parte del listado.</p> <p>No le gustó: --</p> <p>Observaciones:</p> <p>Al llegar a esta parte ya tiene más experiencia con la aplicación y accede fácilmente a escuchar la audición.</p>

Figura 22: Tabla Usuario 2. Test usabilidad competencia app "National Museum of Korea"

4.4.1. Conclusiones del test: Datos útiles para el diseño de la app.

Como conclusiones de las pruebas realizadas a las dos aplicaciones podemos obtener las siguientes:

a) **Lectura de un texto:**

- La interfaz clara con contraste en el texto fue agradecida por un usuario.
- Que la interfaz sea fácil de usar con apartados claros.
- La publicidad no gustó a varios usuarios.
- La falta de concreción del lugar donde se encuentra el usuario es otro aspecto a mejorar en la app “El Escorial”.
- Una mejora sería que en las audioguías apareciese el texto del audio sin tener que buscarlo, dos usuarios tuvieron problemas al encontrar el texto escrito ya que el usuario se esperaba encontrar un texto sobre la aplicación hablada o tuvo que indagar en varios menús hasta encontrarlo.
- No gusto el rebote de los textos de la pantalla inicial de la app “National Museum Of Korea” en adelante (NMK).

b) **Ir al inicio:**

- Algunos usuarios encuentran que en la aplicación de “El Escorial” faltan menús mejor desarrollados.
- Gustó el tener una barra de menús accesible para ir al inicio en la app NMK.
- A otro usuario no le gusto el menú permanente de la app NMK.
- El tener en el encabezado botón de vuelta atrás fue valorado positivamente.
- Uso excesivo del botón atrás de Android para la aplicación “El Escorial”, aspecto a mejorar.
- La falta de botón de inicio y menús no gusto a varios usuarios en la app “El Escorial”.
- En la app NMK en todo momento hay botones para salir.
- Un menú pop up no tenía botón de salir o cerrar.

c) **Escuchar audición:**

- Gustó el diseño de la barra de control de audio.
- Que se iniciase automáticamente la audición.
- El acceso fue rápido por todos los usuarios, se entendían bien los iconos de donde se encontraba la audición.

Todos estos datos los podremos aplicar para mejorar el diseño de nuestra aplicación posteriormente.

Como comentábamos el DCU tiene mucho trabajo, se interactúa mucho con los usuarios, no es una creación de software al modo clásico, sin embargo tiene sus ventajas y es que el diseño se hará pensando en los usuarios y acorde a sus gustos para una mejora de la usabilidad, sin dejar de lado en ningún momento los requisitos o especificaciones requeridas.

5. DISEÑO DE ESCENARIOS, FLUJOS DE INTERACCIÓN Y CASOS DE USO

5.1. Escenarios

De la misma forma que en la Ingeniería del Software el usuario en los casos de uso nos proporciona las funcionalidades de una aplicación, la definición de escenarios nos ayudará a obtener las funcionalidades que nuestro prototipo debe tener.

A continuación definiremos una serie de escenarios pero sin comentar ningún aspecto técnico ni tampoco la manera correcta de realizar la acción para conseguirlo.

5.1.1. Leer un texto sobre algún tema concreto de la aplicación

Luisa es una administrativa joven que ha ido de visita con su familia a visitar el Monasterio de El Escorial, pero antes quería informarse de que podría ver ya que la visita será corta, una mañana ya que por la tarde tendrán que volver al Hotel.

Por lo tanto, se baja una aplicación. Tiene interés en buscar en la aplicación información sobre la Biblioteca ya que tiene entendido posee unos volúmenes y libros interesantes. Por ello accede a la aplicación y busca algún botón o menú que pueda indicarle lo que busca.

Ve que al acceder a la aplicación hay un botón de "Biblioteca". Accede a la información y se encuentra una fotografía de la sala junto con un texto para poder leerlo. También ve que hay posibilidad de escucharlo ya que ve botones de control de audio por lo que los pulsa y va escuchando la grabación en el móvil con sus auriculares mientras va caminando para tomar el autobús a casa desde el trabajo.

5.1.2. Ir al inicio de la aplicación desde la pantalla donde se encuentre

Alumno: Jesús Villegas Guerra

Gabriel es un pequeño de 10 años que le ha tomado el móvil a su madre, Luisa. Está jugando con el mismo y ve que hay una aplicación abierta, es la del Monasterio de El Escorial. El joven sabe que el fin de semana próximo va a ir a Madrid a visitarlo por lo que desde la página que estaba su madre quiere acceder a la pantalla inicial para ver como comienza la aplicación y ver que posee. Por ello busca en la pantalla algún lugar que le indique ir al inicio y encuentra que hay un botón de “home” o más concretamente una casita que le indica que va al inicio. Conoce este símbolo porque en el colegio ya ha experimentado con páginas web en clase de informática y conoce el icono de inicio.

5.1.3. Acceder al texto de Panteón de las Infantas

Jaime es un padre de familia que va a visitar a el R.M. de San Lorenzo de El Escorial, tiene una app instalada en su móvil para utilizarla y le atraía conocer quiénes están en el panteón de las infantas y porqué.

Para acceder ve que hay un botón que indica panteones por lo que piensa que puede estar ahí. Al abrirlo se encuentra con un listado que le indica los dos tipos de panteones que hay el de las infantas y el de los reyes. Por ello accede desde el listado al panteón de las infantas y se le muestra una fotografía con el texto aunque también se da cuenta de que hay la posibilidad de escucharlo en audio.

5.1.4. Escuchar una audición y pararla.

María una chica joven que va con su pareja están visitando el R.M. de El Escorial por lo que se han comprado un adaptador para ir escuchando las audiciones por la aplicación del móvil. Con el adaptador pueden escuchar la audición dos personas con lo que es ideal ir compartiendo la experiencia con su pareja.

En ese momento están accediendo a la Casa del Rey por lo que se ponen los dos los auriculares , pulsa el botón de Casa del Rey y a continuación el botón de “play” o comenzar la audición.

Cuando comienza la audición ven que el total de minutos que va a durar será de tres y como van los segundos pasando conforme escucha la grabación, mientras la pareja va recorriendo la sala y parándose en los detalles que la audición les va transmitiendo.

5.1.5. Buscar el teléfono del Monasterio y llamar.

Carlos acaba de llegar al Hotel en Madrid de vacaciones con su pareja, son las 12.30 de la mañana. En el viaje en AVE a Madrid encontró una aplicación que se bajó sobre el R.M. de El Escorial que visitarán el día próximo a su llegada ya que en ese día por la tarde visitarán el Museo del Prado.

Aunque en la aplicación ha accedido a ver los horarios, quiere estar seguro de que puede visitar al día siguiente el Monasterio ya que puede que sea fiesta o ser el día que el Monasterio cierra al público y desde Madrid se tarda en llegar en coche una hora por lo que si no pudieran verlo cambiarían de planes.

Por ello ve que justo debajo de los horarios hay un botón que indica en el Smartphone que puede llamar al Monasterio.

Pulsa el botón de llamar desde la aplicación y en su teléfono le aparece la pantalla de llamadas, por lo que al pulsar el icono de llamar realiza la llamada y le atienden por teléfono indicándole que el día de mañana estará abierto el Monasterio.

5.2. Flujos de interacción.

A continuación aplicaré el diagrama de Garret, J.J. para realizar los flujos de interacción de los requisitos del sistema. Expondré algunos de los requisitos más importantes.

5.2.1. Leer un texto sobre algún tema concreto de la aplicación

El usuario desde la página de inicio puede acceder a cualquiera de las páginas que aparecerán en ella, dando lugar a poder leer el texto elegido.

Desde el inicio se puede entrar en páginas que tienen un solo acceso y otras que tienen más de uno como "Panteones" y "Más información".

Al entrar desde la página de "Inicio" en cualquiera de las páginas del área "Inicio" habrá un script llamado A) que indica que si el "Pop up" ya se había realizado ya no se mostrará más. El pop up solo se mostrará una vez, de tal forma que si ya se había mostrado se accede directamente a la página que se quería leer.

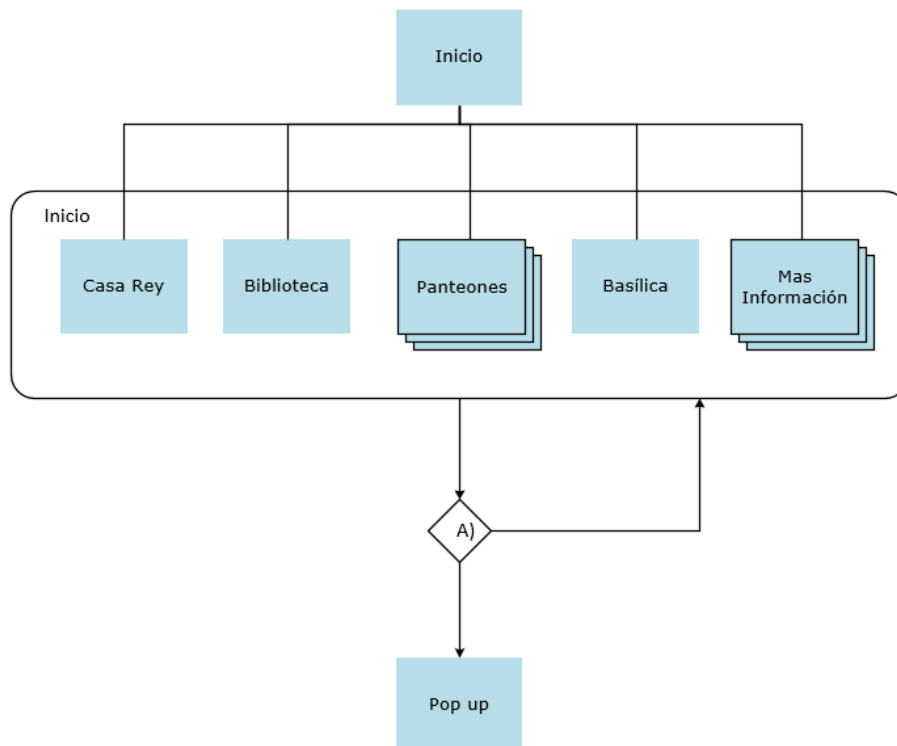


Figura 23: Flujo de interacción. Diagrama leer un texto.

El apartado A) se hará por medio de un script.

5.2.2. Acceder los distintos listados de la aplicación.

En este caso estaríamos en la misma situación que el apartado anterior, sin embargo solamente referido a las páginas “Panteones”, “Misterios” y “Más Información”.

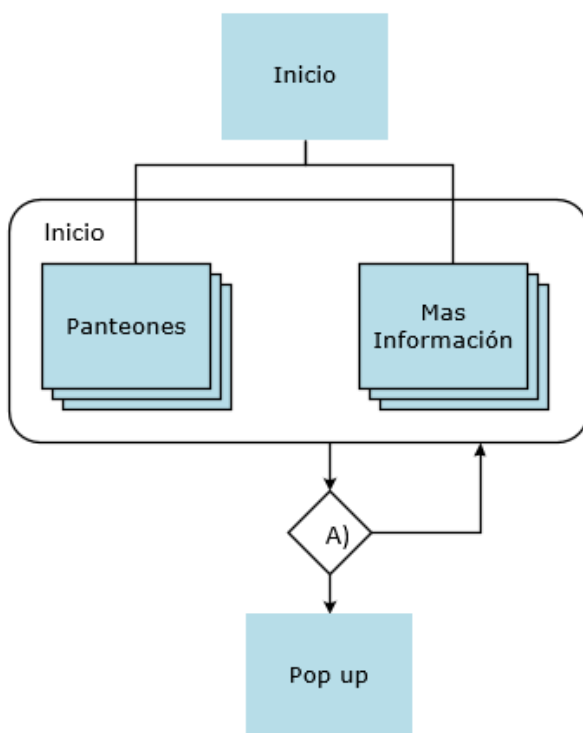


Figura 24: Flujo de interacción. Acceder a listados.

5.2.3. Escuchar una audición.

Al entrar en las páginas “Casa Rey”, “Biblioteca”, “Basílica” o al acceder a algunos de los listados de “Panteones” y “Más Información”, se podrá dar la opción de escuchar un texto. El usuario tendrá las opciones que se indican con el apartado B), es decir, elegirá comenzar, pausar o parar. En el caso de que se elija pausar o parar como primera opción la aplicación no hará nada, pero si se elige primero comenzar a continuación se tendrán las opciones del apartado c), es decir, para o pausar la audición y desde alguno de esas opciones volver a comenzar bien por donde lo dejo con el botón de pausar o comenzando desde el principio al haber pulsado con anterioridad el botón parar.

Desde cualquiera de las páginas donde se encuentre el usuario, se podrá elegir de un menú el poder ir a otros lugares. Viendo el apartado D) puede ocurrir que haya una audición en curso o no. Si no hay una audición en curso directamente no ocurre ninguna acción más que elegir una opción de entre la del menú. Pero si hay una audición en curso pasamos al apartado E) en que

Alumno: Jesús Villegas Guerra

simultáneamente ocurren dos cosas al pulsar una opción del menú: Por un lado se elegirá una opción del menú y por otro se parará la audición al pasar a otra página.

Además desde cualquiera de las páginas se tendrá acceso a ir al inicio.

Los apartados B),C),D) y E) se realizarán mediante scripts en la aplicación.

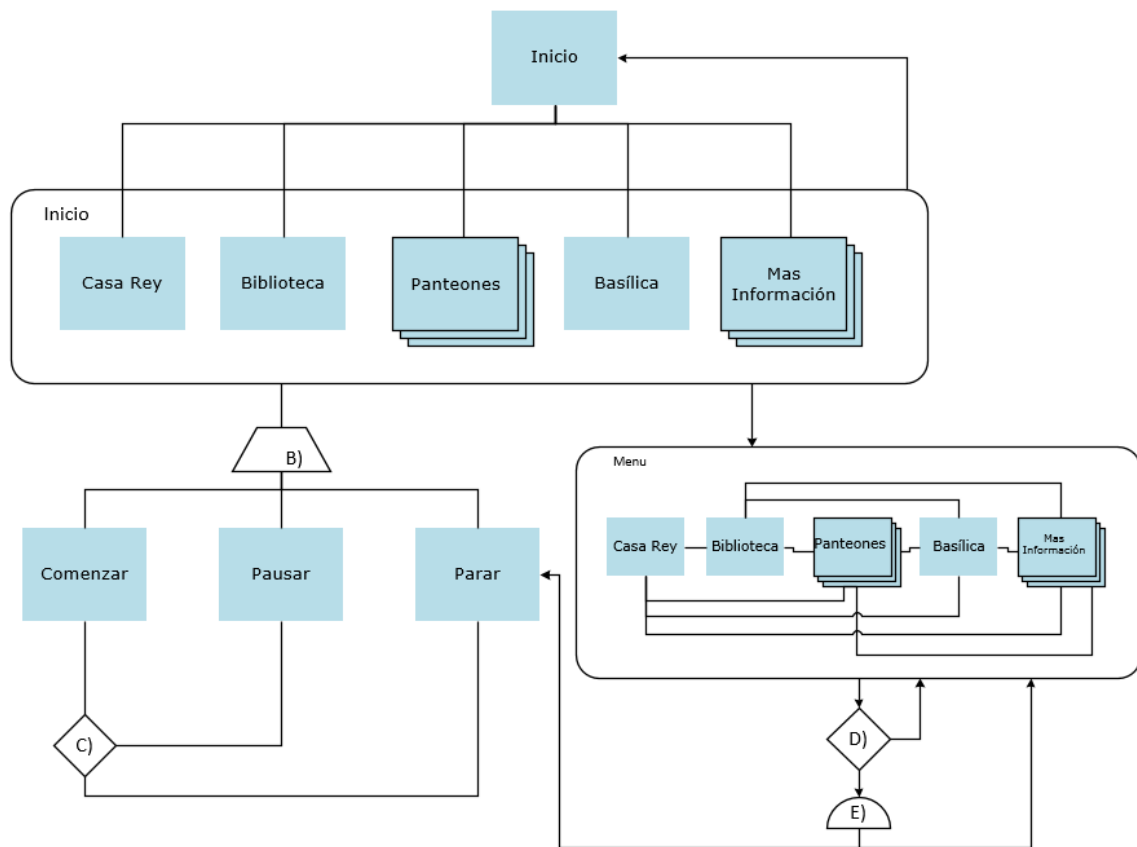


Figura 25: Flujo de interacción. Escuchar audición.

5.3. Casos de uso: Especificación textual y diagrama

En nuestro caso el actor implicado en los casos de uso será el usuario del Smartphone, según vimos en los perfiles de usuarios.

Alumno: Jesús Villegas Guerra

A continuación mostraré en forma de diagrama los casos de uso posibles así como la especificación textual del diagrama en subapartados.

El diagrama de casos de uso sería el siguiente:

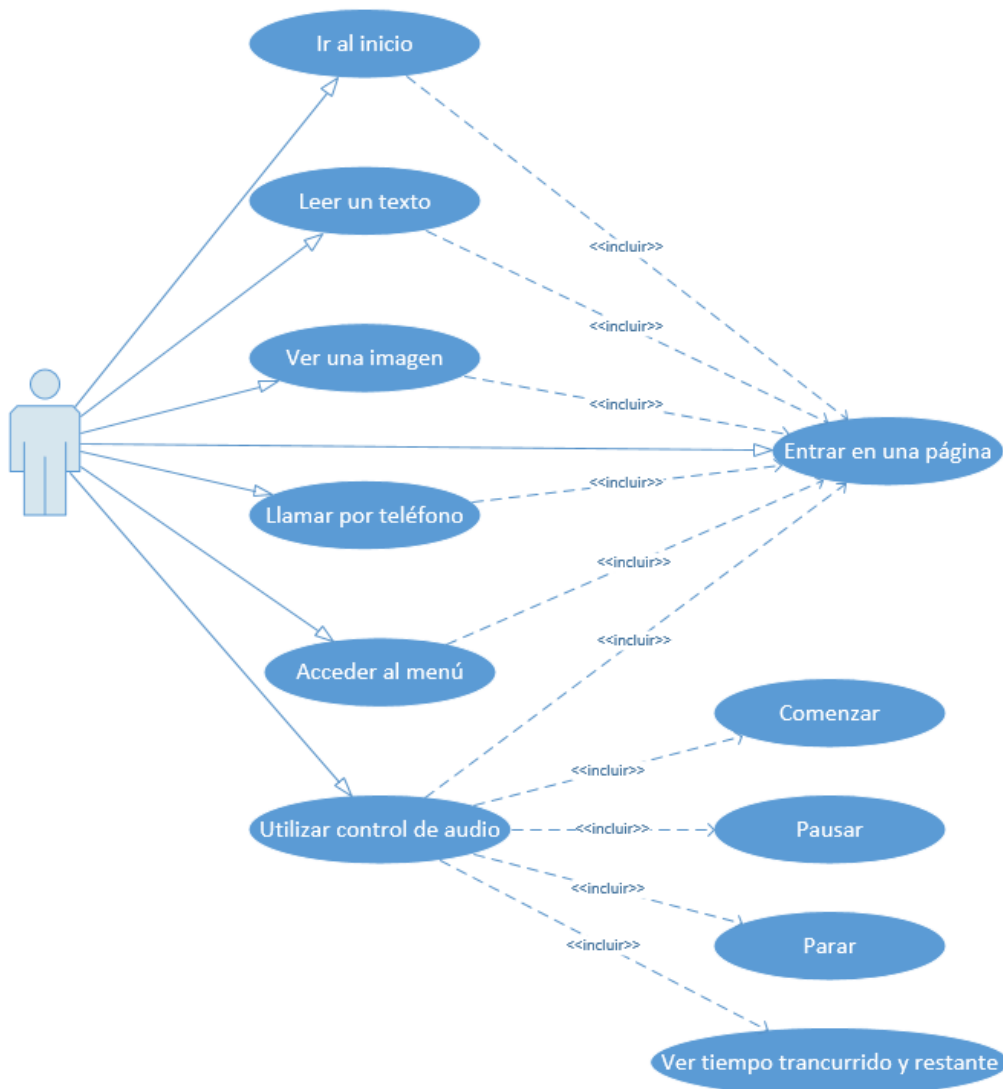


Figura 26: Diagrama: Casos de uso

Alumno: Jesús Villegas Guerra

5.3.1. Ir al inicio

Descripción
Permite ir al inicio de la aplicación.
Actor
Usuario.
Casos de uso relacionados
Entrar en una página.
Precondición
Es necesario haber entrado en una página previamente
Postcondición
Va al inicio de la aplicación.
Disparador
El usuario pulsa el botón de "Inicio"
Flujo básico
El usuario pulsa el botón situado en la cabecera de la aplicación para ir al inicio de la aplicación. El botón estará presente en todas las páginas y de fácil acceso en la parte superior derecha.
Flujo alternativo
--
Observaciones
--
Papel en el trabajo de los usuarios
Facilitar al usuario un punto de partida en la aplicación y que se puedan situar en caso de que no sepan dónde están exactamente.

Figura 27: Tabla Caso de uso. Ir a inicio.

5.3.2. Leer un texto

Descripción
El usuario podrá leer un texto que aparezca en la página elegida.
Actor
Usuario.

Casos de uso relacionados
Entrar en una página.
Precondición
Es necesario haber entrado en una página previamente.
Postcondición
Poder leer el texto de la página elegida.
Disparador
El usuario pulsa una opción del menú para elegir una página concreta.
Flujo básico
El usuario bien desde el inicio o bien desde el menú accesible que hay en cualquier página elige que página desea ver. Una vez dentro podrá leer su contenido.
Flujo alternativo
Puede ocurrir que al elegir una página no se muestre un texto para leer sino un listado con el que podrá elegir que opción le es más interesante y desde ahí pasar a leer un texto.
Observaciones
--
Papel en el trabajo de los usuarios
Es la forma en que un usuario puede leer un texto que elija.

Figura 28: Tabla Caso de uso. Leer un texto

5.3.3. Ver una imagen

Descripción
El usuario podrá ver la imagen que aparezca en la página elegida.
Actor
Usuario.
Casos de uso relacionados
Entrar en una página
Precondición
Es necesario haber entrado en una página previamente.
Postcondición
Podrá ver la imagen

Disparador
El usuario pulsa una opción del menú para elegir una página concreta.
Flujo básico
El usuario bien desde el inicio o bien desde el menú accesible que hay en cualquier página elige que página desea ver. Una vez dentro podrá ver la imagen que contenga.
Flujo alternativo
--
Observaciones
Habrán páginas que puede no contengan imágenes.
Papel en el trabajo de los usuarios
Poder darles una idea de lo que se está explicando en el texto o el lugar que están visitando.

Figura 29: Tabla Caso de uso. Ver una imagen.

5.3.4. Llamar por teléfono

Descripción
Poder llamar por teléfono directamente desde la aplicación al Real Monasterio de El Escorial en caso de necesitar alguna consulta.
Actor
Usuario.
Casos de uso relacionados
Entrar en una página.
Precondición
Haber entrado en Más información>Horarios-Precios-Telf.
Postcondición
Se abrirá la pantalla de llamadas del teléfono móvil con el número al que hay que llamar.
Disparador
Pulsar el botón o botones, si tuviese más de un teléfono, pudiendo elegir el teléfono al que se va a llamar.
Flujo básico
El usuario verá el número de teléfono y podrá pulsar un botón para llamar al mismo,

pasándose a continuación a la pantalla de llamadas del Smartphone y poder realizar de esa forma la llamada pulsando "llamar".
Flujo alternativo
--
Observaciones
--
Papel en el trabajo de los usuarios
Importante opción que integra nuestra aplicación con las funcionalidades del teléfono móvil.

Figura 30: Caso de uso. Llamar por teléfono

5.3.5. Acceder al menú

Descripción
Desde cualquier página en la que el usuario se encuentre, se podrá acceder a un menú.
Actor
Usuario
Casos de uso relacionados
Entrar en una página.
Precondición
Es necesario haber entrado en una página previamente.
Postcondición
Se inicia la página que se ha elegido.
Disparador
El usuario pulsa sobre un botón del menú.
Flujo básico
El usuario al pulsar un botón del menú situado en la parte inferior le traslada a la página elegida.
Flujo alternativo
Hay posibilidad de hacer desaparecer el menú inferior de la pantalla si se pulsa sobre algún lugar que no sean botones en la pantalla, incluido el encabezado. Cada vez que se pulsa en algún lugar de la pantalla aparece y desaparece el menú de opciones.

Alumno: Jesús Villegas Guerra

Observaciones
--
Papel en el trabajo de los usuarios
Da la posibilidad de ir navegando entre diferentes paginas.

Figura 31: Caso de uso. Acceder al menú.

5.3.6. Utilizar el control de audio

Descripción
El usuario podrá escuchar una audición, pararla o pausarla.
Actor
Usuario.
Casos de uso relacionados
Entrar en una página, comenzar, pausar, parar.
Precondición
Haber entrado en una página con las opciones de control de audio.
Postcondición
El usuario podrá escuchar una audición o pararla y pausarla si ya estaba en marcha.
Disparador
El usuario pulsa sobre alguno de los botones del control de audio.
Flujo básico
Se podrá pulsar sobre cualquiera de los botones de comenzar, parar o pausar en el control de audio para escuchar, parar o pausar una audición.
Flujo alternativo
Automáticamente se parará la audición al salir de la página si en ese momento había una en marcha debido a que el usuario ha elegido ver otra página.
Observaciones
--
Papel en el trabajo de los usuarios
Da la opción a los usuarios a poder escuchar el texto mientras recorren la sala donde están ubicados o bien escuchar el texto sin necesidad de leerlo. Con esta opción se atiende a la diversidad en las personas, sobre todo oculares o con falta de visión.

Figura 32: Caso de uso. Utilizar control de audio.

5.3.7. Entrar en una página

Descripción
Da la opción de navegar por las distintas páginas que hay en la aplicación.
Actor
Usuario.
Casos de uso relacionados
--
Precondición
Haber abierto la aplicación.
Postcondición
Se estará dentro de una página.
Disparador
El usuario al abrir la aplicación ya entra en una página o bien al pulsar cualquier botón del menú.
Flujo básico
El usuario abre la aplicación y entra en una página con opciones para poder elegir.
Flujo alternativo
--
Observaciones
--
Papel en el trabajo de los usuarios
Es básico que los enlaces entre páginas funcionen y que éstos lleven al usuario hacia el lugar indicado en ellos.

Figura 33: Caso de uso. Entrar en una página.

6. PROTOTIPO

El prototipo que muestro a continuación ha sido creado en Axure RP en alta definición. En el prototipo no se han creado los logos ni se han incluido imágenes, esta parte se ha planificado para la implementación, pero se han sentado las bases del diseño que se producirá.

Se ha tratado de hacer un prototipo lo más realista posible de la aplicación para ser presentado a los usuarios en sus pruebas. El prototipo, como se verá más adelante, es

Alumno: Jesús Villegas Guerra

totalmente funcional, menos en lo referente a los controles de audio, aun así si se pulsan sobre ellos aparecen indicaciones de lo que realizan.

El prototipo está basado en una pantalla de teléfono de 3,2", aunque en las especificaciones iniciales se comentaba que se adaptase a otros formatos como 4", 5", y tablets, solamente he realizado el diseño para su prueba en este formato. En la implementación se hará pensando en todas las posibilidades aplicando css y media queries.

El prototipo funciona en cualquier navegador en HTML. La imagen que aparece a continuación es la pantalla inicial del prototipo tal como sale en el navegador Firefox. A la izquierda de la imagen se puede ver el mapa del sitio de la aplicación divida en directorios y subdirectorios y jerarquizados tal como se presenta en la navegación a través de las páginas.

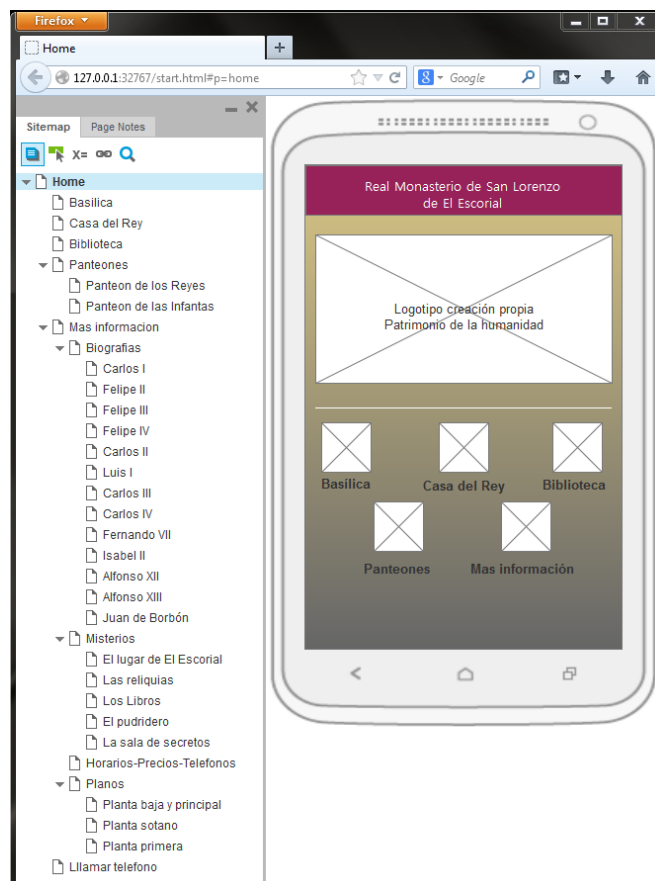


Figura 34: Prototipo. Mapa de la aplicación. Axure RP

Alumno: Jesús Villegas Guerra

Para ver el prototipo seguiremos lo que un usuario puede hacer recorriendo el mapa de la aplicación.

Desde la pantalla de inicio se puede ir a Basílica, Casa del Rey, Biblioteca, Panteones y Más información. Los cuadrados azules muestran acciones que se realizan al pulsar sobre el objeto. Por ejemplo al pulsar sobre el objeto "1" la interacción que se realiza es:

- 1 : OnClick Open Basilica in Current Window, según se ha programado en Axure RP.

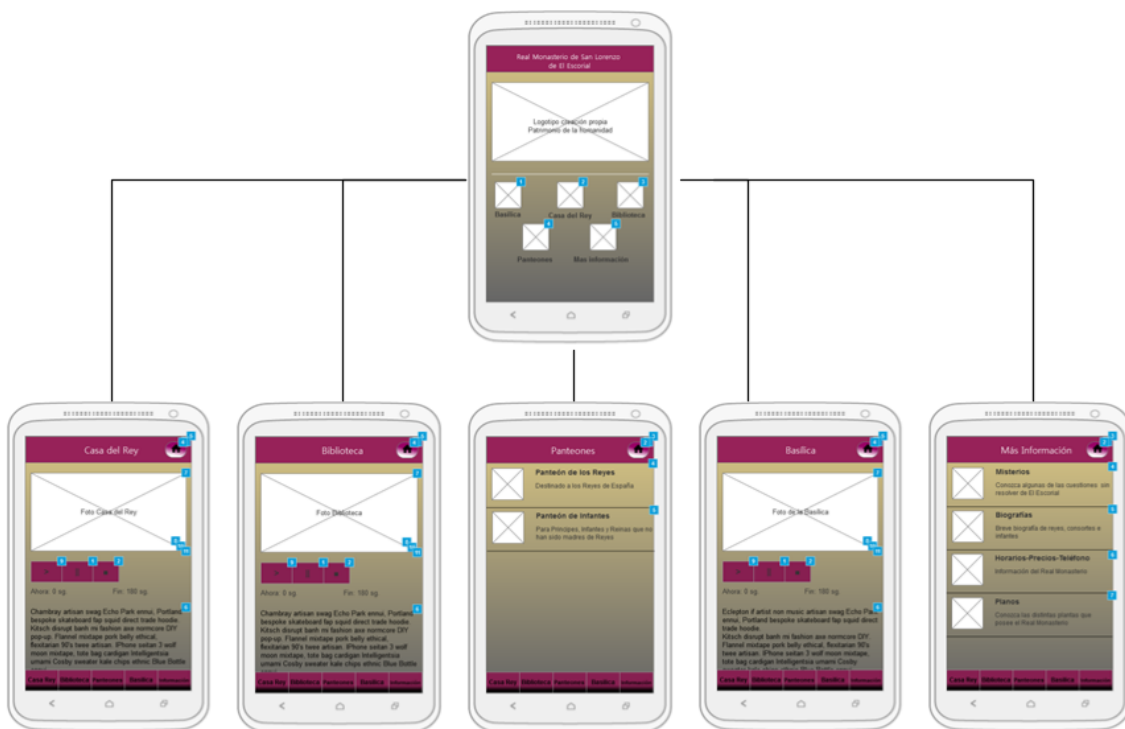


Figura 35: Prototipo. Página Inicio

A continuación se muestra que ocurre al pulsar sobre el botón de "comenzar" audición en la pantalla "Casa del Rey" en la que sale un popup que dura unos segundos, nos indica que se ha pulsado sobre el audio y que aún no está realizado.

Alumno: Jesús Villegas Guerra

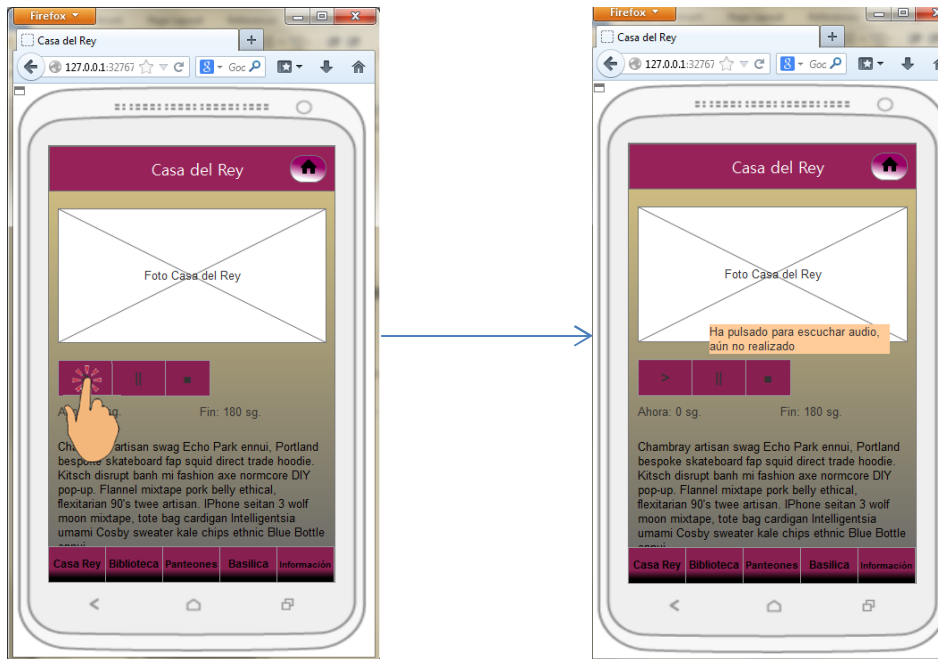


Figura 36: Prototipo. Pulsar "Comenzar".Pop Up

Seguimos con "Panteones" con los enlaces siguientes:

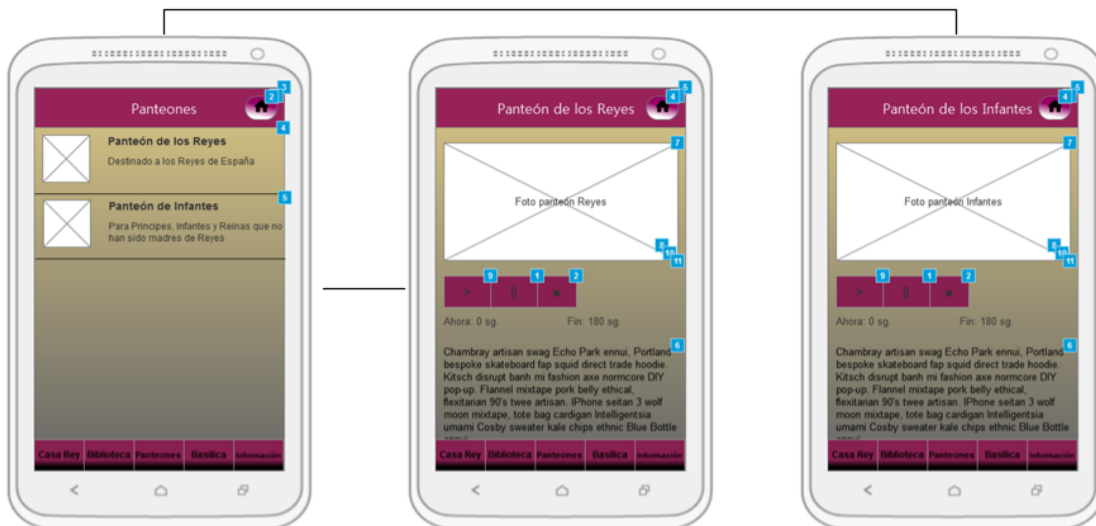


Figura 37: Prototipo. Panteones

Alumno: Jesús Villegas Guerra

A continuación podemos ver los enlaces que tiene la página “Más Información”:

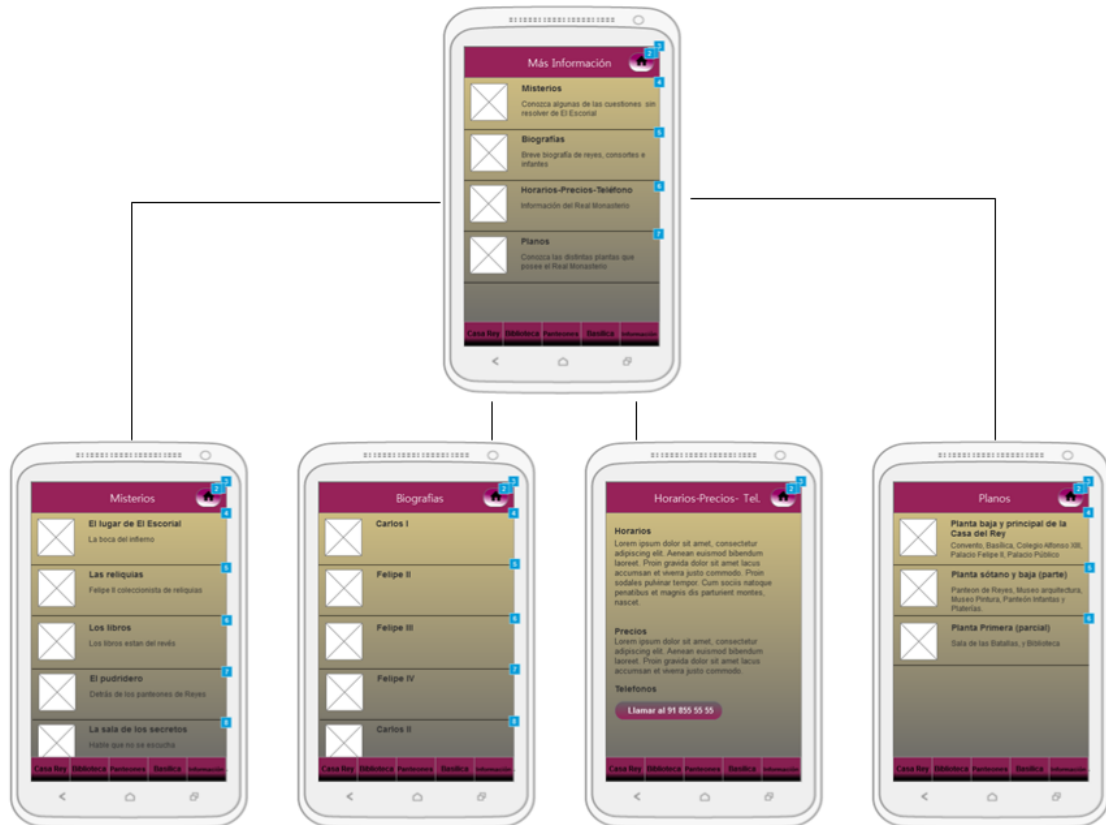


Figura 38: Prototipo. Más información

Continuamos viendo los las páginas a las que se tiene acceso desde “Biografías”, en este caso no se han puesto todas ya que la lista es extensa:

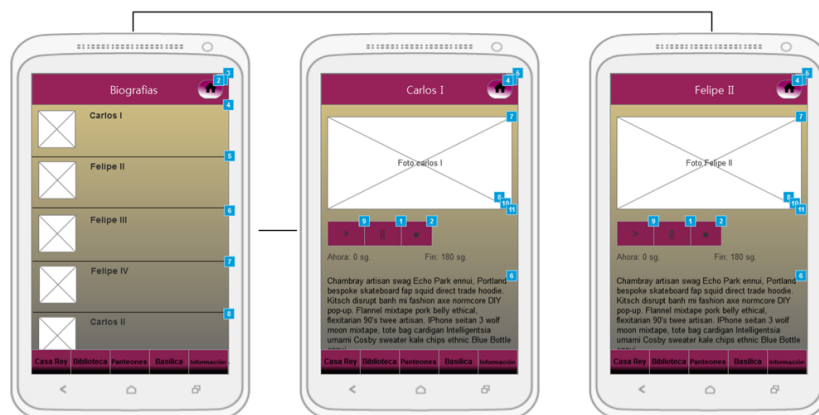


Figura 39: Prototipo. Biografías.

Alumno: Jesús Villegas Guerra

Desde la pantalla de “Misterios” se accede a varias páginas de la lista:

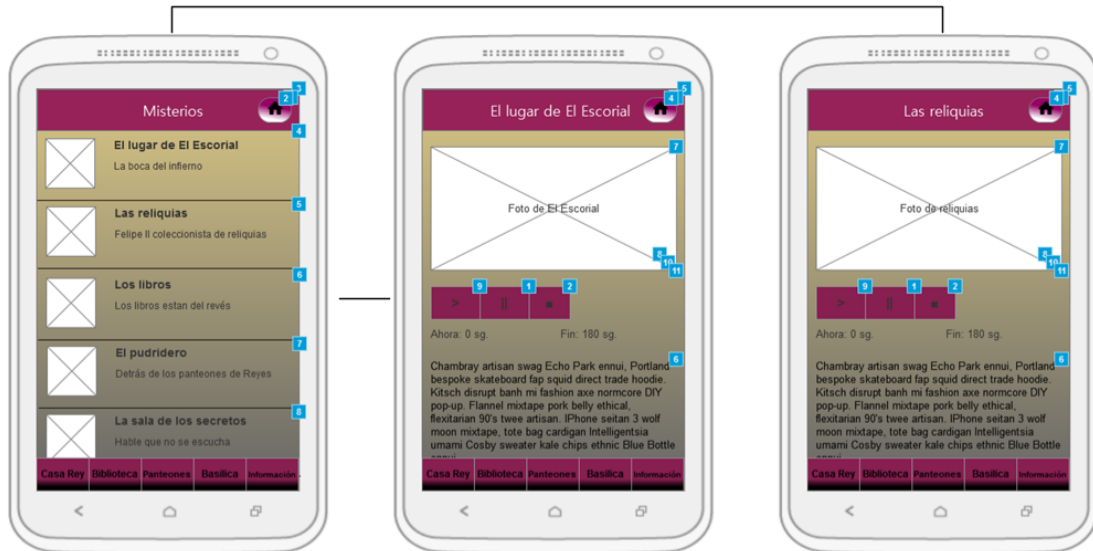


Figura 40: Prototipo. Misterios

La página “Planos” nos muestra una lista y sus enlaces a las páginas:

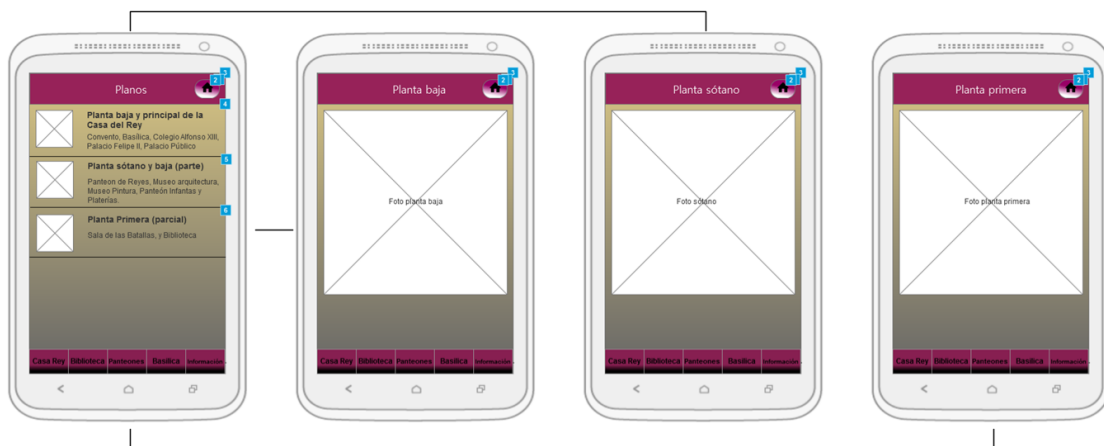


Figura 41: Prototipo. Planos.

Finalmente vamos a ver una pantalla interesante ya que desde la propia aplicación se puede usar las funcionalidades del propio teléfono usando las nuevas etiquetas de HTML5, en este caso se puede hacer una llamada pulsando el botón de llamar a un teléfono. En la parte de implementación veremos cómo se representa esta etiqueta:



Figura 42: Prototipo. Llamar por teléfono.

7. EVALUACIÓN

Se va a evaluar el prototipo creado antes de realizar el diseño real de la aplicación, para conocer cómo se comportan los usuarios con el diseño.

Se repartirá una hoja a los usuarios (ver anexo) que van a realizar la prueba en la que se indica las acciones que tienen que realizar. En ningún momento en la prueba la persona que observa puede interactuar con los usuarios, solo hacer preguntas para obtener información adicional.

Las pruebas que se realizarán a los usuarios son las siguientes:

- Leer el texto de la página de Biblioteca.
- Escuchar y parar la audición de la página Panteón de los Reyes.
- Ir al inicio de la aplicación.
- Buscar el teléfono del R.M. y llamar.

Usuario 1	
Prototipo	
Leer el texto de la página de la Biblioteca	Le gustó: Le gustó el diseño.
	No le gustó: Que no aparezcan fotos o logotipos en la página.
	Observaciones: Accede correctamente a la página de la Biblioteca
Escuchar y parar la audición de la página Panteón de los Reyes	Le gustó: --
	No le gustó: El menú en la parte inferior. Le gusta más el menú como aparecen en otras aplicaciones en la parte superior y que se despliegue.
	Observaciones: Conoce el botón de “comenzar” y lo pulsa. Accede correctamente a la página pulsando el botón de “Panteones” desde el menú de la parte inferior.
Ir a inicio	Le gustó: --
	No le gustó: --
	Observaciones: Reconoce el botón “Home” y accede al inicio.
Buscar el teléfono del R.M. y llamar	Le gustó: --
	No le gustó: --
	Observaciones: Observa dónde puede estar la información que busca y pulsa más información. A continuación ve en el listado “Horarios-Precio-Tel” y lo pulsa, accediendo y pulsando el botón de “Llamar”.

Figura 43: Tabla usuario 1. Evaluación prototipo.

Usuario 2	
Prototipo	
Leer el texto de la página de la Biblioteca	Le gustó: --
	No le gustó: Nos sugiere que el control de audio tuviese más diseño.
	Observaciones: Accede correctamente a la página de la Biblioteca.
Escuchar y parar la audición de la página Panteón de los Reyes	Le gustó: --
	No le gustó: --
	Observaciones: Accede sin problemas a la página buscada y pulsa el botón de "Comenzar".
Ir a inicio	Le gustó: --
	No le gustó: Nos indica que el botón puede contener además de la imagen un texto y que además se podría poner un botón de "Back" (volver) en la parte superior.
	Observaciones: El usuario está acostumbrado a utilizar iPhone y aplicaciones iPhone que poseen el botón "Back" en la mayoría de ellas.
Buscar el teléfono del R.M. y llamar	Le gustó: --
	No le gustó: --
	Observaciones: Busca y accede correctamente a "Mas información" y desde ahí a llamar.

Figura 44: Tabla usuario 2. Evaluación prototipo.

Usuario 3	
Prototipo	
Leer el texto de la página de la Biblioteca	Le gustó:
	No le gustó:
	Observaciones:
Escuchar y parar la audición de la página Panteón de los Reyes	Le gustó: --
	No le gustó:--
	Observaciones:
Ir a inicio	Le gustó:
	No le gustó:
	Observaciones:
Buscar el teléfono del R.M. y llamar	Le gustó:
	No le gustó:
	Observaciones:

Figura 45: Tabla usuario 3. Evaluación prototipo

7.1. Conclusiones a la evaluación.

Como conclusiones podemos obtener lo siguiente:

- En la mayoría de los casos los usuarios se manejan bien en la aplicación, encuentran lo que buscan y la navegación ha sido fluida y rápida entre los menús.

Alumno: Jesús Villegas Guerra

- Algún usuario nos indica que el menú le gusta que aparezca en la parte superior.
- También se ha indicado que el control de audio tuviese más diseño.
- Un usuario ha comentado que el diseño es agradable.
- Se ha comentado que el botón de “Inicio” tuviese texto además del icono.
- Se nos ha sugerido poner un botón de “volver”, aunque el usuario que lo ha recomendado es un usuario de iPhone y las pruebas están haciendo para un Smartphone Android.

Podemos generalizar por tanto que no ha habido problemas graves de usabilidad de la aplicación y se han tenido en cuenta las observaciones de los usuarios previos al diseño y se tendrán en cuenta también las consideraciones de los usuarios en la implementación.

Desde aquí vemos como el DCU es muy importante en el desarrollo del software y cómo a la larga nos ahorra tiempo y costes ya que diseñamos pensando en el usuario, mejorando su experiencia y usabilidad sin perder de vista las especificaciones descritas en este caso para nuestra aplicación pero si fuese para un cliente tampoco se perdería el objetivo de lograr implementar todas las especificaciones previstas.

Conseguimos por tanto con un poco de más trabajo un prototipo más completo y usable que nos hará llegar a la etapa de implementación con un trabajo de testeo prácticamente realizado, ya que es un método iterativo de trabajo y que nos ahorrará costes en tiempo de implementación al conocer que desean y cómo quieren que se comporte la interfaz los usuarios, uno de los aspectos más olvidados por los programadores que se centran sobre todo en las especificaciones sin tener en cuenta al usuario, con lo que al final de la implementación si los usuarios no están de acuerdo con el diseño, interfaz, usabilidad, etc, los costes y el tiempo se disparan con el riesgo que conlleva el tener que realizar una nueva programación de muchos de los aspectos no testeados con anterioridad. Este comentario que acabo de hacer, por desgracia, lo vivo constantemente en mi trabajo diario, donde el cliente no comprende que se tenga que “malgastar” (según palabras del propio cliente) tiempo en “pensar primero, hacer después, o más específicamente diseñar primero, programar después” (Mordecki, D. 2004. Pensar Primero, pg. 44).

8. IMPLEMENTACIÓN

8.1. Creación del proyecto

Para crear el proyecto utilizo la consola de Windows 7. Entramos en la carpeta de la API de Phonegap para crear el proyecto hasta el lugar donde está el ejecutable en la carpeta "bin" para comenzar.

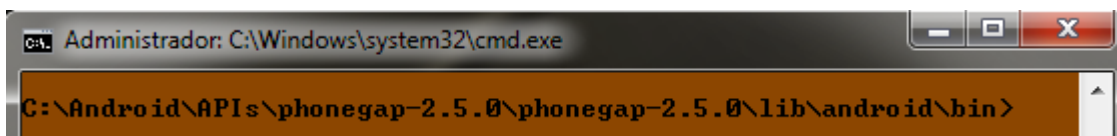


Figura 46: Creación proyecto. Carpeta bin phonegap

Una vez en esta carpeta con el comando "create" primero le decimos la ruta donde queremos que ponga el proyecto que va a crear (C:\Android\WorkSpaceTFC\TFC), a continuación le decimos el nombre del paquete (edu.uoc.tfc) y por último el nombre del proyecto que utilizaremos en Eclipse (TrabajoFinCarrera).

Con esto al pulsar "enter" se comenzará a crear el proyecto Android con Phonegap 2.5.0, creando una estructura que después utilizaremos en Eclipse.

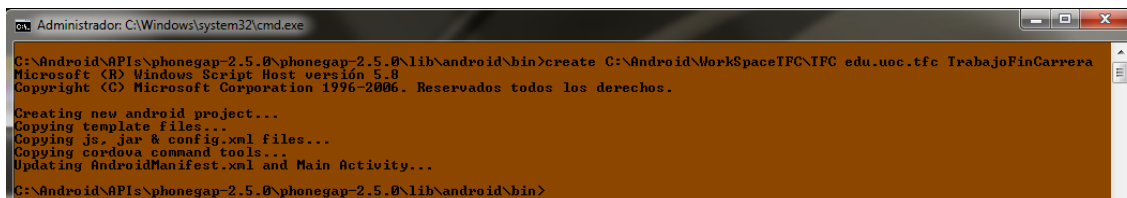


Figura 47: Creación proyecto. Proyecto creado.

En Eclipse elegimos File>New>Other>Android>Android Project from Existing Code, es decir, tomamos el código que se ha generado desde consola para comenzar nuestra aplicación, como puede observarse en la figura 48.

Sin embargo como se puede ver en la figura 49 nos crea dentro assets\ una carpeta "www" de Android que no tiene la estructura que deseamos pues hay muchas carpetas y archivos que no nos van a hacer falta.

Alumno: Jesús Villegas Guerra

Por tanto, cambiamos el assets\www que tenía el proyecto creado por el nuestro. Nuestro “www” está configurado para servir como base de una nueva aplicación. Esta base es la que he ido utilizando en mi planificación Gantt que llamé “pruebas unitarias con Eclipse” para ir probando unitariamente elementos que he ido incorporando a la aplicación.

En la imagen de la derecha de la figura 49 puede verse la diferencia que hay entre el assets\www inicial y al que hemos cambiado.

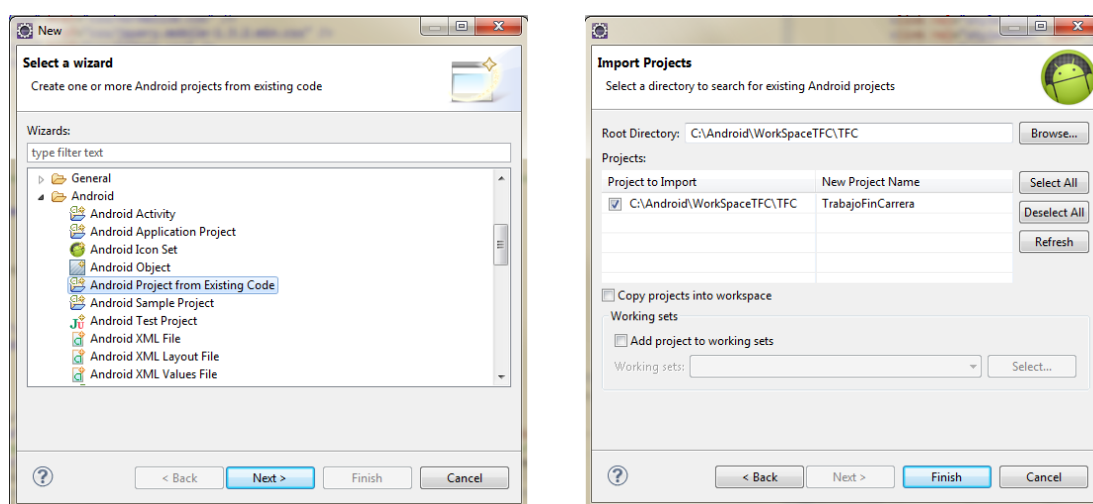


Figura 48: Creación proyecto. Eclipse

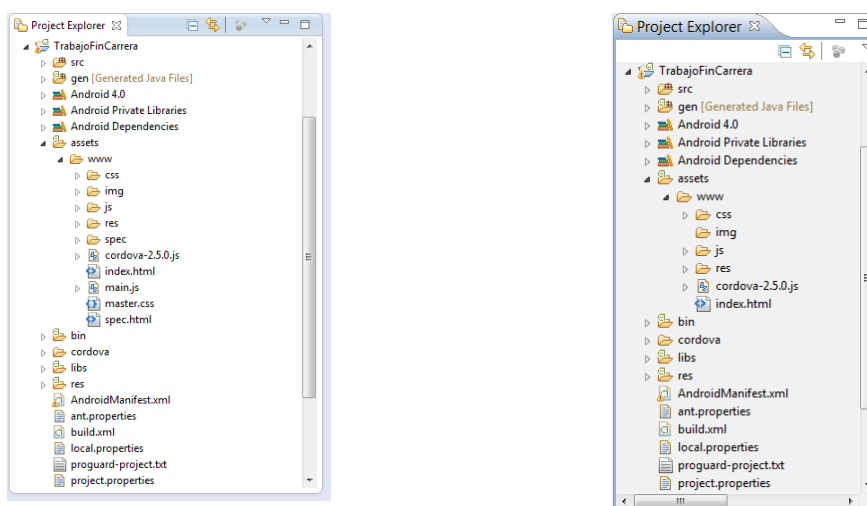


Figura 49: Creación proyecto. Carpeta “assets\www”.

A continuación muestro la estructura inicial que tiene la página "index.html":

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta name="format-detection" content="telephone=no" />
    <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1,
    width=device-width, height=device-height, target-densitydpi=device-dpi" />
    <link rel="stylesheet" type="text/css" href="css/normalize.css" />
    <link rel="stylesheet" type="text/css" href="css/jquery.mobile-1.3.2.min.css" />
    <!-- Aquí introducimos el theme-roller -->
    <link rel="stylesheet" type="text/css" href="css/app.css" />
    <title>Proyecto Fin de Carrera</title>
  </head>
  <body>
    <!-- Plantilla de página***** -->
    <div id="" data-role="page">
      <div data-role="header">
        <h1>Proyecto Fin Carrera</h1>
      </div>
      <div data-role="content">
        <p>Mi contenido</p>
      </div>
      <div data-role="footer">
        <h4>@Jesús Villegas Guerra ITIG</h4>
      </div>
    </div>
    <!-- Scripts de la aplicación ***** -->
    <script type="text/javascript" src="js/jquery.js"></script>
    <script type="text/javascript" src="js/jquery.mobile-1.3.2.min.js"></script>
    <script type="text/javascript" src="cordova-2.5.0.js"></script>
  </body>
</html>
```

Figura 50: Creación proyecto. Estructura inicial "index.html"

8.2. Estructura inicial de la página index.html

Iniciar una página en HTML con HTML5 ahora es más sencillo que nunca, solamente debemos indicar al navegador con:

```
<!DOCTYPE html>
```

Figura 51: Estructura index.html. Doctype

<!DOCTYPE html> no debe ser considerado como etiqueta HTML, sino como una declaración del lenguaje que se está utilizando y que le servirá al navegador para conocer lo que a continuación se espera.

8.2.1. Head

En el encabezado tenemos varias etiquetas interesantes que debemos comentar. Decir que algunas etiquetas con las nuevas innovaciones en HTML5 pueden ser más reducidas como por ejemplo poner <link rel="stylesheet" href="css"/> en vez de la antigua etiqueta <link type="text/css" rel="stylesheet" href="css"/>, sin embargo como este código se ha generado

Alumno: Jesús Villegas Guerra

automáticamente al crear el proyecto desde consola, lo vamos a dejar de esta forma. Se puede decir por tanto que tenemos una forma corta con HTML5 y una más larga antes del HTML5 (Freeman, E & Robson, E, 2011. Head First HTML5 Programming).

Otra de las etiquetas interesantes es la siguiente:

```
<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height, target-densitydpi=device-dpi" />
```

Figura 52: Estructura index.html. Etiqueta viewport.

Con la etiqueta que aparece en la figura 52 lo que hacemos es que el contenido de la página se adapte al ancho y alto del dispositivo en el que se está visualizando.

Hemos incluido en el Head referencia a los estilos que contendrá nuestra página. Conforme vayamos haciendo la aplicación iremos introduciendo nuevos estilos CSS o modificando los actuales para adaptarlos a nuestras necesidades.

En un comienzo podemos ver que según se muestra en la figura 53 tenemos referencias a varios CSS que usaremos.

```
<link rel="stylesheet" type="text/css" href="css/normalize.css" />  
<link rel="stylesheet" type="text/css" href="css/jquery.mobile-1.3.2.min.css" />  
<!-- Aqui introducimos el theme-roller -->  
<link rel="stylesheet" type="text/css" href="css/app.css" />
```

Figura 53: Estructura index.html. Css

Por un lado hemos incluido en nuestra carpeta CSS el archivo [normalize.css](#) que nos permite lograr una normalización de estilos que vienen definidos por defecto en cada uno de los navegadores web.

Cada navegador tiene algo así como su hoja de estilo propia, de forma que cuando no tenemos definida una etiqueta, se le asigna un valor (diferente según el navegador) el resultado de todo esto, experiencias de usuarios distintas frente a un mismo proyecto web.

Normalize.css es un recurso para desarrolladores front-end con el que ayudar a cualquier interesado a que su sitio web se vea igual (casi) independientemente del navegador desde el que se acceda, simplemente agregando un valor determinado a cada etiqueta que no ha sido nombrada en el style.css.

Alternativa al tradicional reseteo de todos los estilos que logramos hacer con reset.css, Normalize.css aplica a todas las etiquetas un estilo por defecto. Este recurso es soportado por IE 8+, Firefox 4+, Safari 5+, Opera, Chrome y usado por proyectos tan importantes como Twitter Bootstrap o HTML5 Boilerplate.

A continuación vemos [jquery.mobile-1.3.2.min.css](#), que es la hoja de estilos de jQuery Mobile y que hace que nuestra aplicación tenga una apariencia propia para el header, footer, botones, color de fondo.

También le añadiremos a nuestra aplicación unos temas personalizados, por lo que hemos dejado el lugar donde tiene que ir insertado nuestros nuevos temas que crearemos con la aplicación web [“ThemeRoller”](#).

Tenemos que decir que es muy importante el lugar donde insertamos nuestros CSS ya que no es lo mismo poner antes o después una referencia a un CSS. Tenemos que tener en cuenta que las referencias CSS que vayamos incluyendo nuevas van anulando los CSS anteriores. Por ello al final hemos creado la referencia [app.css](#) que irá conteniendo aquellos aspectos CSS que queramos modificar de los anteriores o que no estén contenidos en los mismos.

8.2.2. Body : Página

La aplicación se diseña en base a jQuery Mobile por lo que la estructura de una página en jQuery Mobile será como se representa en la figura 54. Podrá contener un “header” (encabezado), “content” (contenido) y un “footer” (pie de página).

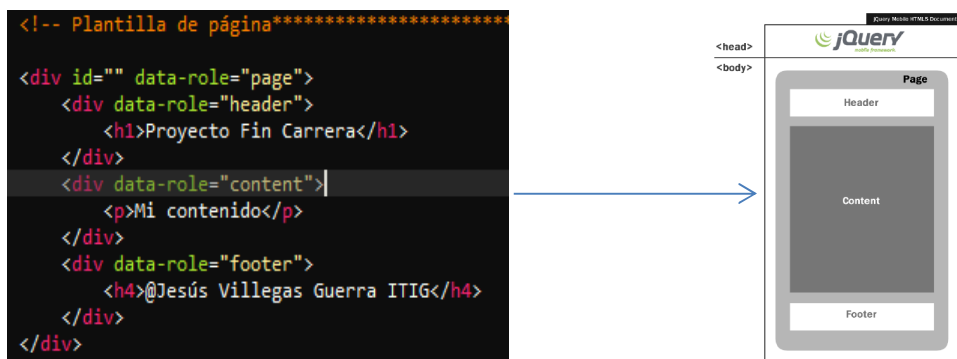


Figura 54: Estructura index.html. Body-Página jQuery Mobile

Como puede observarse cambian un poco las etiquetas pero siguen basándose en HTML5, hay etiquetas <div> y se puede usar todo lo que se usa en HTML5. Sin embargo, vemos que hay un atributo nuevo que es "data-role". HTML5 nos permite personalizar atributos con "data-" al que le añadiremos lo que deseamos. En el caso de jQuery Mobile le añaden "role" con lo que queda el atributo personalizado "data-role", habrá más atributos personalizados en jQuery Mobile que iremos viendo en nuestra aplicación.

Otro aspecto importante es que todas nuestras páginas creadas contendrán el atributo data-role="page" para que podamos ir saltando de unas a otras páginas ya que jQuery Mobile permite ir, vía AJAX, de una página a otra dentro de la misma página index.html y parecer como si estuviésemos navegando por distintas páginas, pero el contenido de todas ellas se mantendrán en solo un archivo, index.html.

Utilizaremos la plantilla de la figura 54 para ir creando todas nuestras páginas.

8.2.3. Body: Scripts

Los scripts, recomiendan las buenas prácticas, se deben colocar antes de finalizar la etiqueta <body> y no dentro de <head> para que no se entorpezca la carga del DOM. Con ello se cargará primero la página y después los scripts.

8.3. Anatomía de la aplicación realizada en Eclipse

A continuación mostraré como está dividida, en que archivos y con qué ficheros cuenta la aplicación realizada para Android, así como el contenido de cada uno de ellos y qué nos aparece en el package explorer de Eclipse, aunque para la compilación en Phonegap Build no será necesario todo el contenido que posee, solamente lo que nos aparece en "www" más un archivo llamado config.xml que veremos más adelante en el apartado de Phonegap Build:

- **src:** Contiene los archivos fuente ".java" del proyecto. Para nuestro trabajo existe un archivo llamado TrabajoFinCarrera.java. Este archivo es que inicia la actividad, así como es el encargado de iniciar la splash-screen antes de que comience la aplicación. El archivo java se lista debajo del nombre del paquete de la aplicación que en nuestro caso es edu.uoc.tfc.

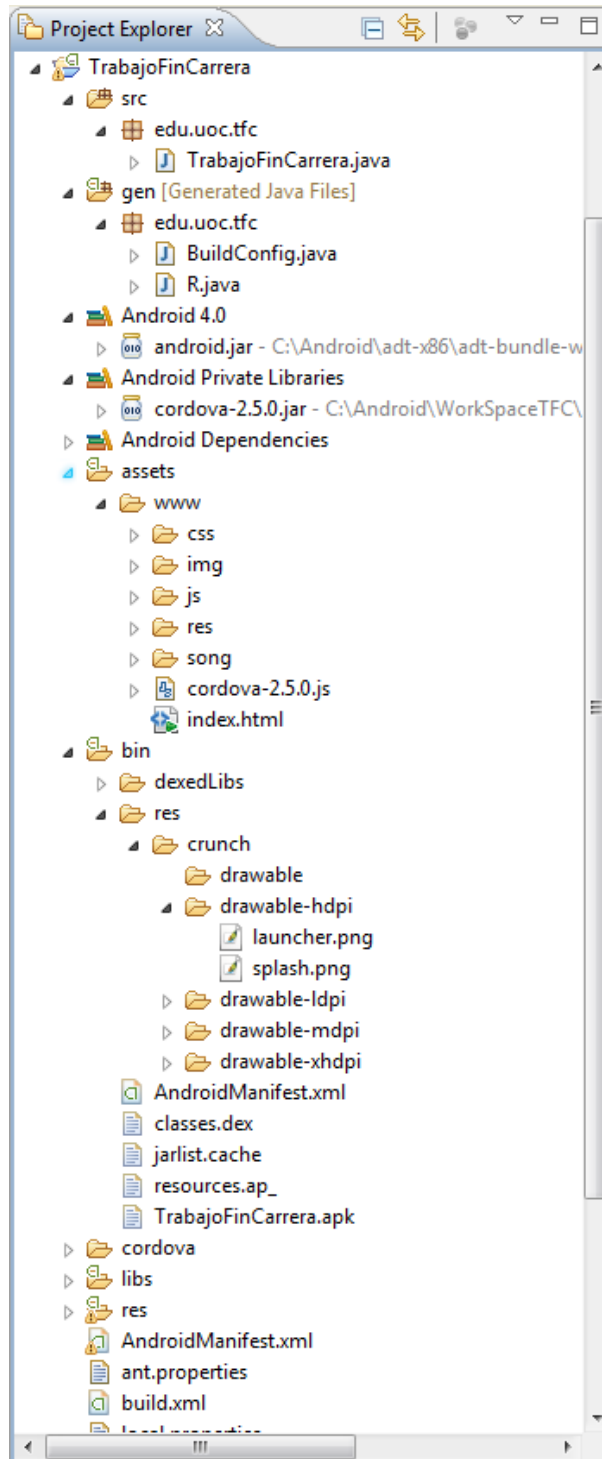


Figura 55: Anatomía de la aplicación. Android.

Alumno: Jesús Villegas Guerra

- **gen:** Contiene el archivo R.java, un archivo generado por el compilador que hace referencia a todos los recursos encontrados en el proyecto. Este archivo no debe tocarse o modificarse.
- **Librería Android 4.0:** Este elemento contiene un archivo, Android.jar que tiene todas las librerías de clase necesarias para la aplicación Android y que hace referencia al SDK instalado en nuestro ordenador, concretamente a la librería de la versión 14 de la plataforma de Android.
- **Librería Android Private Libraries:** Hace referencia a las librerías de Phonegap y que se pueden necesitar.
- **assets:** Contiene todos los archivos utilizados por la aplicación, tales como HTML, CSS, js,mp3,png, etc. Así como, para que pueda compilar en Phonegap Build será necesario incluirle un config.xml adaptado y distinto del config.xml que propiamente se posee en Eclipse para Android. Tengamos en cuenta que el config.xml generado en Eclipse es exclusivo para Android y nosotros necesitamos un config.xml para varias plataformas (Android, iOS, Windows Phone). El config.xml generado por la propia aplicación está situado en la carpeta "res/xml"
- **bin:** Esta carpeta contiene los archivos creados por el ADT (Android Development Tools, plugin de Eclipse) durante el proceso de creación. En particular nos interesa el archivo creado ".apk" que es el archivo binario y listo para poder instalarlo en cualquier Smartphone Android.
- **res:** Contiene recursos utilizados en la aplicación, importe comentar los archivos "drawable" que contiene los archivos splash.png (imagen de la splash-screen) y launcher.png (imagen icono de la aplicación) en sus distintas resoluciones.
- **Android Manifest.xml:** Se especifican los permisos que serán necesarios incluir en la aplicación (geolocalización, internet, etc) así como las versiones mínima y actual para las que se realiza la aplicación.

8.4. Aspecto Visual: Multidispositivo.

Mediante las hojas de estilo (CSS) se crea el aspecto de lo desarrollado en HTML5. A continuación muestro todas las hojas de estilo utilizadas en la aplicación:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
<meta name="format-detection" content="telephone=no" />  
<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1,  
width=device-width, height=device-height, target-densitydpi=device-dpi" />  
<link rel="stylesheet" type="text/css" href="css/normalize.css" />  
<link rel="stylesheet" type="text/css" href="css/jquery.mobile-1.3.2.min.css" />  
<link rel="stylesheet" type="text/css" href="css/miTema.css" />  
<link rel="stylesheet" type="text/css" href="css/app.css" />
```

Figura 56: Aspecto visual. CSS utilizados

jQuery Mobile cuenta con una herramienta que permite modificar el estilo de la interfaz, es llamado ThemeRoller. Con esta herramienta se permite crear diferentes temas a la vez pudiéndose modificar características tales como el color y sombra de los botones, colores solidos o degradados aplicados a fondos, etc.

Tras lograr el diseño deseado podemos bajarnos la hoja de estilo creada con los temas que se hayan realizado para incluirlos en el código. En nuestro caso, he realizado dos temas "a" y "b" que serán los aplicados a la implementación del proyecto, llamados miTema.css y en su defecto tomará los determinados por jquery.mobile-1.3.2.min.css en los casos que no tengamos definido un aspecto determinado.

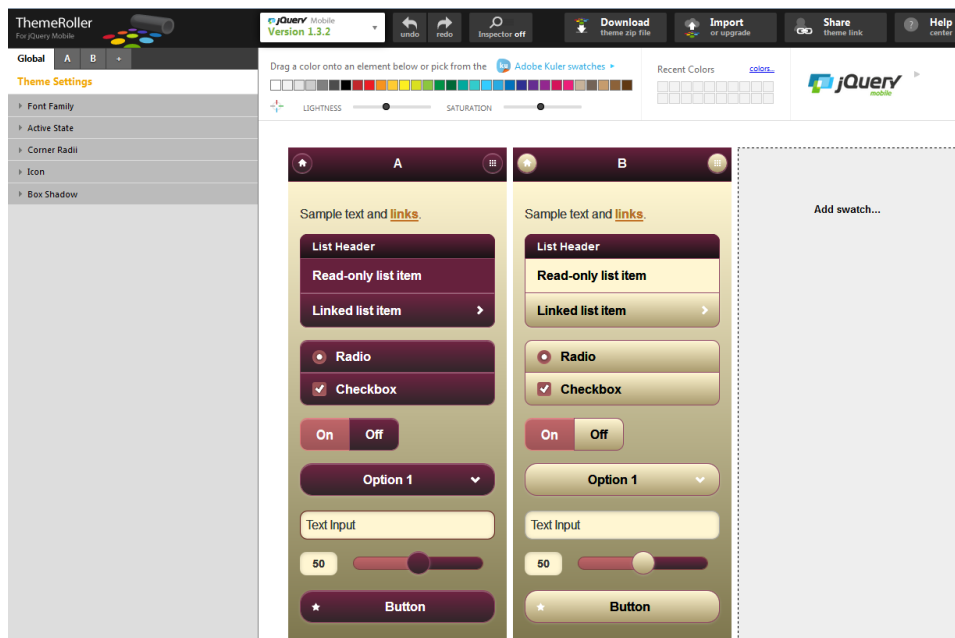


Figura 57: Aspecto visual. ThemeRoller

Alumno: Jesús Villegas Guerra

Además de los CSS ya comentados se ha creado el archivo [app.css](#) para mejorar y modificar algunos aspectos de los estilos y hacer el diseño más personalizado y que veremos a continuación.

8.4.1. App.css: Media queries (etiqueta @media)

Media queries es un módulo CSS3 que permite la representación de contenido para adaptarse a distintas resoluciones de pantalla (desde un Smartphone a pantallas HD). Se convirtió en un estándar recomendado por la W3C en Junio del 2012 y es un principio básico de la tecnología de Diseño web adaptativo.

La aplicación realizada se adapta a todos los dispositivos (multidispositivo) que puedan tener acceso a la aplicación desde teléfonos de 3,5" a tablets de 10" e incluso mayores, como pantallas de sobremesa.

Esto lo hace posible por un lado los CSS de jQuery Mobile (jquery.mobile-1.3.2-min.css y miTema.css) y por otro el que hemos creado nuevo, app.css.

Tanto los CSS de jQuery Mobile como el nuestro utiliza la nueva etiqueta de HTML5 "@media" para hacer la app adaptable a distintos dispositivos (responsive app).

La implementación se ha realizado comenzando por un dispositivo genérico, en nuestro caso el dispositivo de 3,5" y de aquí he ido adaptando la media querye a dispositivos mayores.

Algún código de media querye de nuestra aplicación es el siguiente:

```
/*
 * Modificación de algunos aspectos css cuando el ancho del dispositivo comience a ser igual o superior a 650px y
 * menor que 1150px. 480<=ancho del dispositivo<1150px.
 * Esta regla comienza a ser visible para dispositivos de 4" apaisado(landscape), 5,5" vertical (portrait),
 * tablets de 7" en vertical (portrait)
 */
@media only all and ( min-width:650px){
    .texto{
        font-size:large;
        text-align:justify;
    }

    .google-maps {
        margin: 0 4em 0 4em;
    }
}
```

Figura 58: Aspecto visual. CSS. Media querye

Alumno: Jesús Villegas Guerra

En el código de la figura 58 estamos indicando que se aplique la regla CSS en el caso de que el ancho mínimo sea de 650px (min-width: 650px). Otro aspecto a reseñar es que podríamos no haber puesto el operador "only" ya que está indicado para prevenir que navegadores antiguos que no soportan media queries con funciones apliquen los estilos asignados, pero es bueno ponerlo siempre en el código para prevenir. Así como el operador "all" indica que la regla es aplicable a todos los dispositivos.

Con las siguientes tres etiquetas hacemos nuestra app adaptable a los distintos formatos de dispositivos. En el momento de utilizarse un ancho superior al que se tenía en un principio se adaptan algunos elementos a ese dispositivo, que en nuestro caso han sido las imágenes, el tamaño de la fuente y el área destinada a ver el mapa de google-maps.

```
@media only all and ( min-width:480px){  
@media only all and ( min-width:650px){  
@media only all and ( min-width:1150px){
```

Figura 59: Aspecto visual. CSS. Media querye II.

A continuación vamos a ver cómo se van aplicando las reglas de estilo a distintos dispositivos (multidispositivo- multidevice). Para hacer las pruebas se ha utilizado Ripple de Google ya que el único dispositivo que poseo para hacer las pruebas en real es un Smartphone de 3,5". Comprobaremos como la misma página se comporta en un dispositivo de 320x480, 480x800, Tablet 7", tanto en horizontal como en vertical (portrait, landscape).

Alumno: Jesús Villegas Guerra



Figura 60: Aspecto visual. Ripple.320x480px.

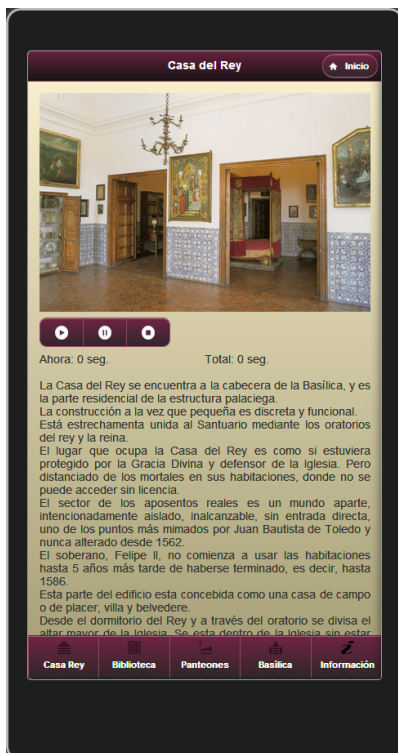


Figura 61: Aspecto visual. Ripple. 480x800



Figura 62: Aspecto visual. Ripple. Tablet 7"

Por tanto vemos como la imagen se va adaptando a los distintos anchos de los dispositivos, así como el tamaño de fuente va cambiando, aunque no se puede apreciar este aspecto en las figuras. Y se ve en el menú situado en el "footer" como se adaptan las letras desde el dispositivo de 320x480 en las que se acortan para que quepan todos los botones, hasta la Tablet de 7" donde se expanden los botones y se puede leer completamente el texto de los mismos.

Las imágenes tienen un ancho no superior a 600px, están en formato "png". Con esto se ha intentado que no tuviesen un peso muy alto, aproximadamente cada imagen tiene unos 500Kb, con lo que no sobrecargar la aplicación. Esta imagen se va adaptando a los distintos dispositivos, según su ancho.

8.5. Funcionalidades

En este apartado describiremos todas las funcionalidades que presentamos en el punto de diseño de la aplicación. Las pantallas que aparecerán a continuación son reales, están tomadas del Smartphone Cubot 7+ de 3,5".

Alumno: Jesús Villegas Guerra

Tenemos que comentar que la aplicación (fichero apk) tiene un peso de 26,40MB. Posee 34 páginas, 5 iconos de diseño propio para los menús, 33 imágenes (utilizando Photoshop) sin contar las imágenes creadas para la pantalla splash-screen e icono de la aplicación y 23 archivos de sonido mp3.

Al comenzar a realizar la app una decisión que tuve que tomar es decidir si los archivos de sonido mp3, así como las imágenes estuviesen en la propia aplicación o fuesen tomadas desde un servidor externo. Probando estas dos opciones, decidí finalmente integrar dentro de la aplicación las imágenes y mp3 ya que al realizar las pruebas se tardaba en ocasiones 20 o 30 segundos en bajar un archivo mp3 para poder oírlo. Las imágenes sin embargo eran más rápidas, instantáneas. Probé la aplicación y la conexión a internet en distintos lugares y dependiendo de la conexión a veces si era rápida la bajada del archivo pero en otras ocasiones no. Con respecto al peso de la aplicación, decir que es grande al tener 26,40MB pero he visto que otras aplicaciones tienen incluso 41MB como la del “National Museum of Korea”. Pero con este aspecto se mejora la calidad de la aplicación, aunque la misma tenga más peso.

A las funcionalidades indicadas en la fase de diseño **se han añadido**:

- Una pantalla de inicio splash-screen.
- Se han cambiado los botones que aparecen en el “head” (cabecera) con un texto explícito de lo que realiza para dar una mayor claridad al usuario (funcionalidad ir a inicio).
- Se ha incluido el poder visualizar la situación del Monasterio de El Escorial al añadirse un objeto en el que se inserta una imagen de google-maps.
- Se ha introducido el botón “atrás” en algunas pantallas.
- Las transiciones entre páginas siguen un patrón coherente, pudiendo verse transiciones hacia delante o atrás (reverse).
- Un botón en la página de Horarios que accede a la página de Patrimonio Nacional para consultar los días festivos en que no abre el Monasterio.

Por tanto, las funcionalidades que vamos a explicar a continuación y que han sido realizadas se muestran como siguen, obteniéndose el 100% de las funcionalidades diseñadas:

Funcionalidades
• Pantalla Splash-Screen.
• Ir a inicio.
• Leer un texto
• Ver una imagen
• Llamar por teléfono
• Acceder al menú
• Utilizar el control de audio
• Entrar en una página
• Visualizar google-maps
• Acceder a página externa

Figura 63: Funcionalidades. Tabla de funcionalidades

8.5.1. Pantalla Splash-Screen

La pantalla splash-screen es una pantalla de bienvenida, es una imagen que aparece mientras el programa está cargando.

Las pantallas splash-screen tienen tanto detractores como aquellos que la ensalzan. A modo de ejemplo en la documentación para desarrolladores de Android se indica expresamente que: "Don't show unsolicited help, except in very limited cases", es decir, nos indica que no deberíamos mostrar splash-screen a no ser que sea muy necesario. Conocemos que estas pantallas han sido muy utilizadas para mostrar publicidad antes de iniciar la aplicación con lo que pueden resultar molestas. Realmente al usuario final lo que le interesa cuando abre una aplicación es encontrarse con la misma y poder usarla al instante.

No obstante, hemos incluido una pantalla de bienvenida al ser un trabajo de Fin de Carrera. La duración de esta pantalla es de 10 segundos.

Alumno: Jesús Villegas Guerra

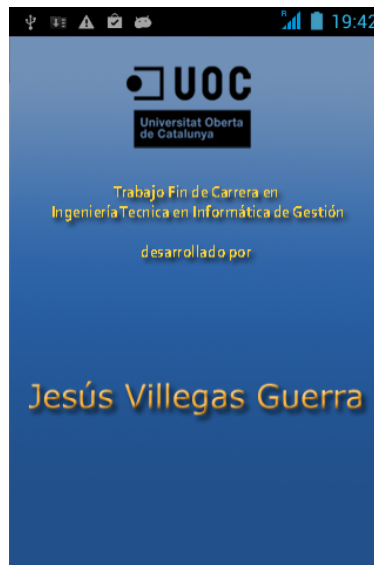


Figura 64: Funcionalidades: Pantalla Splash-Screen.

Para ello hemos usado el siguiente código que se ha puesto dentro del fichero java llamado "TrabajoFinCarrera.java" en la carpeta "src":

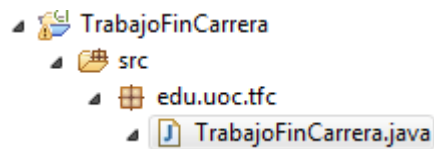


Figura 65: Funcionalidades: Splas-screen. Archivo "src"

Siendo el código como sigue:

```
package edu.uoc.tfc;
import android.os.Bundle;

public class TrabajoFinCarrera extends DroidGap
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        super.setIntegerProperty("splashscreen", R.drawable.splash);
        // Set by <content src="index.html" /> in config.xml
        //super.loadUrl(Config.getStartUrl());
        super.loadUrl("file:///android_asset/www/index.html",10000);
    }
}
```

Figura 66: Funcionalidades. Splash-Screen. Código

Alumno: Jesús Villegas Guerra

Del código de la figura 66 tenemos que reseñar dos líneas:

- **super.setIntegerProperty("splashscreen", R.drawable.splash)** : Esta línea nos indica que se tomará la imagen que haya dentro del archivo "drawable" y llamada "splash".
- **super.loadUrl(file:///android_asset/www/index.html,10000)**: Que nos indica que se muestre la pantalla splash durante 10.000 milisegundos, o sea, 10 segundos.

Antes de iniciar este código habremos de haber puesto dentro del archivo "res>drawable-XXX" las imágenes que serán el propósito de la splash-screen:

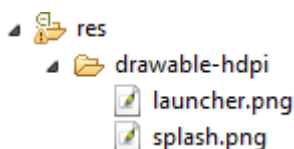


Figura 67: Funcionalidades. Splash-Screen. Archivo "res"

Este código está realizado para poder usarse en Android y poder compilar el fichero "apk", sin embargo comentamos en nuestras especificaciones no funcionales que la aplicación sería también multiplataforma, como veremos más adelante en el apartado de Phonegap Build, este código lo tendremos que adaptar para varias plataformas (Android, iOS, Windows Phone).

8.5.2. Ir a inicio

Desde cualquier pantalla de la aplicación se puede ir al inicio de la aplicación.

Alumno: Jesús Villegas Guerra



Figura 68: Funcionalidades. Ir a inicio.

Se ha modificado el botón de inicio, en el diseño solo aparecía con el icono y se ha añadido la palabra "Inicio" para hacerlo más claro de cara al usuario.

Todas las pantallas poseen el botón de inicio, e incluso a algunas de ellas (las que poseen listados) se les ha añadido el botón de atrás como se puede ver en la siguiente figura.

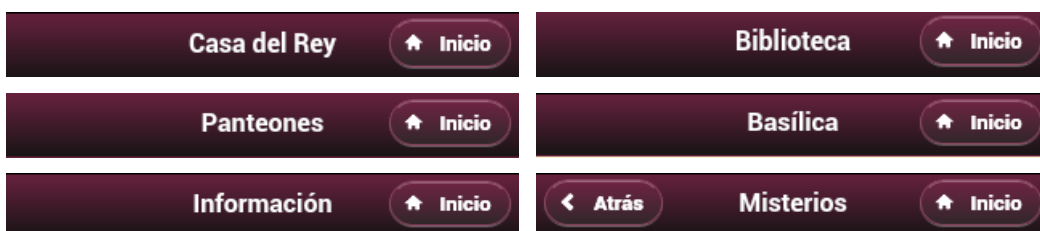


Figura 69: Funcionalidades. Botón Inicio y Atrás.

Para incluir el botón inicio, utilizamos jQuery Mobile indicándoselo en cada caja <div> "header" de cada una de las páginas que hemos realizado.

```
<div data-role="page" id="Misterios" data-add-back-btn="true" data-back-btn-text="Atrás" data-theme="b">
  <div data-role="header">
    <h1>Misterios</h1>
    <a href="#inicio" data-icon="home" class="ui-btn-right" data-direction="reverse">Inicio</a>
  </div>
```

Figura 70: Funcionalidades. Código botón ir a Inicio y Atrás.

Como puede verse en el código, no es necesario indicarle que se cree un botón, solamente con incluir la etiqueta ``, se crea el botón. Con `data-icon="home"` se le indica al botón que use el icono de "home", es decir, inicio. Al indicar `class="ui-btn-right"`, pedimos que el botón se integre en la parte derecha de la pantalla y con `data-direction="reverse"` la transición de la pantalla se realiza hacia atrás.

Respecto al botón de "Atrás", el código se coloca en la caja `<div>` de la página llamada "page" (`data-role="page"`). Para indicar que se cree el botón atrás incluimos `data-add-back-btn="true"` y le indicamos que palabras tienen que aparecer en el botón al indicar `data-back-btn-text="atrás"`. Finalmente `data-theme="b"` nos indica el tema CSS que se aplica a la página.

Las transiciones por defecto serán de tipo "slide" hacia delante, a no ser que le indiquemos, como hemos hecho con `data-direction="reverse"`, tomando la dirección hacia atrás.

Para ello se le indica el siguiente código en nuestra carpeta `js/appConfig.js`:

```
$(document).on('mobileinit', function(){
  /*Transiciones*/
  $.mobile.defaultPageTransition = 'slide';
});
```

Figura 71: Funcionalidades. Transiciones. `appConfig.js`

Para esta aplicación se ha usado solamente una página html, llamada `index.html`. Con jQuery Mobile se pueden usar todas las páginas `.html` que se deseen, sin embargo la ventaja que tiene jQuery Mobile es que con solo una página se tiene la sensación de estar viajando de otra como si fuesen páginas independientes, pero todas ellas están en la misma.

Alumno: Jesús Villegas Guerra

8.5.3. Leer un texto

Al elegir una página desde la pantalla de inicio, el usuario podrá leer un texto. Sin embargo la primera vez que pulse en cualquiera de las pantallas de inicio (Casa del Rey, Biblioteca, Panteones, Basílica y Más información) aparecerá un mensaje “pop-up” indicando un modo de usar la aplicación.

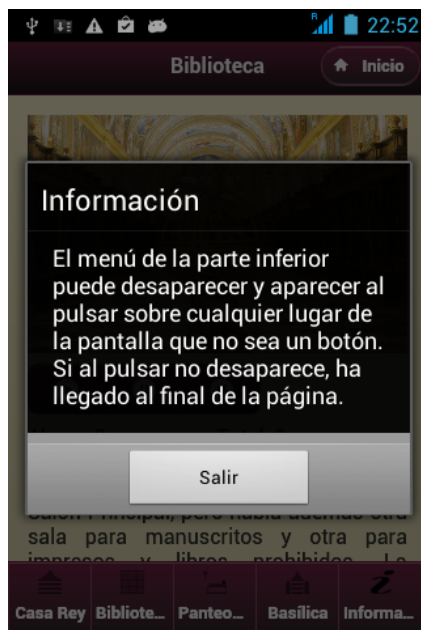


Figura 72: Funcionalidades. Leer un texto. Mensaje pop-up

Este mensaje se ha creado con el siguiente script:

```

/*
 * Muestra un mensaje Pop Up al principio de las páginas Casa Rey, Biblioteca,
 * Panteones, Basílica y Mas información.
 * Solo se muestra una vez en una de las páginas principales.
 */
var contador=0;
$('#CasaRey,#Biblioteca,#Basílica,#Panteones,#Informacion').one('pageshow',function(){
    if(contador==0){
        document.addEventListener("deviceready", onDeviceReady, false);

        function onDeviceReady() {
            // vacío.
        }

        // función que hace algo cuando se pulsa el botón salir. No hacemos nada. Indiqué un mensaje en consola para probar la función.
        function alertDismissed() {
            console.log("Alert dismissed");
        }

        function showAlertIni() {
            navigator.notification.alert(
                'El menú de la parte inferior puede desaparecer y aparecer al pulsar sobre cualquier lugar de la pantalla que no sea',
                alertDismissed, // callback
                'Información', // título
                'Salir' // boton de salir
            );
            showAlertIni();
            contador=1;
        }
    }
});

```

Figura 73:Funcionalidades. Leer un texto. Script Pop-up

Utilizo en el código la expresión `$('#CasaRey,#Basílica,#Panteones,#Informacion).one ('pageshow', function(){` , para indicar que solo se muestre una vez el mensaje pop-up. Al indicar en la expresión "one" se le dice a jQuery Mobile que solo se muestre una vez el mensaje al entrar en una página, si se entra de nuevo en la misma página indicada por los selectores "#" ya no se volverá a mostrar el mensaje. Por ejemplo al entrar en la página CasaRey una vez se muestra el mensaje pop-up, si se entra de nuevo ya no lo hace, y así en todas las demás, pero además quería que si entramos en la página CasaRey por segunda vez ya no se mostrara y que tampoco se mostrara en las demás, para ello he tenido que crear un contador llamado "contador" que hace que si su valor es 1 el mensaje ya no se volverá a mostrar en ninguna otra página ya que no se llamará a la función `showAlertIni()`.

Este código lo podemos ver en la página de documentación de Phonegap siguiente: http://docs.phonegap.com/es/1.0.0/phonegap_notification_notification.md.html, llamado "notification.alert". Además podremos encontrar otros interesantes que usan las propiedades del teléfono como `notification.vibrate` o `notification.beep`.

Al entrar en las páginas se puede leer un texto. Estos textos tienen reglas CSS que se han incorporado para mejorarlo.

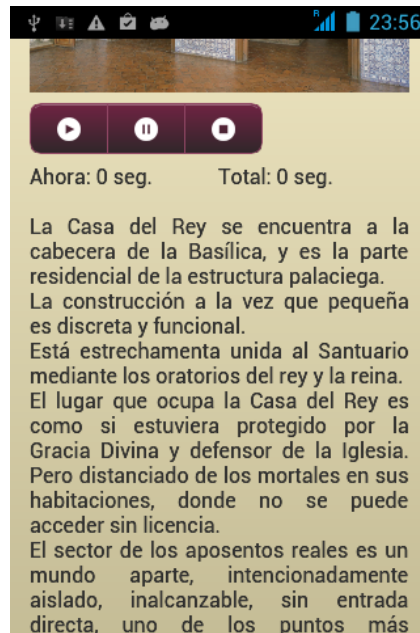


Figura 74: Funcionalidades. Leer un texto. Pantalla

Se puede observar en la figura 74 que el texto está justificado. En esta pantalla no se puede apreciar pero las reglas CSS, como comentábamos en el punto de media queries hacen que el tamaño de fuente se adapte y se incremente en dispositivos mayores en tamaño.

```
/*  
 * Todos los textos en un principio tendrán una fuente de tamaño medio y justificado.  
 * Este aspecto será modificado por @media (media querie) cuando los anchos de los  
 * dispositivos vayan cambiado.  
 */  
.texto{  
    font-size: medium;  
    text-align: justify;  
}  
  
/*  
 * Le damos un margen adicional al texto en la página Mas información>Horarios por arriba y  
 * la izquierda de la página.  
 */  
.texto-horarios {  
    margin: 1em 0 0 2em;  
    font-size: medium;  
}
```

Figura 75: Funcionalidades. Leer un texto. Código CSS.

Alumno: Jesús Villegas Guerra

El aspecto indicado de la justificación del texto se muestra en el código de la figura 75. Todas las páginas tienen un texto con una clase llamada `class="texto"`, al ser llamada desde la regla CSS ubicada en `css/app.css`, el tamaño y alineación cambiarán. También puede verse una regla diferente para el texto de la clase `class="texto-horarios"` situada en la página de "Horarios". Con esta regla queremos dejar espacio adicional arriba y a la izquierda. También utilizo la medida "em" que es escalable, es proporcional a la medida del texto actual que se usa en la página, hace referencia al ancho de la letra "m". Además las recomendaciones de W3C es que se use esta medida lo que hace que las páginas sean más escalable y adaptables a distintos formatos de pantallas.

8.5.4. Ver una imagen

En muchas de las pantallas aparecen imágenes del tema sobre el que se está tratando en la página.

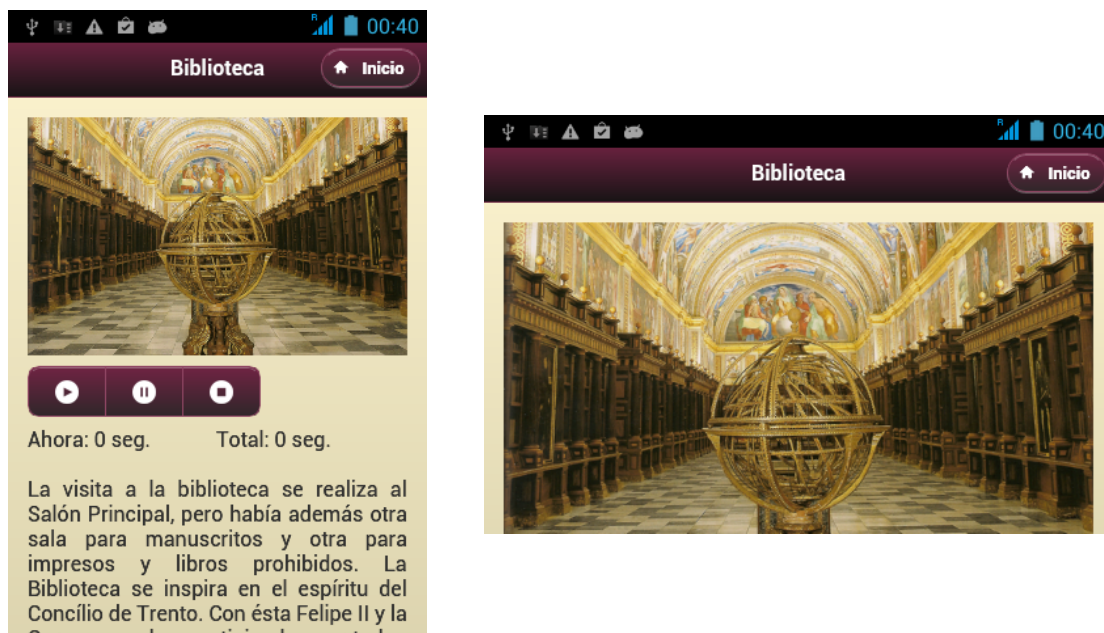


Figura 76: Funcionalidades. Ver una imagen. Pantalla

Las imágenes se adaptan al ancho de pantalla que en ese momento esté activo. En la figura anterior se aprecia la imagen con disposición horizontal y vertical.

Alumno: Jesús Villegas Guerra

Para conseguir este aspecto se han tratado todas las imágenes con reglas css.

```
/*
 * Para todas las imágenes, le indicamos que tomen todo el ancho del dispositivo.
 * Claro esta siempre que la imagen pueda hacerlo por sus dimensiones. Habrá imagenes
 * que sean más pequeñas que el ancho del dispositivo, por ejemplo en aquellos casos
 * que el dispositivo tenga una anchura de 800px, ya que la imagen como máximo tiene
 * 600px.
 */
img{
  max-width:100%;
  display: block;
  margin: auto;
}

/*
 * Modificación de algunos aspectos css cuando el ancho del dispositivo comience a ser igual o superior a 1150px.
 * Esta regla puede comenzar a verse en dispositivos 7" apaisado (landscape) y 10" y superiores.
 */
@media only all and ( min-width:1150px){
  .imagen {
    width:800px;
  }
}
```

Figura 77: Funcionalidades. Ver una imagen. Código CSS y Media querye

Para que se adapte a todos los anchos disponibles en la regla CSS usamos max-width: 100%, solo usamos el ancho ya que el alto lo hace automáticamente porque lo que nos interesa en esta ocasión es la adaptabilidad al ancho del dispositivo. Además a la imagen se la trata como un bloque con los márgenes automáticos, de hecho deja márgenes a los lados del dispositivo. En la regla media querye "@media" usamos como ancho la expresión "width: 800px", con lo que indicamos que incremente la imagen para que pueda verse mejor.

8.5.5. Llamar por teléfono

Con la aplicación también podemos usar las funcionalidades del teléfono. Podemos acceder a llamar por teléfono con solo pulsar un botón en la app.

Siendo el código que lo hace posible el siguiente:

```
<h2>Teléfonos</h2>
<p>
  <a href="tel:918905903" data-mini="true" data-role="button" data-theme="a" data-inline="true">Llamar al 91 890 59 03</a>
  <a href="tel:918905904" data-mini="true" data-role="button" data-theme="a" data-inline="true">Llamar al 91 890 59 04</a>
  <a href="tel:918905905" data-mini="true" data-role="button" data-theme="a" data-inline="true">Llamar al 91 890 59 05</a>
</p>
```

Figura 78: Funcionalidades. Llamar por teléfono. Código.

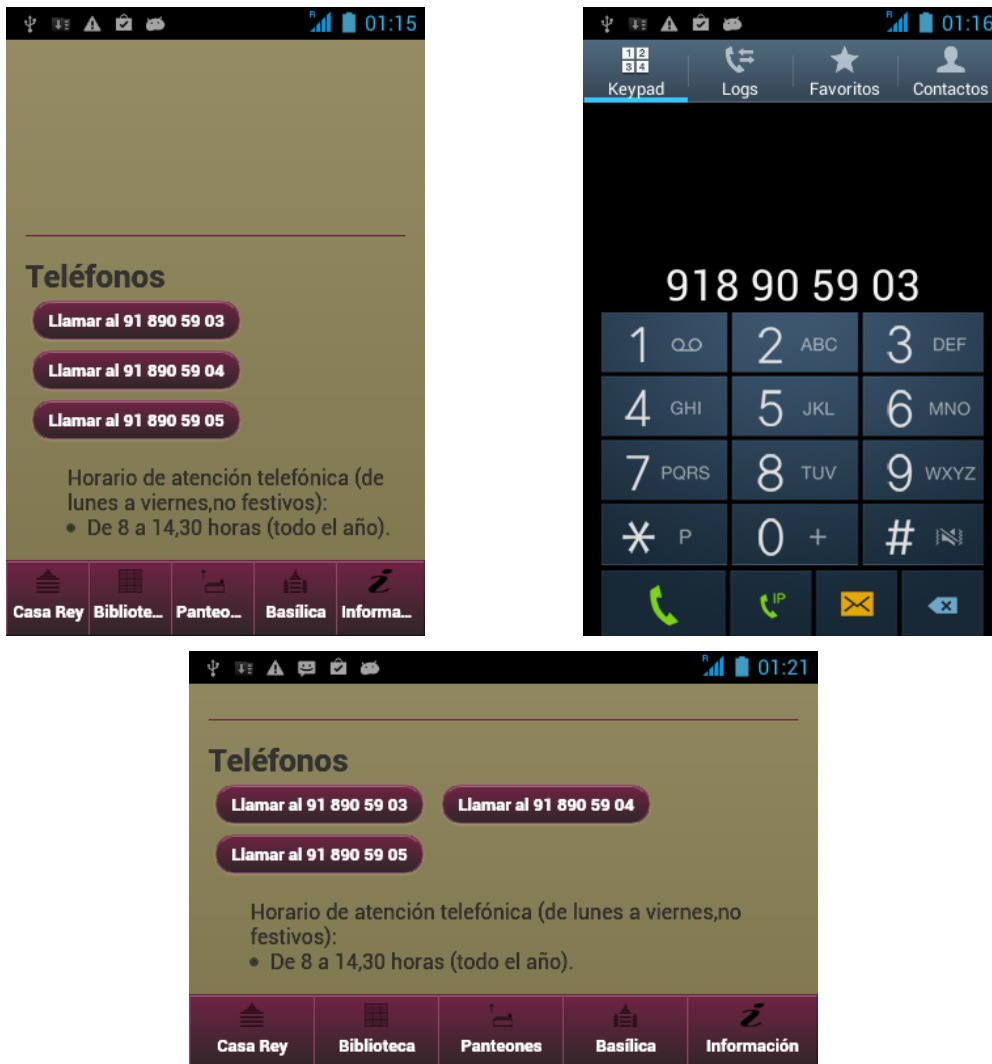


Figura 79: Funcionalidades. Llamar por teléfono. Pantalla

Al pulsar sobre cualquier botón de Llamar hace uso de la funcionalidad de llamada telefónica de nuestro móvil. Esto se consigue con la etiqueta ``. También observamos que los botones cambian de posición según el ancho del dispositivo, esto se consigue con la expresión `data-inline="true"` y no se ocupa el ancho total de la pantalla.

8.5.6. Acceder al menú

Desde todas las páginas se tiene acceso al menú situado en el "footer" (pie de página) de la aplicación. El pie de página se ha creado de manera fija, y el usuario puede quitarlo, según hemos visto en otros apartados al pulsar sobre algún lugar de la pantalla que no sea un botón,

Alumno: Jesús Villegas Guerra

así lo explicaba también en el pop-up de información que nos aparecía al iniciar por primera vez una página.

En las imágenes siguientes se puede ver la misma página con y sin el menú.



Figura 80: Funcionalidades. Acceder al menú. Pantallas

Para lograr este efecto se ha usado el siguiente código:

```
<!-- Página Casa del Rey ***** -->
<div data-role="page" id="CasaRey" data-theme="a">
  <div data-role="header">
  <div data-role="content">
  <div data-role="footer" data-position="fixed" >
    <div data-role="navbar">
      <ul class="navbar-text">
        <li><a href="#CasaRey" data-icon="img-casarey">Casa Rey</a></li>
        <li><a href="#Biblioteca" data-icon="img-biblioteca">Biblioteca</a></li>
        <li><a href="#Panteones" data-icon="img-panteon">Panteones</a></li>
        <li><a href="#Basilica" data-icon="img-basilica">Basílica</a></li>
        <li><a href="#Informacion" data-icon="img-informacion">Información</a></li>
      </ul>
    </div>
  </div>
</div>
</div>
```

Figura 81: Funcionalidades. Acceder al menú. Código 1.

Alumno: Jesús Villegas Guerra

En el código aparece la expresión `data-position="fixed"` para lograr que el menú este permanentemente en la página y accesible al usuario en todo momento.

Otro aspecto que podemos ver es que no todos los menús son iguales en su código ya que para que haya una navegación coherente y de la sensación al usuario de estar yendo hacia el final o principio de la aplicación, se ha incluido en cada botón de la navegación si la misma va hacia delante o atrás. Por defecto la navegación es "slide" hacia delante. Sin embargo en el siguiente extracto de código de otra página se puede ver que no todos los botones tienen la misma dirección:

```
<!-- Página Carlos III ***** -->
<div data-role="page" id="CarlosIII" data-add-back-btn="true" data-back-btn-text="Atrás" data-theme="a">
  <div data-role="header">
  <div data-role="content">
  <div data-role="footer" data-position="fixed">
    <div data-role="navbar">
      <ul class="navbar-text">
        <li><a href="#CasaRey" data-icon="img-casarey" data-direction="reverse">Casa Rey</a></li>
        <li><a href="#Biblioteca" data-icon="img-biblioteca" data-direction="reverse">Biblioteca</a></li>
        <li><a href="#Panteones" data-icon="img-panteon" data-direction="reverse">Panteones</a></li>
        <li><a href="#Basilica" data-icon="img-basilica" data-direction="reverse">Basilica</a></li>
        <li><a href="#Informacion" data-icon="img-informacion" data-direction="reverse">Información</a></li>
      </ul>
    </div>
  </div>
</div>
</div>
```

Figura 82: Funcionalidades. Acceder al menú. Código 2.

En este caso se ha incluido `data-direction="reverse"` con lo que la transición se realiza hacia atrás.

Para poder crear una barra de menú se ha introducido `data-role="navbar"`, además se ha incluido la etiqueta `` de listas no ordenadas junto con la etiqueta `` y la clase `class="navbar-text"`. Lo conseguido con este código es que el texto si no cabe dentro del botón se adapte y cambie en su forma, por ejemplo biblioteca no cabe y se acorta la palabra quedando de la siguiente forma:



Aunque en dispositivos mayores la palabra se ve completamente.

Otro aspecto que se ha incluido es la personalización de los iconos. Para ello nos ayuda la expresión `data-icon`.

Sin embargo el proceso de personalizar iconos es un poco complejo. Se han creado los iconos con un tamaño definido de 18x18 pixeles colocándolos en la carpeta css/images. Además hay que indicarle a jQuery que hemos creado nuevos iconos y modificar el CSS que teníamos introduciendo nuestro nuevo CSS en css/app.css el siguiente código:

```
/*
 *Introducimos con este css las nuevas imágenes de 18x18px utilizadas en los menus
 * del footer como iconos.
 */
.ui-icon-img-basilica { background-image: url("images/18-basilica.png");
                        border-radius: 0;}
.ui-icon-img-biblioteca { background-image: url("images/18-biblioteca.png"); }
.ui-icon-img-casarey { background-image: url("images/18-casarey.png"); }
.ui-icon-img-informacion { background-image: url("images/18-informacion.png"); }
.ui-icon-img-panteon { background-image: url("images/18-panteon.png"); }
```

Figura 83: Funcionalidades. Acceder al menú. CSS. Iconos personalizados

Con este código le indicamos donde están situadas las imágenes de nuestros iconos y como se llaman. Para poder llamarlas hay que incluir como termino fijo en el CSS ".ui-icon" y a continuación el nombre que usaremos para llamar a esas imágenes, es decir, "-img-basilica", quedando la expresión completa ".ui-icon-img-basilica". Desde la página index.html se llama por tanto a cada icono por su nombre de la forma data-icon="img-basilica".

Sin embargo los iconos de esta forma aparecen redondeados:



Por lo que hay que modificar otra regla CSS para que no nos aparezca ese redondeado incluyendo el código siguiente en css/app.css:

```
/*
 *Quitamos en los iconos el aspecto redondeado para poder usar las nuevas
 *imágenes creadas como iconos en los menus del footer.
 */
.ui-icon{
    border-radius:0;
}
```

Figura 84: Funcionalidades. Acceder al menú. CSS. Quitar iconos redondeados.

8.5.7. Utilizar el control de audio

El control de audio utiliza tres botones, comienzo, parar y pausa. Se puede apreciar en la siguiente imagen como han sido creados:



Figura 85: Funcionalidades. Utilizar control de audio. Pantalla

También se indica el tiempo actual transcurrido y el total en segundos de la audición.

Para incluir el control de audio se ha utilizado un código ya predefinido y creado por Mobile Developer Solutions bajo licencia MIT. Aunque parte de este código ya aparece en el API de Phonegap.

Por tanto he modificado el código para adaptarlo a nuestra aplicación. Se ha utilizado el archivo javascript llamado media.js situada en js/media.js para nuestra app.

En cuanto a la parte de **diseño de los botones**, se han realizado iconos personalizados para nuestra aplicación. Situándolos en el fichero img, llamados img/18-Pause.png, img/18-Play.png y img/Pause.png con una dimensión de 18x18 pixeles y aplicándole nuestro tema personalizado data-theme="a".

```
<div data-role="controlgroup" data-type="horizontal">
  <div data-role="button" id="audio-FelipeII" data-theme="a"></div>
  <div data-role="button" class="pauseaudio" data-theme="a"></div>
  <div data-role="button" class="stopaudio" data-theme="a"></div>
</div>
<div class="ui-grid-a">
  <div class="ui-block-a"> Ahora: <span class="audio_position">0 seg.</span></div>
  <div class="ui-block-b">Total: <span class="media_dur">0</span> seg.</div>
</div>
```

Figura 86: Funcionalidades. Utilizar control de audio. Código HTML5+ jQuery Mobile

Alumno: Jesús Villegas Guerra

Para crear el efecto de botones unidos se ha aplicado data-role="controlgroup" y además si se le añade data-type="horizontal", los botones permanecen unidos en una línea. Por defecto si no se indica data-type, toma que los botones deben de ir en posición vertical.

Otro aspecto usado en el código es la forma en que se usan las celdas, en html5 se usan las etiquetas <td> y <tr> sin embargo con jQuery Mobile se pueden también usar la clase class="ui-grid-a", class="ui-grid-b" o class="ui-grid-c", dependiendo si se necesitan dos, tres o cuatro columnas.

Respecto al comportamiento de la aplicación con el control de audio tenemos que comentar que al estar en **funcionamiento una audición y cambiar de página**, la audición se para, consiguiendo este aspecto con el siguiente código:

```
/*  
 * Antes de entrar en cualquier página, si hay una audición en marcha, llamamos a la función  
 * stopAudio para hacerla parar, además de poner a cero los contadores del control de audio.  
 */  
$(document).on('pagebeforeshow', function(){  
    stopAudio();  
    setAudioPosition(0);  
    $(".media_dur").html(0);  
});
```

Figura 87: Funcionalidades. Utilizar control de audio. Código para audición.

Al indicar \$(document), la acción se tendrá en cuenta para cualquier página. Al utilizar 'pagebeforeshow' se le dice a jQuery Mobile que realice alguna función antes de entrar en la página. Por consiguiente, antes de entrar en cualquier página se llamará a la función stopAudio() que parará la audición en caso de haberse iniciado. Con setAudioPosition (0) y \$(".media_dur").html (0) se pondrán a cero los contados del control de audio en segundos.

Al pulsar sobre el botón de **comenzar el audio**, que se identifica con el id="audio-Felipell", se activa un evento de ratón (mouse event) de jQuery de la forma siguiente:

```
$("#audio-FelipeII").on('click', function() {  
    var src = '/android_asset/www/song/felipeii.mp3';  
    playAudio(src);  
});
```

Figura 88: Funcionalidades. Utilizar control de audio. Evento

La función nos muestra que para el selector "#audio-FelipeII" dentro de \$("#audio-FelipeII") y cuando ocurra el evento click (.on ('click', function ()....)) ocurrirá lo que hay dentro de los paréntesis. Es decir, se indica mediante una variable donde está situado nuestro archivo mp3. A continuación se ejecutará la función playAudio (src) a la que se le pasa el parámetro que contiene la variable src. Por tanto, el audio está contenido en la propia app.

Como habíamos comentado anteriormente, se hicieron pruebas de audio, donde los archivos mp3 estaban alojados externamente en un servidor. Utilicé como servidor una carpeta pública de Dropbox. He dejado el código utilizado para ello en una de las funciones por si fuese necesario en otra ocasión alojar dichos mp3 en un servidor externo:

```
$("#audio-CasaRey").on('click', function() {  
    /*  
    * Codigo utilizado para realizar las pruebas de audio  
    * con el archivo mp3 bajado de internet.  
    */  
    /* check_network();  
    if ($('#connection').html() === 'No network connection') {  
        alert("Se necesita conexion a internet para escuchar esta audición.");  
        return false;  
    }  
    var src = 'https://dl.dropboxusercontent.com/u/144916658/meninas.mp3';*/  
  
    var src = '/android_asset/www/song/casarey.mp3';  
    playAudio(src);  
});
```

Figura 89: Funcionalidades. Utilizar control de audio. Audio externo a la app.

Lo único que cambia con respecto a nuestro código actual es que se chequea si hay conexión a internet con anterioridad, en caso de que no hubiera conexión, aparecería la alerta y en caso de que la hubiera, asignaría a la variable la url donde está situada el archivo mp3.

8.5.8. Entrar en una página

Entrar y navegar de una página a otra lo hace posible la navegación vía AJAX que tiene jQuery Mobile usando la referencia a la página en la etiqueta <a> con “href”, de la forma , al indicar almoadilla y el nombre de la página a la queremos que vaya, al pulsar sobre el botón nos lleva a la página llamada “pagina”. En nuestro caso:

```
<!-- Página de inicio***** -->
<div data-role="page" id="inicio" data-theme="a">
  <div data-role="header">
    <p>Real Monasterio de</p>
    <p>San Lorenzo de El Escorial</p>
  </div>

  <div data-role="content">
    <div id="img-logoInicial">
      
    </div>
    <p><hr></p>
    <div class="ui-grid-b" id="img-grid1">
      <div class="ui-block-a"><a href="#CasaRey"></a>Casa del Rey</div>
      <div class="ui-block-b"><a href="#Biblioteca"></a>Biblioteca</div>
      <div class="ui-block-c"><a href="#Panteones"></a>Panteones</div>
    </div>
    <div class="ui-grid-a" id="img-grid2">
      <div class="ui-block-a"><a href="#Basilica"></a>Basílica</div>
      <div class="ui-block-b"><a href="#Informacion"></a>Más información</div>
    </div>
  </div>
</div>

<!-- Página Casa del Rey ***** -->
<div data-role="page" id="CasaRey" data-theme="a">

<!-- Página Biblioteca ***** -->
<div data-role="page" id="Biblioteca" data-theme="a">
```

Figura 90: Funcionalidades. Leer un texto. Navegación AJAX

Al pulsar sobre el botón “Casa del Rey” nos lleva a la página indicada en la etiqueta con href="#CasaRey” que tiene el identificador id="CasaRey”. Además para que funcionen tanto las transiciones como la navegación es necesario que en todas las páginas aparezca data-role="page”, indicando que se pertenece a una misma página.

En ocasiones también será útil y necesario desactivar la navegación AJAX, para lo cual se indicará en la etiqueta la expresión data-ajax="false" como en la siguiente figura dentro de la página llamada “Horarios”:

```
<a href="http://www.patrimonionacional.es/Home/Noticias/cierres.aspx#escorial"
```

Alumno: Jesús Villegas Guerra

```
id="cierre" data-role="button" data-inline="true" data-ajax="false">Consultar cierres-festivos</a>
```

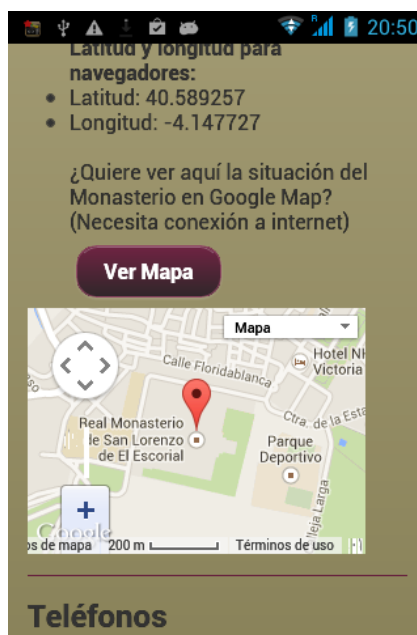
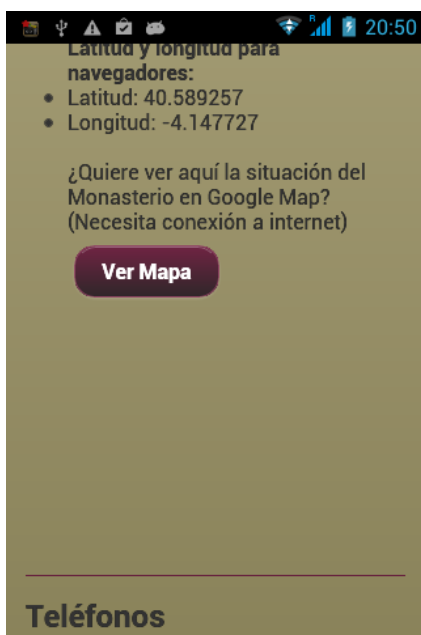
Figura 91: Funcionalidades. Leer texto. Desactivar AJAX.

Y en otros casos para referirnos a otra página dentro de nuestra aplicación será necesario indicar rel="external", que en nuestro caso no tenemos.

8.5.9. Visualizar google-maps

En la página de Horarios es posible ver la geolocalización del Monasterio como se muestra a continuación en las figuras.

El usuario si quiere tiene la posibilidad de pulsar el botón de "Ver Mapa" para visualizar el mapa. Como se aprecia queda una zona reservada al objeto del mapa. He optado por esta opción en vez de hacerlo automáticamente al entrar en la página para optimizar los recursos del teléfono y no obligar a usar un servicio si el usuario no quiere.



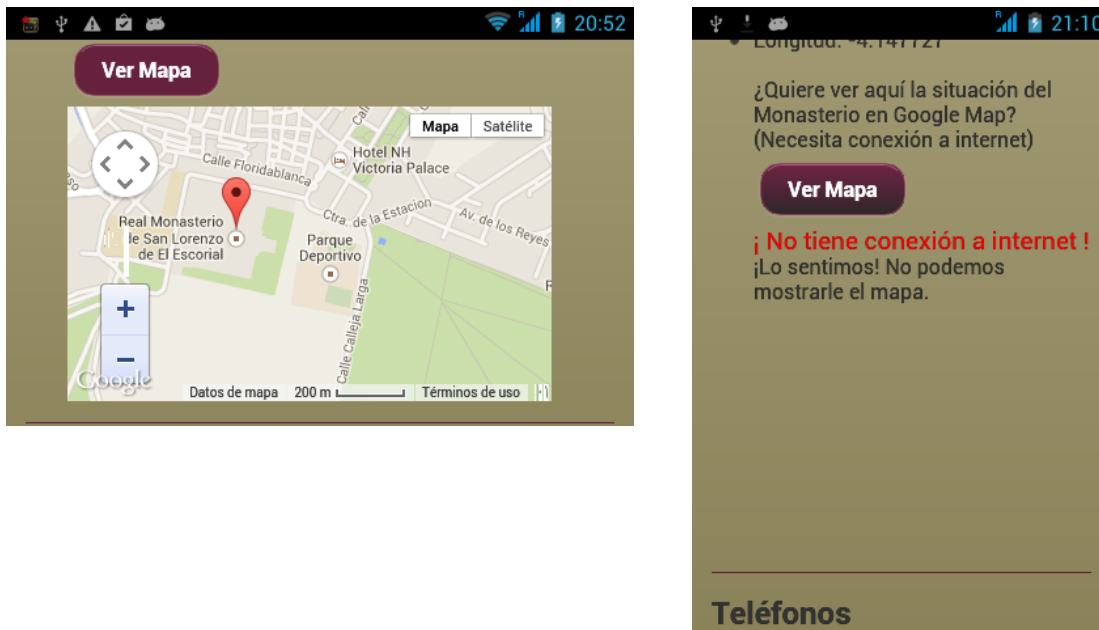


Figura 92: Funcionalidades. Visualizar Google-maps. Pantallas

En este apartado de geolocalización se utilizan los archivos de javascript siguientes:

- js/gmaps.js
- js/connection.js

Y se aplican reglas CSS situadas en css/app.sss.

Para poder usar la geolocalización hemos tenido que introducir varias líneas nuevas a nuestro código en el archivo index.html. Por un lado mostramos como quedan las referencias a estilos en nuestra página:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta name="format-detection" content="telephone=no" />
<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1,
width=device-width, height=device-height, target-densitydpi=device-dpi" />
<link rel="stylesheet" type="text/css" href="css/normalize.css" />
<link rel="stylesheet" type="text/css" href="css/jquery.mobile-1.3.2.min.css" />
<link rel="stylesheet" type="text/css" href="css/miTema.css" />
<link rel="stylesheet" type="text/css" href="css/app.css" />
<title>R.M. El Escorial</title>
</head>
```

Figura 93: Funcionalidades. Visualizar Google-maps. Estilos CSS

Alumno: Jesús Villegas Guerra

Y por otro los scripts quedarían como siguen:

```

<!-- Scripts de la aplicación *****
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/appConfig.js"></script>
<script type="text/javascript" src="js/jquery.mobile-1.3.2.min.js"></script>
<script type="text/javascript" src="cordova-2.5.0.js"></script>
<script type="text/javascript" src="js/connection.js"></script>
<script type="text/javascript" src="js/media.js"></script>
<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?sensor=true"></script>
<script type="text/javascript" src="js/gmaps.js"></script>
<script type="text/javascript" src="js/app.js"></script>

```

Figura 94: Funcionalidades. Visualizar Google-maps. Scripts

Viéndose en la figura anterior que además de los archivos de scripts situados en el fichero "js", además hemos tenido que integrar una librería de google de una fuente externa:

```

<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?sensor=true"></script>

```

El código que hemos integrado en js/app.js se muestra en la figura siguiente.

```

/*
 * Al pulsar el botón "Ver Mapa" se realiza una conexión a internet. Sin embargo
 * antes de realizar la conexión se llama a la función check_network(). En caso
 * de que no haya conexión mostrará el mensaje que se indica en check_network() y
 * no mostrará nada más.
 *
 * En caso de que check_network() tenga conexión, se llamará a la función
 * geolocalizar() y se mostrará el mapa.
 */
$('#verMapa').on("click",function(){
function geolocalizar(){
    GMaps.geolocate({
        success: function(position) {
            var map = new GMaps({
                el: '#map',
                lat: 40.589257,
                lng: -4.147727
            });
            map.addMarker({lat:40.589257,lng:-4.147727});
        },
        error: function(error) {
            alert('Fallo en la Geolocalización: '+error.message);
        },
        not_supported: function() {
            alert("Su navegador no soporta la geolocalización");
        }
    });
};

check_network();
geolocalizar();
});

```

Figura 95: Funcionalidades. Visualizar Google-maps.js/app.js. Ver Mapa.

Alumno: Jesús Villegas Guerra

De tal forma que hemos creado un evento de ratón de modo que al pulsar el botón “Ver Mapa” realice unas acciones de terminadas con el código:

- `$('#verMapa').on("click",function () {});`

Una vez en la función, se llamará a otra llamada “check_network()” situada en **js/connection.js**. En caso de que no haya conexión nos mostrará un mensaje como el que se indica en la figura 92 (¡No tiene conexión a internet!).

El código de js/connection.js es el siguiente:

```
function check_network() {
    var networkState = navigator.network.connection.type;

    var states = {};
    states[Connection.UNKNOWN] = 'Conexión desconocida';
    states[Connection.ETHERNET] = 'Conexión Ethernet';
    states[Connection.WIFI] = 'Conexión WIFI';
    states[Connection.CELL_2G] = 'Conexión 2G';
    states[Connection.CELL_3G] = 'Conexión 3G';
    states[Connection.CELL_4G] = 'Conexión 4G';
    states[Connection.NONE] = '¡ No tiene conexión a internet !';

    $('#noMap,#noMap2').html('');

    if(states[networkState]=='¡ No tiene conexión a internet !'){
        $('#noMap').html(states[networkState]);
        $('#noMap2').html('<p>¡Lo sentimos! No podemos mostrarle el mapa.</p>');
    }
}
```

Figura 96: Funcionalidades. Ver Google-maps. Código check_network ()

Por tanto, en cuanto se entra en la función se modifica en DOM en aquellos lugares que se indican en los selectores: #noMap y #noMap2. En este caso se indica que se dejen en blanco esos lugares. Y si se cumple la condición de que no haya conexión se mostrarán en los lugares de los selectores los mensajes indicados en las funciones (Figura 47),

Alumno: Jesús Villegas Guerra

Continuando con la función `$('#verMapa'). on ("click", function (){})`, si no se ha mostrado el mensaje de no conexión, se llamará a la función `geolocalizar()`, con lo que se mostrará el mapa.

Otro aspecto a tener en cuenta es la creación del objeto para que se muestre el mapa. Esto conlleva inclusión de reglas CSS y media queries.

El código que hemos incluido en el fichero index.html es como sigue:

```
<p>¿Quiere ver aquí la situación del Monasterio en Google Map? (Necesita conexión a internet)</p>
<a href="#" data-role="button" data-inline="true" id="verMapa" >Ver Mapa</a>
<div id="noMap" ></div>
<div id="noMap2" ></div>
</div>
<div class="google-maps" id="map" style="position: relative; background-color: transparent;overflow: hidden;">
  <object width=600 height=400 ></object>
</div>
```

Figura 97: Funcionalidades. Ver Google-maps. Object.

Hemos tenido en cuenta dos lugares con las etiquetas `<div>` con identificadores `id="noMap"` e `id="noMap2"` que serán rellenados, como hemos visto antes en caso de que se tenga que mostrar algún mensaje.

Se ha creado un espacio para incluir el mapa con la etiqueta `<object>`, según recomendación de W3C en vez de crearla con `iframe`. Se le ha dado una anchura y altura específicas, aunque estas las cambiaremos con reglas CSS para que se adapten a cualquier dispositivo.

Las reglas CSS que hacen posible que el objeto (mapa de Google) se adapte a cualquier dispositivo son:

```
.google-maps {
  position: relative;
  margin: 0 2em 0 0em;
  padding-bottom: 65%;
  height: 0;
  overflow: hidden;
}
.google-maps object {
  position: absolute;
  top: 0;
  left: 0;
  width: 100% !important;
  height: 100% !important;
}
```

```
@media only all and ( min-width:480px){
  .google-maps {
    margin: 0 3em 0 2em;
  }
}
```

```
@media only all and ( min-width:650px){
    .google-maps {
    margin: 0 4em 0 4em;
    }
}
@media only all and ( min-width:1150px){
    .google-maps {
    margin: 0 5em 0 5em;
    }
}
```

Figura 98: Funcionalidades. Ver Google-maps. CSS. Object.

El ancho y el alto se consiguen con el valor 100% en width y height, además se le han añadido márgenes según el ancho del dispositivo.

8.5.10. Acceder a página externa

Desde la página de horarios se puede acceder a una página externa al pulsar el botón de “Consultar cierres-festivos”.

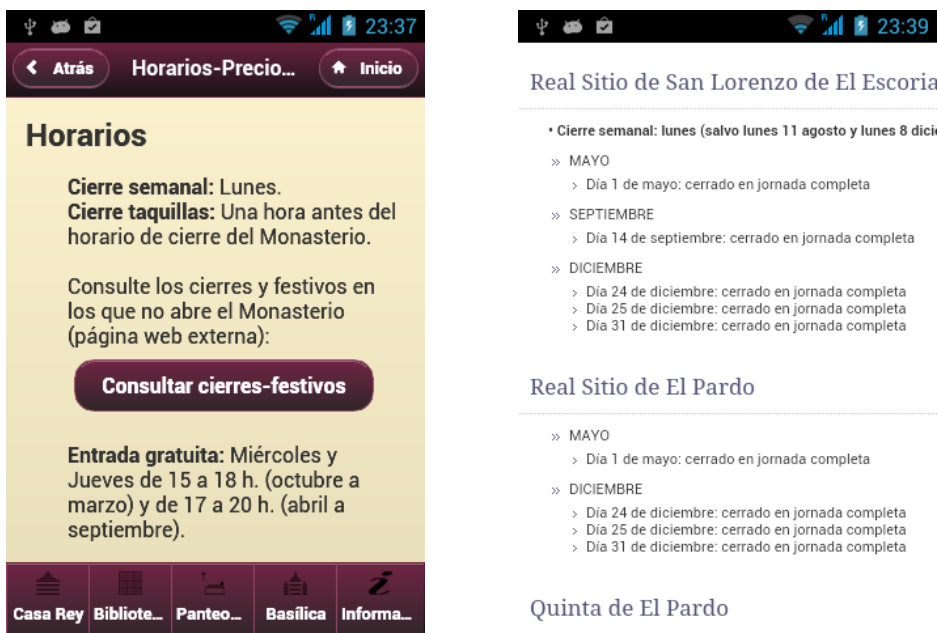


Figura 99: Funcionalidades. Acceder página externa. Pantallas

Para realizar el botón se ha añadido el siguiente código a la página index.html:

```
<a href="http://www.patrimoniacional.es/Home/Noticias/cierres.aspx#escorial" id="cierre" data-role="button" data-inline="true" data-ajax="false">Consultar cierres-festivos</a>
```

Figura 100: Funcionalidades. Acceder página externa. Código

Alumno: Jesús Villegas Guerra

Con data-role="button", se indica dentro de la etiqueta <a> que se crea un botón. Con data-inline="true" hace que el botón no ocupe toda la pantalla y finalmente desactivamos Ajax para que pueda ir a la página externa con data-ajax="false".

8.6. PhoneGap Build: Desarrollo multiplataforma

PhoneGap no es exactamente lo mismo que PhoneGap Build.

Phonegap es un producto de código abierto, open source, gratuito, generado por la empresa Nitobi que desarrollaba aplicaciones móviles para Android, iOS. Descubrieron que la compatibilidad entre plataformas distintas era difícil de resolver con las tecnologías nativas como Objective C para iOS o Java para Android. Por tanto, Nitobi creó Phonegap para que usando un mismo código basado en HTML5, CSS y Javascript se pudiera desarrollar una app para distintas plataformas a la vez (Android, iOS, etc). Para usar PhoneGap en nuestros desarrollos es necesario tener incluida en nuestras librerías la última versión estable de Phonegap, ahora llamado Cordoba con su archivo javascript.

De otra forma, **PhoneGap Build**, también introducido por Nitobi, es un servicio que permite compilar las aplicaciones en la nube, resultando aplicaciones híbridas.

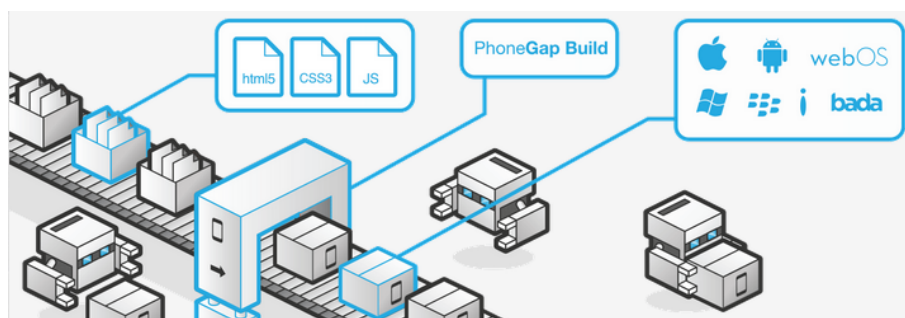


Figura 101: PhoneGap Build. Imagen de compilación
Imagen extraída de <https://build.phonegap.com/>

Al compilar nuestro código con PhoneGap Build, se empaqueta la aplicación para cualquiera de las plataformas soportadas (iOS, Android, Windows-Phone, Blackberry, Web Os y Symbian).

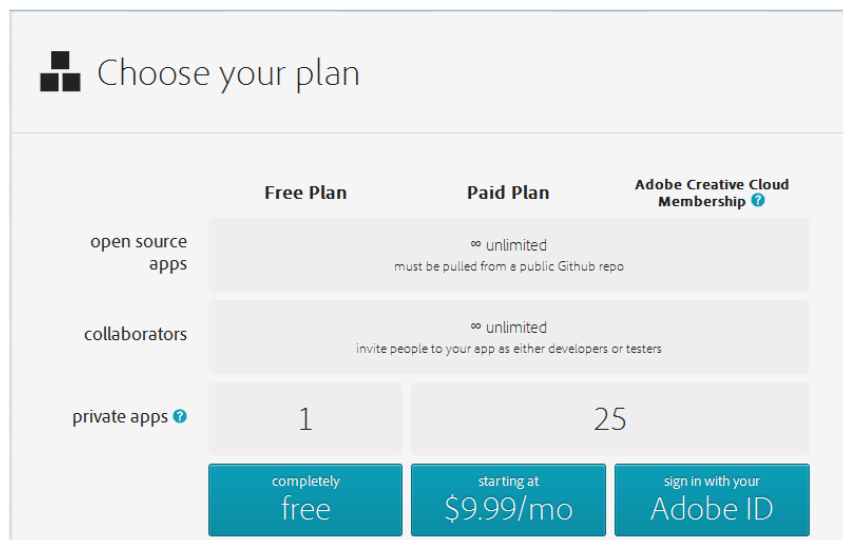


Figura 102: PhoneGap Build. Planes del servicio.

El servicio que se ofrece es gratuito, aunque solo se puede realizar una aplicación que debe estar alojada en un repositorio público llamado Github, en caso necesitar más, sería necesario pagar una cuota mensual. Sin embargo, en nuestro caso con la cuenta gratuita será suficiente.

Una vez registrados solo será necesario subir el código al compilador cumpliendo con unos requisitos y es que la carpeta que integre todos los archivos CSS, html, javascript, etc, debe contener un archivo llamado index.html y, si se quiere, otro archivo llamado config.xml que nos servirá para dar unas especificaciones concretas al compilador como nombre del autor, descripción de la aplicación, habilitar funcionalidades del dispositivo, indicar donde se encuentran las imágenes para el icono o pantalla de bienvenida, si tenemos preferencia para que la aplicación se instale en el dispositivo o en tarjeta externa, la versión de la app o la versión de PhoneGap a usar, entre otros.

Sin embargo antes de que nuestra aplicación se pueda compilar ha sido necesario subirla a Github que es un repositorio para el control de versiones ya que PhoneGap Build toma los archivos de este repositorio para compilar.

Nuestro repositorio queda de la siguiente forma en Github:

Alumno: Jesús Villegas Guerra

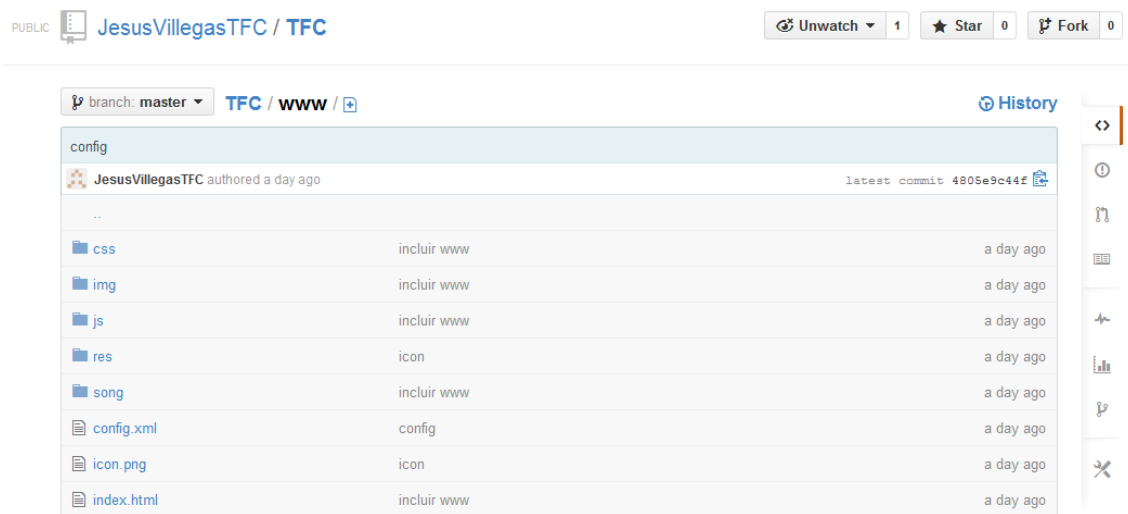


Figura 103: PhoneGap Build. Repositorio Github

Por tanto nuestro proyecto una vez compilado en Phonegap queda de la siguiente forma:

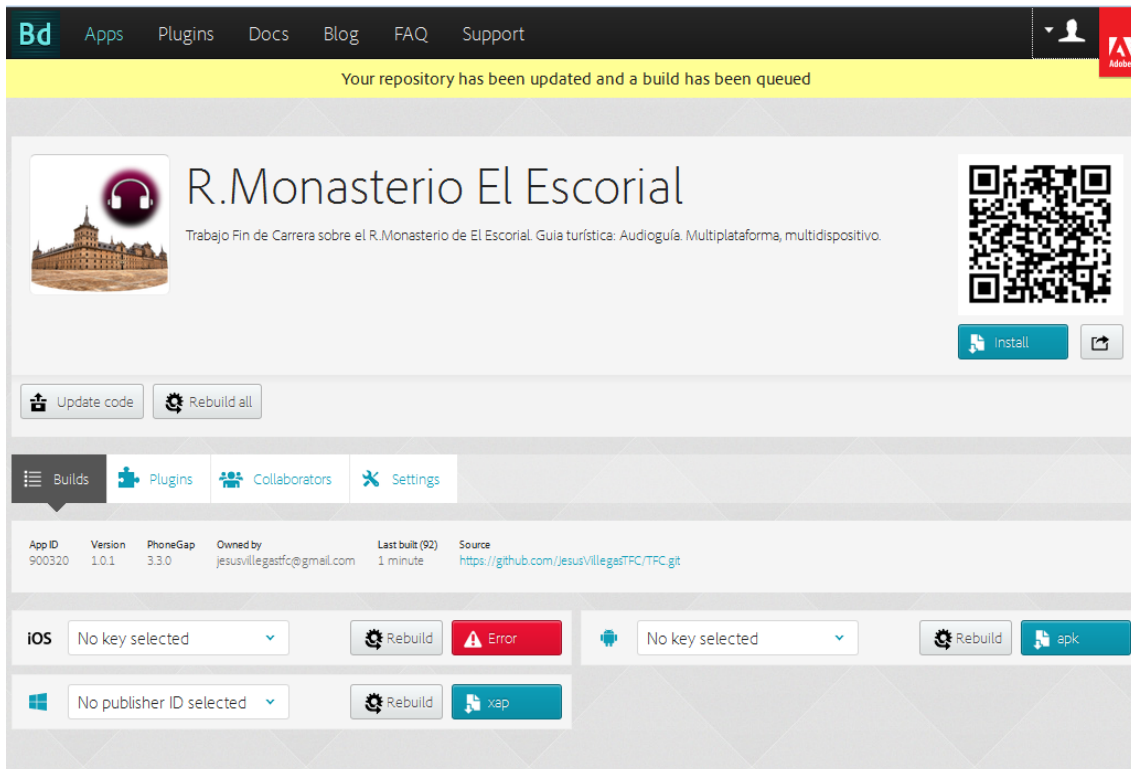


Figura 104: PhoneGap Build. App empaquetada

Para realizar el **config.xml**, necesario si se quiere compilar para varias plataformas me he basado en la api y en el modelo de la dirección url siguiente:

<https://github.com/phonegap/phonegap-start/blob/master/www/config.xml>

El config.xml que genera Eclipse y el config.xml que hemos realizado para compilar en Phonegap difieren, no es el mismo. Hemos tenido que crearlo nuevo.

Para obtener el nombre de la aplicación y su descripción, que aparecen en la figura 104 al comienzo, he usado el siguiente código XML:

```
<name>R.Monasterio El Escorial</name>

<description>
  Trabajo Fin de Carrera sobre el R.Monasterio de El Escorial.
  | Guia turística: Audioguía. Multiplataforma, multidispositivo.
</description>
```

Figura 105: PhoneGap Build. Config.xml. Name-Description

La imagen que aparece al comienzo de la figura 104, es la imagen que por defecto será el icono de la aplicación. Ha sido necesario crearlo en un tamaño de 512x512 pixeles y hasta 1024MB y situar el archivo .png llamado icon.png dentro de la raíz de la aplicación, según las indicaciones que da la API de PhoneGap Build. Por un lado en la figura 103 se puede ver donde está situado el archivo icon.png (directorio raíz) y por otro el código utilizado para integrar todos los iconos, launcher y splash-screen usados para todos los formatos es el siguiente:

```
<!-- Define app icon for each platform. -->
<icon src="icon.png" />
<icon src="res/icon/android/icon-36-ldpi.png" gap:platform="android" gap:density="ldpi" />
<icon src="res/icon/android/icon-48-mdpi.png" gap:platform="android" gap:density="mdpi" />
<icon src="res/icon/android/icon-72-hdpi.png" gap:platform="android" gap:density="hdpi" />
<icon src="res/icon/android/icon-96-xhdpi.png" gap:platform="android" gap:density="xhdpi" />
<icon src="res/icon/ios/icon-57.png" gap:platform="ios" width="57" height="57" />
<icon src="res/icon/ios/icon-72.png" gap:platform="ios" width="72" height="72" />
<icon src="res/icon/ios/icon-57-2x.png" gap:platform="ios" width="114" height="114" />
<icon src="res/icon/ios/icon-72-2x.png" gap:platform="ios" width="144" height="144" />
<icon src="res/icon/windows-phone/icon-48.png" gap:platform="winphone" />
<icon src="res/icon/windows-phone/icon-173.png" gap:platform="winphone" gap:role="background" />

<!-- Define app splash screen for each platform. -->
<gap:splash src="res/screen/android/screen-ldpi-portrait.png" gap:platform="android" gap:density="ldpi" />
<gap:splash src="res/screen/android/screen-mdpi-portrait.png" gap:platform="android" gap:density="mdpi" />
<gap:splash src="res/screen/android/screen-hdpi-portrait.png" gap:platform="android" gap:density="hdpi" />
<gap:splash src="res/screen/android/screen-xhdpi-portrait.png" gap:platform="android" gap:density="xhdpi" />
<gap:splash src="res/screen/ios/screen-iphone-portrait.png" gap:platform="ios" width="320" height="480" />
<gap:splash src="res/screen/ios/screen-iphone-portrait-2x.png" gap:platform="ios" width="640" height="960" />
<gap:splash src="res/screen/ios/screen-ipad-portrait.png" gap:platform="ios" width="768" height="1024" />
<gap:splash src="res/screen/ios/screen-ipad-landscape.png" gap:platform="ios" width="1024" height="768" />
<gap:splash src="res/screen/windows-phone/screen-portrait.jpg" gap:platform="winphone" />
```

Figura 106: PhoneGap Build. Config.xml. Launcher, Splash-Screens

Otro aspecto que recoge la figura 104 es la versión de la app y la de Phonegap utilizada que es tomada del código siguiente:

```
<widget xmlns      = "http://www.w3.org/ns/widgets"
        xmlns:gap   = "http://phonegap.com/ns/1.0"
        id          = "edu.uoc.tfc"
        version     = "1.0.1">

<!-- Customize your app and platform with the preference element. -->
<preference name="phonegap-version"       value="3.3.0" />
<preference name="orientation"           value="default" />

<preference name="ios-statusbarstyle"     value="black-opaque" />
<!-- asn't go beneath the status bar -->

<preference name="android-installLocation" value="preferExternal" />
```

Figura 107: PhoneGap Build. Config.xml. Versiones-Orientacion-Instalación

Como se observa en la figura 107, se incluye el nombre del paquete de la aplicación con el id="edu.uoc.tfc", la versión de la app, la versión de Phonegap (3.3.0) y la preferencia de instalación de la app que la queremos externa, es decir, en tarjeta de memoria SD, si la tuviera el Smartphone.

También hemos incluido en el código lo que nuestra aplicación necesitará del Smartphone, ya que para que se puedan usar las funcionalidades del teléfono es necesario indicarle las requeridas que en nuestro caso son:

```
<!--
  Enable individual API permissions here.
  The "device" permission is required for the 'deviceready' event.
-->
<feature name="http://api.phonegap.com/1.0/device" />
<!-- feature name="http://api.phonegap.com/1.0/battery"/-->
<!--feature name="http://api.phonegap.com/1.0/camera"/-->
<!--feature name="http://api.phonegap.com/1.0/contacts"/-->
<feature name="http://api.phonegap.com/1.0/geolocation"/>
<feature name="http://api.phonegap.com/1.0/media"/>
<feature name="http://api.phonegap.com/1.0/network"/>
<feature name="http://api.phonegap.com/1.0/notification"/>
```

Figura 108: PhoneGap Build. Config.xml. Funcionalidades requeridas

Otro aspecto que aparece en la figura 104 es que no están presentes todos los posibles dispositivos para los que se puede crear la aplicación. Solamente están presentes iOS, Android y Windows-Phone. Faltan por tanto BlackBerry, Bada y Symbian. Esto es debido a que así se lo hemos indicado en nuestro config.xml, al señalar que solo creamos iconos para Android, iOS y Windows-Phone.

Finalmente comentar que esta parte ha sido realizada con Dreamweaver. Se ha copiado el fichero “www” en un proyecto nuevo y se ha ido actualizando el repositorio Github, usado por PhoneGap Build, con la aplicación instalada para Windows 7, de tal forma que cada vez que se hacía un cambio en algún fichero el recuadro “No uncommitted changes” cambiaba a “Uncommitted changes show”. Con lo que al pulsar sobre ese espacio el botón “sync” cambiaba. Y al oprimir el botón sync se realizaban los cambios que eran subidos a nuestro repositorio Github, listo para poder compilarse y crear nuestra aplicación.

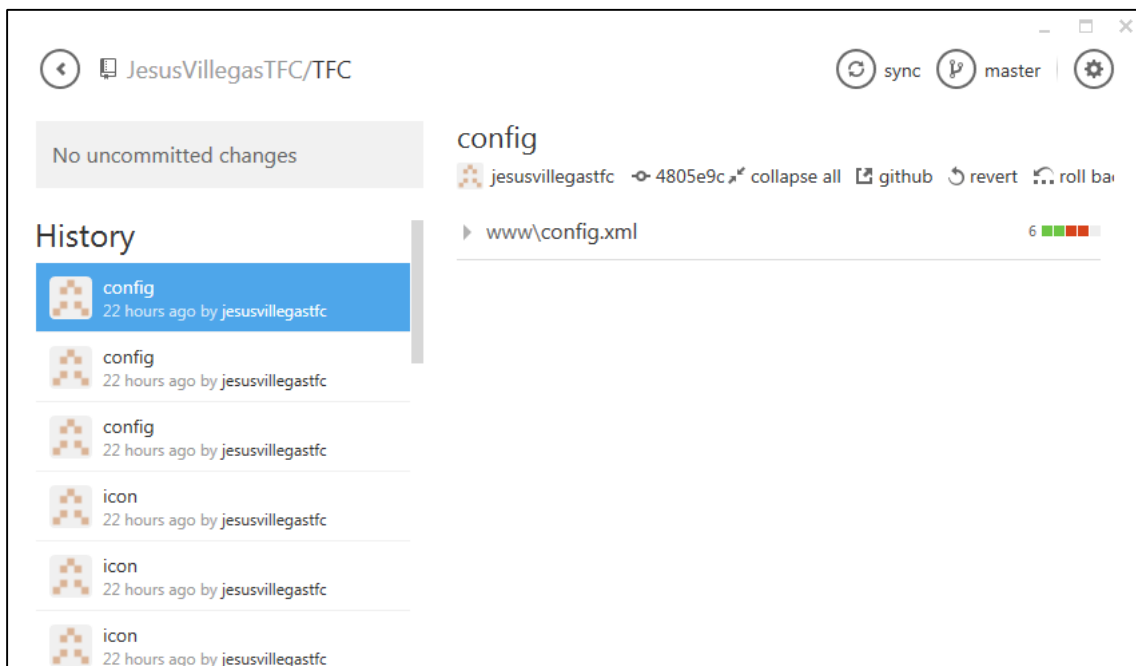


Figura 109: PhoneGap Build. Aplicación Github instalada en Windows 7.

8.7. Google Play: Distribución de la aplicación.

Una vez compilada la aplicación, bien mediante Eclipse o bien por PhoneGap Build, se genera un archivo “.apk”.

El archivo generado “.apk” es para Android y por tanto solamente podremos distribuirlos en canales habilitados para ello como Google Play o Amazon Appstore, pero hay muchos más.

No subiremos el archivo a Google Play ya que no es gratuito, hay que pagar 25\$ por una sola vez. En contraposición a este aspecto en la Apple Store hay que pagar 99\$ anuales.

No obstante, para subir una aplicación a Google Play debe estar firmada para que solo el que la ha realizado pueda modificarla o actualizarla.

PhoneGap Build nos permite firmar la aplicación. Para ello es necesario crear un almacén de claves (KeyStore) en nuestro ordenador.

Deberemos tener instalado el JDK y mediante consola de Windows utilizar el comando “keytool”, generando la KeyStore de la manera siguiente:



```
Administrador: C:\Windows\system32\cmd.exe
C:\Program Files (x86)\Java\jdk7.0_13\bin>keytool -genkey -v -keystore mi-almacen-de-claves.keystore
-alias mi-alias -keyalg RSA -keysize 2048 -validity 365
```

Figura 110: Google Play. KeyStore

Donde “mi-almacen-de-claves” es el nombre dado al almacén (lo podemos cambiar), “mi-alias” es el alias que debemos cambiar ya que será el que posteriormente nos pida Phonegap y 365 es la validez en días de la clave, normalmente se pone 10000.

Nos pedirá a continuación que pongamos una contraseña inventada y que la repitamos:

**Introduzca la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:**

A continuación preguntará una serie de cuestiones y finalmente se generará la clave que será guardada en `jdk/bin/mi-almacen-de-claves.keystore`. Esta clave luego la tendremos que buscar cuando PhoneGap Build nos la pida.

Finalmente, nos iremos a PhoneGap Build, pulsaremos "Add a key" y nos preguntará por un título para nuestra llave (Title), más tarde nos pedirá nuestro alias (el que hayamos puesto en el comando por consola) y por último elegiremos "examinar", buscando la keystore en la dirección comentada anteriormente (`jdk/bin/...`) y obtendremos el paquete compilado ".apk" con certificado de la siguiente forma:

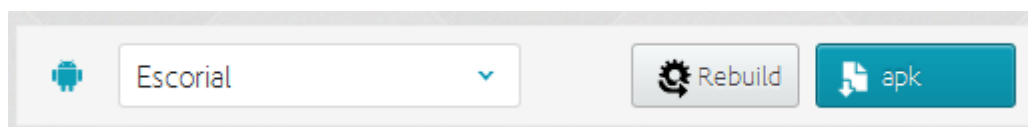


Figura 111: Google Play. Apk compilada con certificado

Ya tendremos nuestra aplicación lista para llevarla a Google Play. Aunque hay que cumplir unos requisitos para ello:

- Tener una cuenta de desarrollador, con un precio de 25\$.
- La aplicación no debe de ocupar más de 50MB.
- Indicar el título de la aplicación.
- Añadir una descripción de la aplicación.
- Mostrar al menos dos capturas de pantalla que sirvan como vista previa de la app.
- Proporcionar un icono de 512x512 pixeles, alta resolución.
- Indicar el idioma y el país al que va destinada la aplicación.
- Especificar el tipo de aplicación, categoría y clasificación del contenido.
- Y aceptar algunas políticas de privacidad.

9. CONCLUSIONES

Se ha creado una aplicación multidispositivo y se ha intentado que sea además multiplataforma con la compilación en PhoneGap Build.

Alumno: Jesús Villegas Guerra

Desde mi punto de vista el desarrollo del proyecto: Análisis, diseño e implementación ha sido realizado satisfactoriamente. Sin embargo me hubiera gustado poder haberla probado en otras plataformas como iOS o Windows-Phone, y dispositivos, sin embargo no me ha sido posible al carecer de medios para poder hacer estas comprobaciones.

Este trabajo me ha servido para aprender varios lenguajes que no conocía como javascript, jQuery y jQuery Mobile (basados en javascript), CSS3 y mejorar mis conocimientos de HTML con HTML5, además de poder incluir en mi curriculum este proyecto como mejora en mi carrera profesional y haber utilizado recursos estudiados a lo largo de los años de carrera como Ingeniería del Software.

Los objetivos que quería alcanzar al comienzo de este proyecto han sido conseguidos con éxito y con creces. Tengo que reconocer que el nivel de dificultad encontrado ha sido mayor del que me esperaba al comenzar el trabajo ya que no solamente he tenido que aprender un lenguaje como HTML5 o CSS3 sino que para crear aplicaciones profesionales y con diseño, nos tenemos que basar en otros lenguajes y conocer la forma de aunarlos.

Por tanto he conseguido los siguientes objetivos:

- Mejorar con la práctica de HTML5 y CSS3. Tenía algunos conocimientos básicos de HTML y CSS de nivel 2, pero tener que utilizar las nuevas etiquetas tanto de HTML5 como reglas de CSS3 me ha permitido conocer y explorar nuevas posibilidades.
- Aprender lenguajes como JavaScript, jQuery, jQuery Mobile. He visto la gran potencia que poseen y como son capaces de manejar el DOM. Aunque he tenido bastantes dificultades en entender como jQuery Mobile accedía al DOM, usar los eventos y como poder personalizar el CSS que posee. Ya que el uso de jQuery Mobile es rápido cuando se limita al uso simple, pero cuando se quiere modificar su contenido CSS para adaptarlo a nuestro diseño, hay que estudiar a fondo las reglas CSS que permiten hacer posible su modificación.
- Poder conocer cómo realizar aplicaciones multiplataforma con PhoneGap Build.
- He aprendido como usar la librería javascript de Phonegap y cómo poder usar algunos de los recursos internos del Smartphone como la geolocalización, internet o notificaciones.

- En el aspecto del diseño he podido aprender la base de la utilización de la nueva regla @media para hacer la aplicación adaptable a distintos dispositivos, por lo que con estas reglas y con las propias que tenía jQuery Mobile, se hace posible que la app sea multidispositivo.
- Crear a partir de una aplicación que usa lenguajes destinados para la web (HTML5, CSS3, javascript) una app que pueda instalarse en smartphones y que use los recursos disponibles en el mismo.

10. GLOSARIO

- **App:** Es una aplicación de software que se instala en dispositivos móviles o tablets para ayudar al usuario en una labor concreta, ya sea de carácter profesional o de ocio y entretenimiento.
- **App:** La palabra app corresponde al término inglés de “application”, del que coge las tres primeras letras. Por tanto una app es una aplicación de software que se instala en dispositivos móviles o tablets para ayudar al usuario en una labor concreta, ya sea de carácter profesional o de ocio y entretenimiento.
- **Diseño conceptual:** Hace referencia al perfil de usuario, Personas y escenarios que nos acercan a los usuarios y sus motivaciones, objetivos y situaciones de uso. Es una técnica que nos sirve para entender y analizar a los usuarios y el uso que hacen de los sistemas interactivos, utilizado para orientarnos en el diseño de la aplicación.
- **DOM:** Document Object Model (Modelo de Objetos de Documento) .Define la estructura lógica de los documentos HTML y el modo en que se accede y manipula un documento. Los programadores pueden construir documentos, navegar por su estructura, y añadir, modificar o eliminar elementos y contenido. La estructura es de tipo árbol y está compuesta de Document: nodo raíz del que derivan todos los demás como “Element” que es cada una de las etiquetas HTML, es el único nodo que puede contener atributos y del que pueden derivar otros nodos. “Attr” es un atributo de una etiqueta que genera un nodo Attr., el cual contiene también su valor (como una propiedad). Es hijo del nodo element (etiqueta) que lo contiene. “Text” que contiene el texto encerrado por una etiqueta HTML (hijo del nodo Element) y “Comment” que son los comentarios incluidos en la página HTML y también generan sus nodos.

- **Escenario:** Es la descripción de una persona en una situación de uso de la aplicación interactiva con unos objetivos concretos. Se incluye en la descripción el contexto de la acción y la secuencia de acciones realizadas. El estudio de estos elementos nos ayudan a explorar ideas y aspectos en el diseño.
- **Etiqueta:** El lenguaje HTML es un lenguaje de marcas, estas marcas serán fragmentos de texto destacado de una forma especial que permiten la definición de las distintas instrucciones de HTML, tanto los efectos a aplicar sobre el texto como las distintas estructuras del lenguaje. A estas marcas las denominaremos etiquetas y serán la base principal del lenguaje
- **Flujos de interacción:** Es la representación mediante un diagrama de flujo por el cual con figuras básicas se puede expresar las posibles respuestas del sistema a la interacción del usuario con la aplicación. Jesse James Garrett perfeccionó este sistema con un lenguaje visual claro y sencillo.
- **Front-End:** Parte del software que interactúa con el usuario.
- **Market Places:** Son lugares donde se suben las aplicaciones creadas por sus desarrolladores para poder ser vendidas o utilizadas por los usuarios en sus distintos smartphones o tablets. Algunos markets son Google Play, Amazon App Store, Opera Mobile Store, Samsung App, LG Smart Mundial,
- **Normalize:** Es un archivo pequeño de tipo CSS que proporciona una mejor consistencia entre navegadores en el estilo predeterminado de los elementos de HTML. Es una alternativa moderna al tradicional al reseteo de HTML.
- **Perfil de usuario:** Agrupaciones de usuarios según unas características como la edad, sexo, aspectos sociodemográficos, etc.
- **Prototipo de alta fidelidad:** Trata de construir un modelo de aplicación lo más próximo al sistema que se diseña y desarrolla. Se utiliza para evaluar los aspectos funcionales y de usabilidad bien con un experto en usabilidad o con test a usuarios.
- **Prototipo:** Consiste en la creación de un modelo de aplicación que se diseña. Simulan o ya tiene parte de las funcionalidades requeridas y que se utiliza para hacer pruebas que de otro modo no se podrían realizar hasta que el proyecto estuviese terminado. Se quiere representar no la totalidad de la aplicación sino aquellas partes que se quieren estudiar por su especial interés.
- **Smartphones:** Teléfono móvil de última generación con capacidad de comunicación wifi, 3G,4G, bluetooth, mensajería, e-mails que puede ejecutar

aplicaciones diversas.

- **Splash-screen:** Es una pantalla de inicio o presentación que aparece con anterioridad a la aplicación mientras ésta se carga.
- **Test de usabilidad o evaluación:** Son test con usuarios que nos proporcionan información de carácter cualitativo, interpretando los hechos realizados por los usuarios respecto de la aplicación estudiada, con estos test vemos si la interfaz está bien diseñada y las dificultades para lograr los objetivos propuestos.
- **W3C:** El World Wide Web Consortium es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.
- **Whatsapps:** Aplicación para smartphones con la que los usuarios pueden comunicarse por escrito, enviar mensajes, fotos, archivos.
- **WHATWG:** El Web Hypertext Application Technology Working Group, es una comunidad de personas interesadas en la evolución de HTML y las tecnologías conexas. Fue fundado por integrantes de Apple, Fundación Mozilla y Opera Software.
- **WordPress:** Aplicación web con la que los usuarios puede realizar paginas web o blogs. Desarrollada por Google.

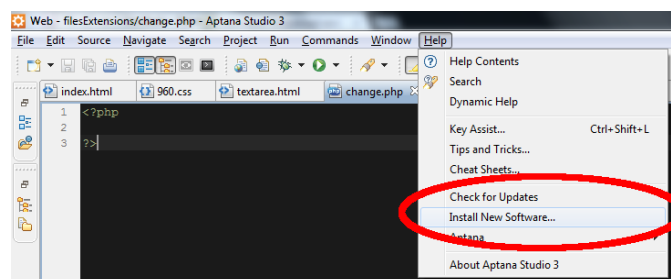
11. ANEXOS

11.1. Cómo preparar Windows para trabajar con Android+Phonegap

Dejo un enlace para la instalación de Eclipse, SDK y Phonegap:

<http://www.neoteo.com/como-instalar-phonegap-en-eclipse/>

Cuando se termine esa instalación se incluirá Aptana como plugin de Eclipse:



Y una vez en la pantalla de instalación se añade la dirección url:

<http://download.aptana.com/studio3/plugin/install>

Alumno: Jesús Villegas Guerra

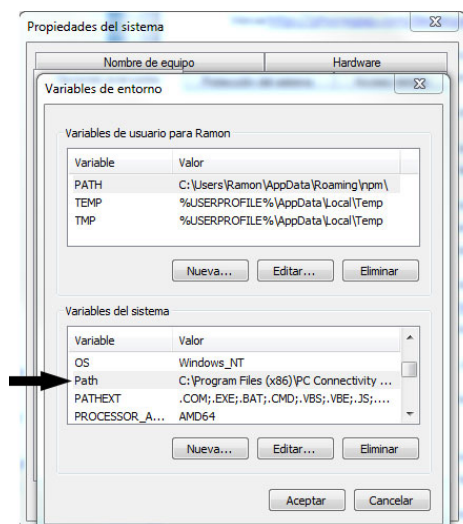
Por tanto, con estos pasos ya tendremos Eclipse funcionando con el SDK y Phonegap.

Un paso muy importante que tenemos que tener en cuenta es la Configuración del entorno que muestro a continuación.

CONFIGURACIONES DE ENTORNO

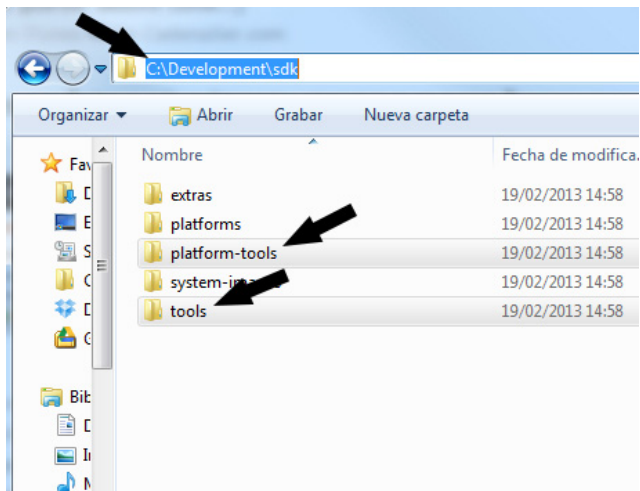
En el punto Setup your PATH environment variable on Windows es muy recomendable seguir todos los pasos, incluso si ya tenéis instalado lo que se os pide.

Id al path de las variables de entorno: haced click en el icono de Equipo y haced click en Configuración avanzada del sistema. Seleccionad variables de entorno, y en la pestaña que os aparecerá, abajo, haced click en Path y Editar.

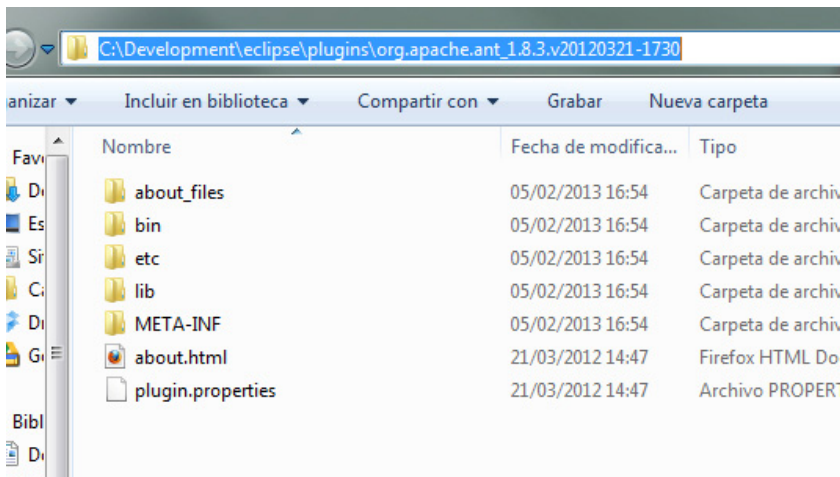


Añadid al final el path para los directorios "platform-tools" y "tools" del sdk de Android. Es decir, si habéis puesto vuestro directorio de desarrollo en C:\Development, podéis poner los que os vienen de ejemplo: ;C:\Development\sdk\platform-tools;C:\Development\sdk\tools. No olvidéis los punto y comas.

Alumno: Jesús Villegas Guerra

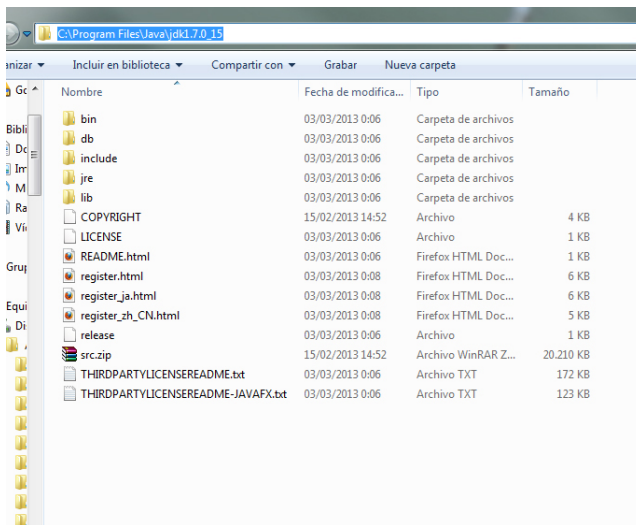


Cread el ANT_HOME que se os indica con el path de la instalación que habéis hecho (si habéis utilizado C:\Development, será C:\Development\eclipse\plugins\org.apache.ant_1.8.3.v201301120609\.

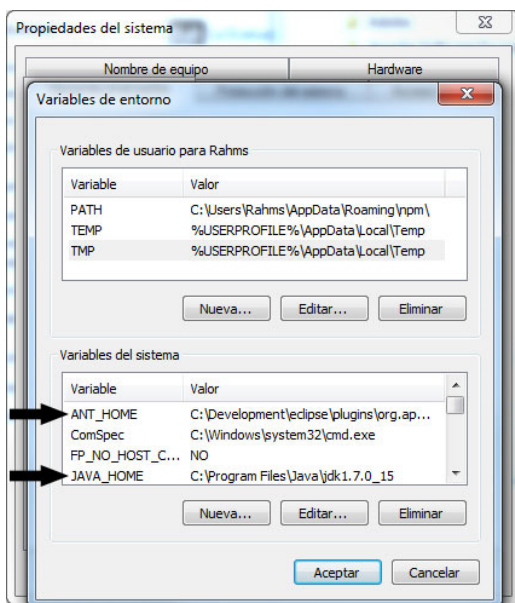


Cread el JAVA_HOME con el path de la instalación que habéis hecho y el directorio bin (típicamente C:\Program Files\Java\jdk1.7.0_02).

Alumno: Jesús Villegas Guerra



Haced de nuevo click en Path y Editar. Añadid en el path las variables:
;%JAVA_HOME%\bin\;%JAVA_HOME%\lib\;%ANT_HOME%\bin;

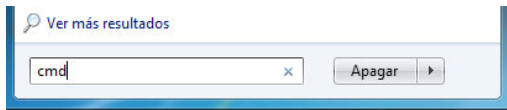


Una vez hechos estos pasos, ya podéis crear vuestro proyecto Cordova desde Eclipse. Si queréis hacerlo desde la consola (aunque parece que las últimas versiones del SDK son bastante estables) pasad al paso 3 de este tutorial.

NUEVO PROYECTO DESDE CONSOLA

Alumno: Jesús Villegas Guerra

Abrimos la consola de Windows 7.

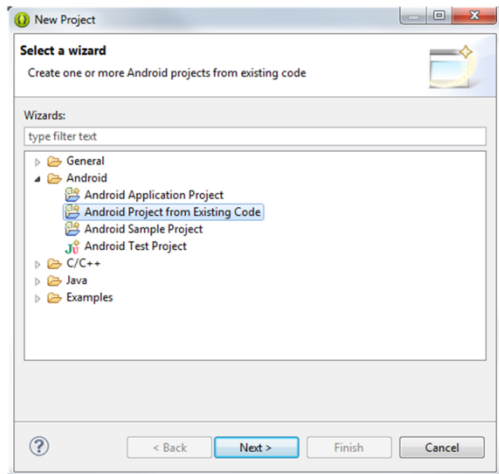


Tenemos que situarnos dentro del directorio que tengamos Phonegap y llegar hasta la carpeta llamada lib\android\bin. Una vez situado en la carpeta "bin" ya podremos crear nuestro proyecto introduciendo en consola:

- > create C:\Android\TFC edu.uoc.tfc tfc, donde :
- create es la sentencia que crea el nuevo proyecto.
- C:\Android\TFC es el directorio donde creará el proyecto.
- edu.uoc.tfc es el nombre del paquete del proyecto de Eclipse.
- tfc es como se llama el proyecto que será creado dentro del directorio TFC.

Si todo ha ido bien, os creará el proyecto sobre el que trabajaréis.

Para continuar, abrid el Eclipse que habéis instalado en el C:\Android y abrid un proyecto desde un proyecto existente.



Se elige el directorio TFC que contendrá el proyecto listo para comenzar con las librerías de Phonegap para poder usarse y compilar la aplicación.

11.2. Copia de la entrevista a los usuarios

FICHA DE USUARIO NUM _____

Edad ____

Alumno: Jesús Villegas Guerra

Sexo___

Profesión_____

Formación_____

Miembros en la familia_____

Nivel de idioma Inglés_____

Uso de internet: ¿Qué dispositivos usa para conectarse a internet?

¿Qué nivel de conocimientos informáticos posee?_____

¿Dónde se realiza el test?_____

11.3. Hoja presentada a los usuarios: Tarea del Test a la competencia.

Usuario número_____

TAREAS A REALIZAR

Vamos a realizar una serie de tareas en las páginas que les voy a indicar en el ordenador a continuación.

Se realizarán sobre dos aplicaciones de Smartphone que están ya instaladas y que le indicaré a continuación y en cada una de ellas tendrá que realizar las siguientes tareas:

- Leer un texto sobre algún tema concreto de la aplicación
- Ir al inicio de la aplicación desde la pantalla donde se encuentre.
- Escuchar una audición y pararla (si la aplicación lo posee).

11.4. Hoja presentado a los usuarios: Tarea evaluación del prototipo.

Usuario número _____

TAREAS A REALIZAR

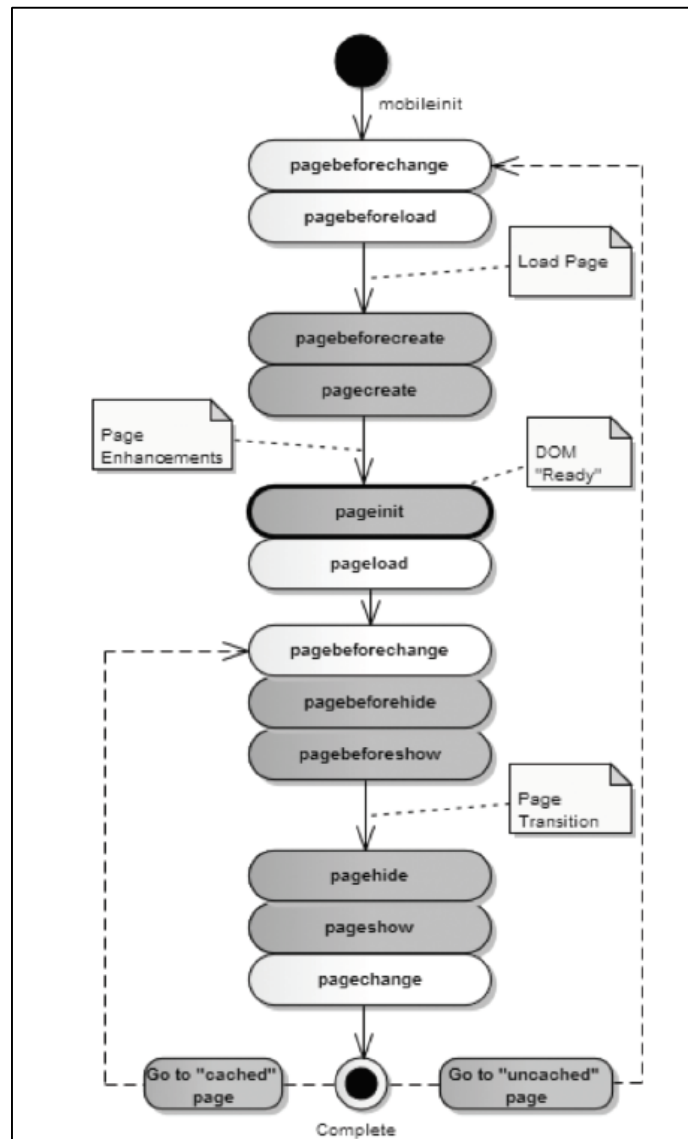
Vamos a realizar una serie de tareas que se harán sobre un prototipo de aplicación que simulará un teléfono móvil en el ordenador. El prototipo no funcionará de la misma manera que una aplicación completa porque le faltarán algunos elementos (como fotos). Las pruebas a realizar sobre la aplicación en cuanto comience serán las siguientes:

Alumno: Jesús Villegas Guerra

- Leer el texto de la página de Biblioteca.
- Escuchar y parar la audición de la página Panteón de los Reyes.
- Ir al inicio de la aplicación.
- Buscar el teléfono del R.M. y llamar.

11.5. Eventos de jQuery Mobile

A continuación presento la página que ha servido de guía para utilizar algunos eventos utilizados en la aplicación y que pueden modificar el ciclo de vida de una página.



Alumno: Jesús Villegas Guerra

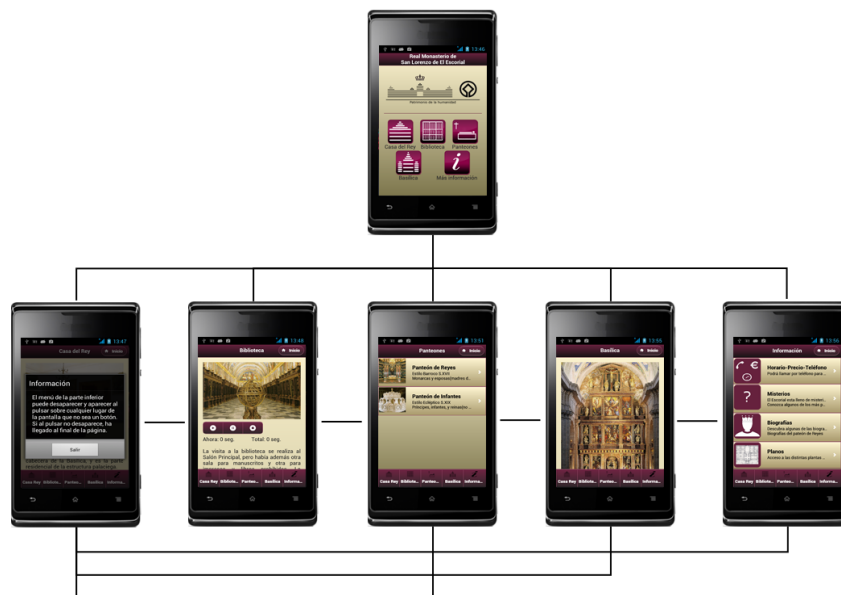
11.6. Pantallas de la aplicación.

La flecha indica el sentido único que puede tener la navegación a través de la app. Si no se indica dirección en la flecha, la navegación se puede realizar en los dos sentidos.

11.6.1. Pantallas de Inicio



11.6.2. Pantallas principales



Alumno: Jesús Villegas Guerra

11.6.3. Pantallas Panteones



11.6.4. Pantallas de información



12. BIBLIOGRAFIA

- <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>. Consulta el 11-03-2014
- <http://www.appio.es/tipos-de-apps/>. Consultado el 11/03/2014.
- <http://blog.aplicacionesmovil.com/aplicaciones-celular/desarrollo-de-aplicaciones-hibridas/> . Consultado el 11/03/2014.
- Broulik, B.,2011. Pro jQuery mobile. Editorial Apress.
- Firtman, M.,2012. jQuery Mobile: Up and Running. Editorial O'Really.
- Minguez García, M^a Carmen (Anales de Geografía, 2007, vol. 27 num 1), Planificación y Gestión Turística en destinos patrimoniales: el caso del Real Sitio del San Lorenzo de El Escorial (Comunidad de Madrid).
- Informe sobre el turista de la Comunidad de Madrid. http://www.hosteltur.com/144000_perfil-turista-visita-comunidad-madrid.html. Consultado el 29/03/2014.
- <http://www.jjg.net/ia/visvocab/spanish.html>. Consultado el 29/03/2014.
- Krug, Steve. (2010). Haz fácil lo imposible. La guía práctica para aficionados sobre cómo encontrar y solucionar problemas de usabilidad. Anaya multimedia.
- Theresa, Neil (2012). Mobile Design Pattern Gallery. O'Reilly.
- <http://jquerymobile.com/>. Página de jQuery con demos, API, etc.
- <http://docs.phonegap.com/en/3.4.0/index.html>. API de Phonegap.
- <http://themroller.jquerymobile.com/?ver=1.3.2>. Página de ThemeRoller. Consultado el 05/03/2014.
- <http://nicolasgallagher.com/about-normalize-css/>. About Normalize.css. Consultado el 12/04/2014.
- <http://www.labnol.org/internet/embed-responsive-google-maps/28333/> : How to Make the New Google Maps Embeds Responsive. Consultado 18-04-2013.
- <http://www.untipodigital.com/disenio-de-moviles-tablets-las-diferentes-densidades/>. Diseño de móviles & tablets: las diferentes densidades. Consultado el 18-04-2014.
- https://developer.mozilla.org/es/docs/CSS/Media_queries. CSS media queries (Mozilla developer neetwork). Consultado el 18-04-2014.
- <http://efe13.com/2013/01/como-crear-mapas-de-google-maps-con-gmaps-js/> . Como crear mapas de Google Maps con Gmaps. Consultado el 25/04/2014.

Alumno: Jesús Villegas Guerra

- <http://hpneo.github.io/gmaps/> . Librería gmaps.js. Consultado el 25/04/2014.
- <http://www.pepfarinblog.com/clave-de-android-para-phonegap-build/> . Generar clave con keytool. Consultado 02-05-2014.
- <http://cyrilmottier.com/2012/05/03/splash-screens-are-evil-dont-use-them/> Splash Screens Are Evil, Don't Use Them! Consultado el 03-05-2014.