



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

“Desarrollo de una aplicación
para visualización de predicciones
meteorológicas en dispositivos
móviles”

*Development of an application for displaying weather
forecasts on mobile devices*

Eliezer Cruz Suárez

La Laguna, 25 de Junio de 2015

D. **Juan Carlos Pérez Darías**, con N.I.F. 45.441.625-L profesor Titular de Universidad adscrito al Departamento de física de la Universidad de La Laguna, como tutor

C E R T I F I C A

Que la presente memoria titulada:

“Desarrollo de una aplicación para visualización de predicciones meteorológicas en dispositivos móviles.”

Ha sido realizada bajo su dirección por D. **Eliezer Cruz Suárez**, con N.I.F. 78.729.019-L.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 25 de junio de 2015.

Agradecimientos

A D. Juan Carlos Pérez Darías por darme su apoyo y tiempo, sin los cuales este Trabajo de Fin de Grado no hubiera sido posible.

A D. Ángela Hernández Delgado por la su gran labor realizada en el Trabajo Fin de Grado “Portal web para información meteorológica”, del cual parte esta aplicación.

A mi familia por el apoyo continuo, sin el cual no hubiera sido posible realizar esta carrera.

A mis amigos en especial a Jonathan Días Palma, por el apoyo y consejo que me ha prestado durante toda la carrera.

Y para finalizar quiero agradecer a Eider Corral Rafols el apoyo prestado durante la realización de este trabajo y también le quiero agradecer el empuje que me ha dado en la recta final de la carrera.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.

Resumen

La meteorología influye en todo momento en las actividades que realiza el ser humano, afectando en la complejidad de su realización. Debido a esto, predecir las condiciones meteorológicas es uno de los retos a los que se ha enfrentado el hombre desde sus inicios. Sin embargo, por su complejidad, la predicción de las condiciones meteorológicas es poco fiable.

En los últimos años, los modelos numéricos para la predicción meteorológica se han convertido en una herramienta indispensable para proporcionar información cada vez más precisa del estado de la atmósfera. El desarrollo de éstos y el incremento de la potencia de cómputo de los ordenadores actuales, han permitido mejorar la eficacia de las predicciones a corto plazo.

El Grupo de Observación de la Tierra y la Atmósfera (GOTA) de la Universidad de La Laguna, haciendo uso de técnicas para el estudio de variables atmosféricas, como los modelos numéricos de área limitada, realiza diariamente predicciones meteorológicas desde el año 2012. Dichas predicciones se realizan a 48 horas para la región de Canarias usando el modelo regional WRF.

Actualmente, GOTA dispone de un “Portal web para información meteorológica”, fruto de otro Trabajo Fin de Grado. El objetivo de este portal es la de proporcionar información a los componentes de GOTA y otros investigadores, pero su funcionalidad no está pensada para usuarios “normales”.

En este contexto, el objetivo de este trabajo consiste en elaborar una aplicación, que seleccione los datos relevantes de este portal y los muestre a usuarios de forma sencilla y amigable. De esta forma, permite visualizar los mapas de predicción para las múltiples variables meteorológicas, además de mostrar una evolución temporal de las principales variables para una localización concreta. También se pretende que esta plataforma sirva como herramienta de consulta rápida para el equipo de investigación, ya que será capaz de mostrar las predicciones realizadas por el grupo GOTA, disponibles en cualquier dispositivo móvil.

Palabras clave: WRF, modelos numéricos, portal meteorológico, aplicación móvil.

Abstract

Weather influences the human activities, affecting the complexity of their realization. Because of this, prediction of weather conditions is one of the challenges the man has always faced. However, due to its complexity, predicting the weather is unreliable.

In recent years, numerical weather prediction models have become increasingly precise and an indispensable tool to provide information of the state of the atmosphere. Their development and the increasing computing power of today's computers, have improved the effectiveness of short-term forecasts.

The research group GOTA (*Grupo de Observación de la Tierra y la Atmósfera*) of the University of La Laguna, using techniques for the study of atmospheric variables such as limited area numerical models, provides daily weather forecasts since 2012. These 48 hours predictions are made using the WRF regional model for the Canary Island region.

Currently, GOTA offers a "*Meteorological information website*", the result of another Final Degree Project. The aim of this portal is to provide information to GOTA staff and other researchers, but its functionality is not intended for "normal" users.

In this context, the objective of this project is to develop an application to select the relevant data from this website and to display it in a simple and user-friendly way. In this way, the application should show maps which can display predictions for several meteorological variables evolution of the main variables for a particular location. It is also intended that this platform will serve as a quick reference tool for the research team, as they will be able to show the predictions made by the group GOTA, available on any mobile device. Furthermore it should also display the temporal evolution.

Keywords: *WRF, numerical models, weather website, mobile application.*

Índice General

Capítulo 1. Introducción	5
1.1 Antecedentes	7
1.2 Cluster de GOTA.....	9
1.3 Demonio de la aplicación web.....	9
1.4 Aplicación web	10
Capítulo 2. Objetivos	11
Capítulo 3. Estado del arte de aplicaciones para dispositivos móviles	13
3.1 Dispositivo móvil.....	13
3.1.1 Características.....	13
3.1.2 Tipos de dispositivos móviles	13
3.2 Teléfono inteligente	14
3.3 Sistema operativo móvil	15
3.3.1 Modelo de Capas:.....	15
3.3.2 Sistemas operativos	15
3.4 Aplicación móvil.....	17
3.4.1 Desarrollo.....	17
3.4.2 Distribución	17
3.5 Evaluación de soluciones existentes:	19
3.5.1 El tiempo por eltiempo.es.....	20
3.5.2 El tiempo 14 días	21
3.5.3 El tiempo de AEMET	22
3.5.4 Conclusiones.....	22
3.6 Selección de la tecnología a utilizar:	23
3.6.1 Aplicaciones nativas (tabla 3.1.).....	23
3.6.2 Aplicaciones Web (tabla 3.2.)	24
3.6.3 Aplicación híbrida (tabla 3.3.).....	25

3.6.4 PhoneGap	26
3.6.5 JavaScript	27
3.6.6 HTML5	27
3.6.7 CSS3	28
3.6.8 jQuery	28
JQuery Mobile	28
Capítulo 4. Análisis de requisitos	29
4.1 Análisis de requisitos	29
4.1.1 Requisitos funcionales	29
4.1.2 Requisitos no funcionales	31
Capítulo 5. Fases y desarrollo del proyecto	32
5.1 Diseño	32
5.1.1 Obtención de datos del servidor	33
5.2 Implementación	34
5.2.1 Vistas	35
5.2.2 Controlador	41
Capítulo 6. Evaluación y resultados.	46
6.1 Resultados	47
Capítulo 7. Conclusiones y líneas futuras	54
7.1 Conclusiones	54
7.2 Líneas futuras	55
Capítulo 8. Summary and Conclusions	56
8.1 Summary and Conclusions	56
Capítulo 9. Presupuesto	58
9.1 Presupuesto	58
Bibliografía	59

Índice de figuras

Figura 1.1. Técnica de reducción de escala o downscaling dinámico.....	6
Figura 1.2. Portal web para información meteorológica.	8
Figura 1.3. Diseño del portal web.	8
Figura 3.1. Dispositivos móviles.....	14
Figura 3.2. Smartphone.....	14
Figura 3.3. Sistemas operativos.....	16
Figura 3.4. Tiendas virtuales.....	18
Figura 3.5. Google play.....	19
Figura 3.6. Búsqueda.	19
Figura 5.1. Modelo vista controlador.	32
Figura 5.2. Servidor puente.....	34
Figura 5.3. JQuery Mobile.	35
Figura 5.4. Descarga de JQuery Mobile.	36
Figura 5.5. Menú principal.....	37
Figura 5.6. Mapa selección.....	38
Figura 5.7. Mostrar mapa.	38
Figura 5.8. Mapa horizontal.....	39
Figura 5.9. a) Predicción. b) Ubicación. c) Seleccionar.	39
Figura 5.10. a) Hacer favorito. b) Favoritos.....	40
Figura 5.11. Predicción.	41
Figura 5.12. Directorios de Phonegap.	42
Figura 5.13. Carpeta “www”.....	43
Figura 5.14. Phonegap Build.....	44
Figura 6.1. Códigp QR.....	46
Figura 6.2. Pantalla de inicio.	47
Figura 6.3. Selección de mapas de predicción.....	48

Figura 6.4. Mapa de predicción.....	49
Figura 6.5. Mapa de predicción horizontal.	49
Figura 6.6. Menú evolución temporal.....	50
Figura 6.7. Geolocalización.	51
Figura 6.8. Favoritos.....	52
Figura 6.9. Evolución temporal.....	53

Índice de tablas

Tabla 3.1. Aplicaciones nativas.....	23
Tabla 3.2. Aplicaciones Web.....	24
Tabla 3.3. Aplicación híbrida.....	25
Tabla 9.1. Presupuesto.....	58

Capítulo 1.

Introducción

En esta sección se presenta el Trabajo de Fin de Grado y se describe de forma breve algunos conceptos sobre meteorología, necesarios para comprender cuáles han sido los motivos por los que se ha realizado el proyecto.

Por lo tanto en primer lugar se realizará un breve estudio sobre la meteorología. La meteorología es la ciencia que estudia el estado del tiempo, el medio atmosférico, los fenómenos producidos y las leyes que lo rigen. Se debe distinguir entre el tiempo atmosférico, que es la percepción de las variables atmosféricas en las condiciones actuales y su evolución, y el clima, que son las condiciones medidas durante un largo periodo de tiempo. A lo largo de la presente memoria vamos a centrarnos en las predicciones meteorológicas. [1]

Los datos proporcionados por los diferentes sistemas de observación (estaciones meteorológicas, radiosondeos, satélites, radares...) actuales se utilizan para alimentar a complejos modelos físico-matemáticos que, mediante modelos numéricos, predicen la evolución del sistema atmosférico. Para ello, se requiere la manipulación de grandes conjuntos de datos y la realización de cálculos avanzados, siendo esto posible gracias a la llegada de los superordenadores, que permiten la computación y simulación de estos modelos en tiempo real. Los modelos numéricos de predicción son utilizados para realizar previsiones en todo el mundo, tanto a escala global como regional. La ejecución de varios modelos permite una mayor precisión en la predicción además de ampliar el rango de días predecibles.

En este contexto, un modelo es un programa informático que resuelve las ecuaciones de dinámica atmosférica y que tiene en consideración los diferentes procesos que ocurren en el sistema Tierra-atmosfera de una forma más o menos aproximada. Un modelo puede ser global o regional, dependiendo de si abarca la totalidad de la Tierra o solo una determinada parte.

Las soluciones que obtienen estas previsiones son aproximadas, debido al gran número de hipótesis que se realizan para poder resolver estas ecuaciones.

La metodología que se sigue para la predicción meteorológica a gran resolución es la siguiente:

A partir de las observaciones de las diferentes variables y usando modelos globales (con resoluciones espaciales en torno a los 50-100 kms), se realizan las predicciones a escala global. Estas predicciones, que no representan correctamente los procesos que ocurren a escala local, son capaces de representar los fenómenos que ocurren a escala sinóptica (zonas de altas y bajas presiones, frentes, vientos a diferentes niveles, etc). Las salidas de estos modelos se utilizan posteriormente como condiciones iniciales y de contorno a los modelos de área limitada (modelos regionales) que se ejecutan a una resolución superior en una región particular del planeta (figura 1.1.). En este caso, al incrementar la resolución espacial, el modelo tiene en cuenta los procesos que ocurren a escala local por lo que es capaz de resolver con mayor precisión. Sin embargo, se requiere de una mayor potencia de cómputo para su ejecución. Este proceso se conoce como técnica de reducción de escala o downscaling dinámico.

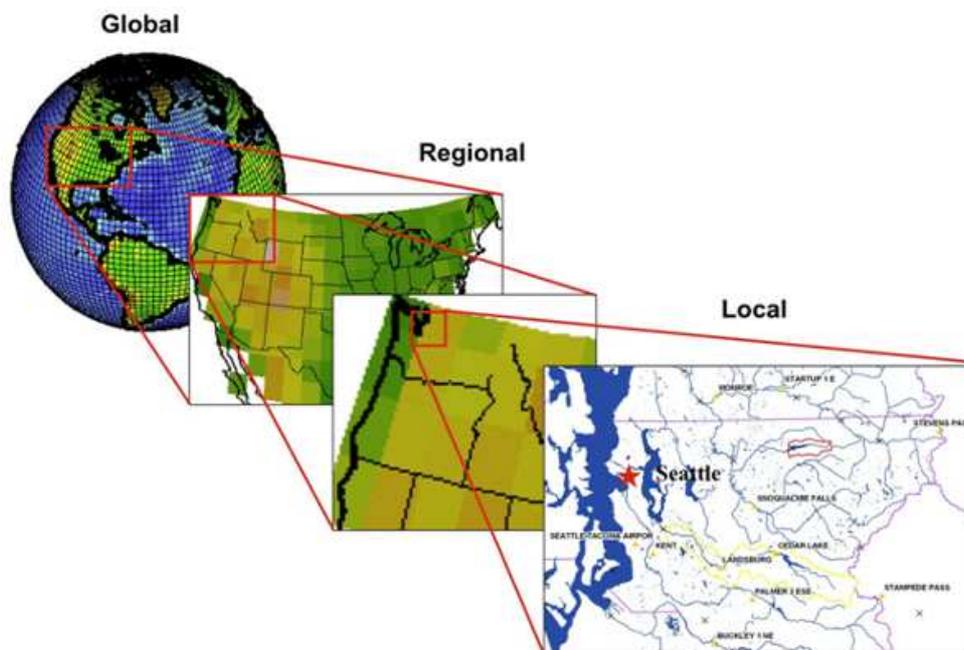


Figura 1.1. Técnica de reducción de escala o downscaling dinámico.

El modelo de investigación y previsión del tiempo WRF (Weather Research and Forecasting) es un sistema de predicción numérico del tiempo mesoscalar y de nueva generación, diseñado para las necesidades de investigación y predicción en el estudio de la atmósfera. Ofrece una amplia gama de aplicaciones meteorológicas, a través de escalas de decenas de metros a miles de kilómetros y puede generar simulaciones atmosféricas utilizando datos reales o condiciones idealizadas. [3]

Este modelo permite realizar trabajos tanto en líneas de investigación como en aplicaciones relacionadas tanto con la meteorología como con el clima, permitiendo evaluar diferentes configuraciones y realizar modificaciones que ajusten mejor el modelo a las características especiales de la región de interés, en nuestro caso el archipiélago canario.

El grupo GOTA viene realizando predicciones meteorológicas para las Islas Canarias utilizando el modelo WRF de forma operacional desde el año 2012. Para ello se realizan simulaciones usando 4 dominios que permiten obtener predicciones a 27, 9, 3 y 1 km para Canarias, Islas Occidentales y Tenerife respectivamente.

1.1 Antecedentes

Existe un amplio número de aplicaciones que muestran este tipo de información haciendo uso de datos obtenidos principalmente de modelos globales. Pero estos modelos no disponen de una resolución lo bastante precisa en una región con las características de las Islas Canarias. Para estos casos, es necesario describir los procesos que ocurren a escala “local”, teniendo que utilizar simulaciones con una gran resolución espacial.

Siguiendo esta línea, actualmente GOTA dispone de un portal web (figura 1.2), resultado de otro Proyecto Fin de Grado. Esta aplicación hace uso de las predicciones meteorológicas que realiza GOTA para Canarias, para mostrar estos resultados de forma interactiva, a los componentes de GOTA y a otros investigadores.

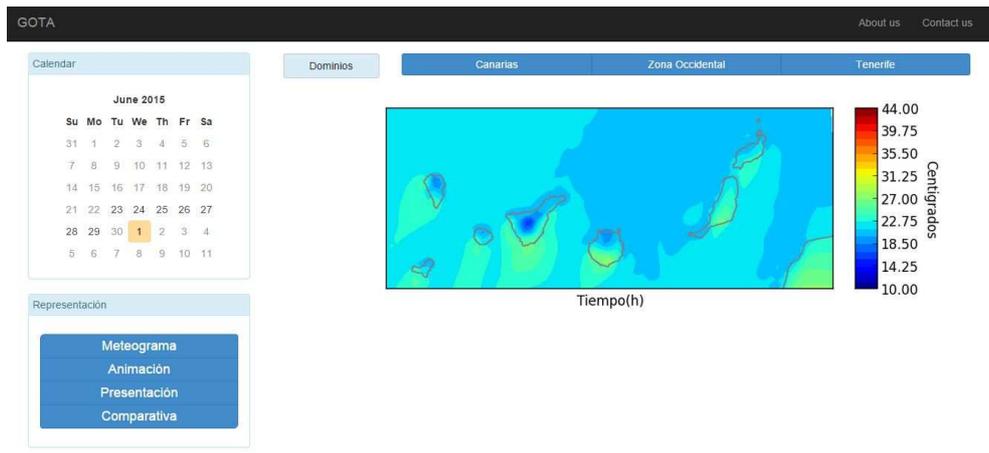


Figura 1.2. Portal web para información meteorológica.

El diseño del portal web se divide en tres partes (figura 1.3):

- La primera de ellas, el “cluster gota”, se encarga de obtener los datos necesarios para su representación y del preprocesamiento de los mismos.
- Por otra parte, la aplicación web consta de un demonio que recoge los datos proporcionados por el cluster. Además, genera las imágenes georreferenciadas para su posterior presentación.
- En último lugar, un servidor en Django se encarga de procesar los datos enviados por el demonio. Este servidor genera la representación de los datos que se muestra en la página web.

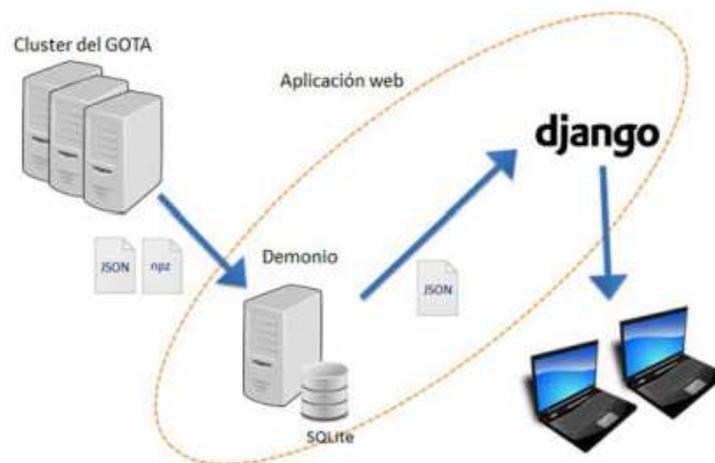


Figura 1.3. Diseño del portal web.

1.2 Cluster de GOTA

El Cluster Gota se encarga de generar las simulaciones de predicción meteorológica del grupo GOTA de la Universidad de La Laguna. Estas simulaciones son realizadas de forma rutinaria cada día y en ellas se generan las predicciones para las siguientes 48 horas en la región de las Islas Canarias.

El modelo numérico de predicción utilizado (WRF), genera una gran cantidad de información que da cuenta de las distintas variables atmosféricas, del terreno o de los componentes del modelo que son considerados en el sistema. Sin embargo, para este trabajo se consideraron únicamente las variables más utilizadas para describir la meteorología, como son la temperatura, las precipitaciones, los vientos y la cobertura nubosa.

Los datos de estas variables se almacenan en formato netcdf, pero debido al gran tamaño de estos ficheros, se extrae únicamente la información de interés a unos ficheros más manejables que se almacenan en formato npz.

La información recopilada permite representar la temperatura, lluvia y viento de una región durante las horas de predicción, o la visualización de la evolución temporal de la variable en cuestión para un punto determinado de la región.

1.3 Demonio de la aplicación web

El demonio de la aplicación web recoge los datos preprocesados en el Cluster y genera las imágenes con la representación visual de las diferentes variables para cada simulación. También almacena en la base de datos la información relevante de las simulaciones: variables, parámetros, dominios y la ruta donde se guardan las imágenes generadas. En él se crea un fichero JSON denominado ‘prediccion.json’ que agrupa los datos almacenados en la base de datos del demonio, permitiendo guardar y compartir de una forma rápida y sencilla los datos más relevantes, con la salvedad de que únicamente contiene los datos correspondientes a la última semana. Para ahorrar espacio de almacenamiento en el servidor.

Este fichero contiene la siguiente información:

- Fecha de realización de cada predicción.
- La hora de inicio.
- El número de dominios que se han utilizado.
- El número de parametrizaciones que se han usado y sus respectivos identificadores.
- El directorio donde se han guardado los .npz resultantes.
- Duración en horas de la predicción por cada zona.
- Un array “output_time_increments” que define el intervalo en minutos en los que se almacenan los datos en cada simulación.

1.4 Aplicación web

La aplicación web muestra de forma interactiva la información meteorológica en Canarias. Representa las variables meteorológicas más habituales para las distintas simulaciones realizadas, así como de los meteogramas. Además, permite la comparación de los resultados obtenidos en las predicciones de los días previos con los datos de observaciones registrados para el día anterior.

La web tiene implementada, entre otras, las siguientes funciones que son importantes para nuestra aplicación:

- `prediccion_prueba2.json`: Se encarga de devolver el fichero “prediccion.json”, con el contenido que antes hemos mencionado.
- `variables_request`: Se le pasa el dominio, latitud, longitud y la fecha. Éste devuelve una evolución temporal de la variable para una localización concreta.
- Ruta donde se almacenan las imágenes: Permite visualizar los mapas de predicción de las múltiples variables para un dominio y una fecha determinadas.

Capítulo 2.

Objetivos

En este trabajo se pretende extender la funcionalidad de esta web elaborando una aplicación que estará disponible en los diferentes tipos de dispositivos móviles. De esta manera la aplicación podrá mostrar a cualquier tipo de usuario los datos más relevantes de este portal. Permitiendo la visualización de mapas de predicción para las diferentes variables meteorológicas, además de poder realizar una consulta de la evolución temporal de las principales variables para una determinada localización seleccionada desde el propio dispositivo móvil.

Para llevar a cabo este objetivo principal, hay otros objetivos secundarios a conseguir:

- Evaluación de soluciones existentes.
Comprende una búsqueda de aplicaciones que proporcionen información meteorológica. En ellas se evaluará su usabilidad, el tipo de datos que presentan y cómo se presenta esta información al usuario.
- Selección de la tecnología a utilizar.
Se realizará una revisión del estado del arte en cuanto a tecnologías existentes para la realización del proyecto. Se evaluará la complejidad de las mismas, su flexibilidad (plataformas para las que se dispone) y las funcionalidades que ofrece.
- Análisis de requisitos.
En este apartado se realizará un análisis de las funcionalidades que debe contener la aplicación.
- Diseño.
Se realizará el diseño de la aplicación usando una estrategia modular que facilite su interpretación y flexibilidad
- Implementación.
Se debe prestar especial atención en la interconexión de los diferentes agentes involucrados: sistema de computación donde se realizan las simulaciones y servidor web.

Además, se desarrollarán las diferentes funcionalidades propuestas:

- Selección de dominio (Canarias, Islas Occidentales, Tenerife).
- Selección de variables:
 - Temperatura.
 - Precipitaciones.
 - Viento.
- Método de presentación:
 - Imágenes del dominio (con selección de la fecha a presentar).
 - Evolución temporal de la variable para un punto seleccionado en un mapa o almacenado como favoritos en el dispositivo.
- Pruebas.

Se llevarán a cabo las pruebas en las plataformas seleccionadas.

Capítulo 3.

Estado del arte de aplicaciones para dispositivos móviles

Para poder comprender el uso de las aplicaciones y su importancia, primero hay que hablar de los dispositivos móviles, sus características más importantes, su movilidad, su potencia de cómputo y conectividad. Entre los distintos tipos de dispositivo hablaremos del más extendido, el smartphone o teléfono inteligente, la propiedad de instalar programas o aplicaciones en ellos, cómo se distribuyen estas aplicaciones y cómo se desarrollan según el sistema operativo del teléfono.

3.1 Dispositivo móvil

Un dispositivo móvil es un tipo de computadora de tamaño pequeño con capacidad de procesamiento, conexión a Internet y memoria, diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales.

3.1.1 Características

La característica más importante de este dispositivo es el concepto de movilidad, son pequeños para poder portarse y ser fácilmente empleados durante su transporte. En muchas ocasiones pueden ser sincronizados con algún servidor externo para actualizar aplicaciones y datos.

3.1.2 Tipos de dispositivos móviles

Existe un variado tipo de dispositivos móviles: teléfono inteligente, tableta, videoconsola portátil, Netbook, PDA, reloj inteligente... En el desarrollo de

este proyecto nos centraremos en los teléfonos inteligentes aunque también es aplicable a tabletas. [4]



Figura 3.1. Dispositivos móviles.

3.2 Teléfono inteligente

El teléfono inteligente es un tipo de teléfono móvil con una mayor capacidad de almacenar datos y realizar actividades, y con una mejor conectividad que un teléfono móvil convencional.

Permiten al usuario instalar programas adicionales, hecho que dota a estos teléfonos de muchísimas aplicaciones en diferentes terrenos.

Entre otros rasgos comunes está la función multitarea, el acceso a internet vía Wi-Fi o internet móvil (4G, GPRS, GSM...), acelerómetros y GPS. [5]

En este proyecto se quiere desarrollar una aplicación que esté disponible en los dispositivos móviles más comunes, tales como son los Smartphone. Se precisará tanto de la conexión a internet para acceder a los datos del servidor de meteorología, como del acceso a GPS para geoposicionarnos.



Figura 3.2. Smartphone.

3.3 Sistema operativo móvil

Es el encargado de controlar el dispositivo móvil. Están orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

Algunos de los sistemas operativos más utilizados en los dispositivos móviles están basados en el modelo de capas.

3.3.1 Modelo de Capas:

- **Kernel:** O “núcleo”, proporciona el acceso a los distintos elementos del hardware del dispositivo. Ofrece a las capas superiores servicios como controladores o drivers para el hardware, la gestión de procesos, el sistema de archivos y el acceso y gestión de la memoria.
- **Middleware:** Es el conjunto de módulos que hacen posible la existencia de aplicaciones para móviles. Es transparente para el usuario y ofrece servicios como el motor de mensajería y comunicaciones, códecs multimedia...
- **Entorno de ejecución de aplicaciones:** Son un gestor de aplicaciones e interfaces programables abiertas por parte de los desarrolladores que facilitan la creación de software.
- **Interfaz de usuario:** Facilitan el diseño de la interacción con el usuario y la presentación visual de la aplicación. Incluye servicios como los componentes gráficos (botones, pantallas, listas, etc.) y el marco de interacción.
- Además de estas capas también existe una familia de aplicaciones nativas del teléfono que suelen incluir los menús, el marcador de números de teléfono etc...[6]

3.3.2 Sistemas operativos

Entre los sistemas operativos móviles más relevantes (figura 3.3.) y que ocupan una mayor cuota del mercado podemos encontrar los siguientes:

- **Android:** Líder del mercado móvil en sistemas operativos, está basado en Linux. Fue vendido a Google y modificado para ser utilizado en dispositivos móviles como los teléfonos inteligentes. Las aplicaciones para Android se escriben y desarrollan en Java con unas APIS propias. Una de las grandes cualidades o características de este sistema operativo es su carácter abierto.
- **iOS:** Creado por Apple originalmente para el iPhone. Es el sistema operativo que da vida a dispositivos como el iPhone, el iPad, el iPod Touch o el Apple TV. La simplicidad y optimización son las características principales que hacen que millones de usuarios se decanten por iOS en lugar de escoger otras plataformas que necesitan más software para mover el sistema con fluidez.
- **Windows Phone:** Sistema operativo móvil compacto desarrollado por Microsoft, diseñado para ser similar a las versiones de escritorio de Windows. Cuenta con un conjunto de aplicaciones básicas, además de una gran oferta de software de terceros disponible para Windows Mobile.



Figura 3.3. Sistemas operativos.

Después de haber hablado de los diferentes sistemas operativos, debemos destacar que uno de los objetivos del proyecto es el desarrollo de una aplicación multiplataforma disponible en los sistemas operativos más importantes, en este caso Android, iOS y Windows Phone.

3.4 Aplicación móvil

Una aplicación móvil es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Generalmente se encuentran disponibles a través de plataformas de distribución, facilitando su búsqueda e instalación al usuario, operadas por las compañías propietarias de los sistemas operativos como Android, iOS, BlackBerry OS y Windows Phone (figura 3.4.). En dichas tiendas virtuales existen aplicaciones móviles gratuitas y otras de pago.

3.4.1 Desarrollo

El desarrollo de aplicaciones para dispositivos móviles requiere tener en cuenta las limitaciones de estos dispositivos: el uso de batería, la existencia de una gran variedad de tamaños de pantalla y los diseños específicos de software y hardware de cada dispositivo.

3.4.2 Distribución

Cada plataforma o sistema operativo tiene su propio mercado de la empresa que lo desarrolla. También existen algunos mercados independientes multiplataforma como Amazon, desde estos mercados se puede buscar aplicaciones, descargarlas gratuitamente, comprarlas y actualizarlas automáticamente.

- **Google Play:** Es una plataforma de distribución de software en línea desarrollado por Google Inc. para dispositivos con sistema operativo Android. Fue lanzado en octubre de 2008. Hasta octubre de 2012, Google Play contaba con más de 700,000 aplicaciones.
- **App Store:** Fue el primer servicio de distribución de aplicaciones, siendo lanzada el 10 de julio de 2008. En 2012, el director ejecutivo de Apple, Tim Cook, anunció que existían más de 650,000 aplicaciones disponibles para dispositivos con iOS.
- **Windows Phone Store:** Es la plataforma de distribución de Microsoft para los dispositivos que cuentan con el sistema operativo móvil Windows Phone. Fue lanzado en octubre de 2010. Para octubre de 2012, contaba con más de 120,000 aplicaciones disponibles. En

mayo de 2013 Microsoft anunció que ya contaba con más de 145.000 aplicaciones en Windows Phone Store.

- **BlackBerry World:** Las aplicaciones para los dispositivos BlackBerry se encuentran disponibles mediante descarga a través de este servicio. Fue lanzada en 2009. En julio de 2011 se reportaron 3 millones de descargas al día sin problemas.
- **Amazon Appstore:** Es una aplicación móvil de distribución de software disponible para los dispositivos con sistema operativo Android. Fue lanzada en marzo de 2011, contando con 3,800 aplicaciones. [7]



Figura 3.4. Tiendas virtuales.

3.5 Evaluación de soluciones existentes:

Para la búsqueda y estudio de aplicaciones sobre información meteorológica vamos a utilizar el Play Store de Google, dado que el sistema operativo del dispositivo móvil que disponemos es Android (figura 3.5).

Una vez registrados con una cuenta de Gmail en el Play Store, podemos empezar a buscar y descargar aplicaciones, juegos, música...

El buscador es una aplicación que viene preinstalada en el sistema operativo de Android, además de aplicaciones nos permite hacer búsqueda de juegos, películas, música y libros.

En la primera página de la aplicación aparece el buscador en la parte superior, y a continuación aparecen recomendaciones, juegos nuevos, más populares, top ventas, recomendaciones, etc.



Figura 3.5. Google play.

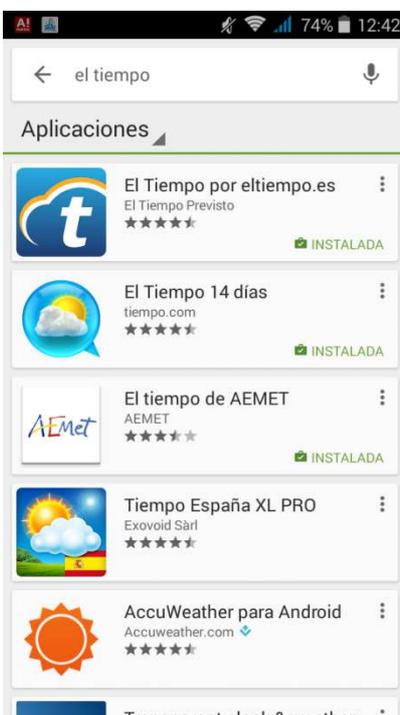


Figura 3.6. Búsqueda.

Realizaremos la búsqueda seleccionando “aplicaciones” e introduciendo “el tiempo” en el buscador. El resultado es una lista de aplicaciones, donde las más descargadas y mejor valoradas aparecen las primeras (Figura 3.6.)

Para el estudio utilizaremos tres aplicaciones en concreto, decantándonos por ellas por sus buenos comentarios, su número de descargas y las funcionalidades que aportan:

- El tiempo por el tiempo.es
- El tiempo 14 días
- El tiempo de AEMET

3.5.1 El tiempo por eltiempo.es

Eltiempo.es ofrece la previsión para los próximos 14 días en más de 200.000 localidades.

Funcionalidades:

- Página de inicio que ofrece una previsión horaria geolocalizada.
- Widget que muestra la información del tiempo en todo momento sin tener que entrar en la aplicación.
- Permite subir fotos relacionadas con el tiempo y unirte a su comunidad usuarios en la web.
- Postales del Eltiempo.es, que permite personalizar tus fotos con información meteorológica geolocalizada y compartirla en las redes sociales.
- Consulta del tiempo a través de la sección de Twitter en la propia aplicación.
- Permite escuchar el pronóstico del tiempo o activar sonidos asociados a la previsión.

Características:

- Previsión meteorológica detallada hora a hora para los próximos días.
- Previsión para los siguientes 14 días con información sobre lluvias, temperatura, sensación térmica, viento, nubes y presión atmosférica.
- Mapas de predicción de lluvias, temperaturas, presión, nubosidad para España, Europa, Reino Unido, Francia y Grecia, así como información detallada para Baleares y Canarias.
- Imágenes de radar y de satélite para todo el mundo.
- Información meteorológica para costas y mar con la predicción de olas, viento y temperatura del agua, así como mapas de las 12 principales áreas costeras de España.
- Avisos meteorológicos oficiales con alertas sobre oleaje, lluvias, etc. para todas las provincias e islas españolas.
- Información sobre estaciones de esquí en todo el mundo y partes de nieve para las estaciones de esquí de España.

- Información detallada de puntos de interés tales como aeropuertos, estadios de fútbol, colegios, plazas de toros, playas, campos de golf, etc.[8]

3.5.2 El tiempo 14 días

Permite consultar el tiempo a 14 días para más de 20.000 localidades de España y 200.000 de todo el mundo. Incluye información para las estaciones de Esquí.

Dispone de una predicción detallada por horas para hoy y mañana, cada 3 horas durante el resto de la semana, así como una predicción para la semana siguiente.

Ofrece múltiples variables meteorológicas:

- Símbolo de predicción
- Temperaturas máximas, mínimas diarias así como temperatura actual y sensación térmica hora a hora.
- Velocidad y dirección del viento incluyendo rachas máximas.
- Lluvia prevista.
- Nubosidad.
- Cota de nieve.
- Humedad relativa.

Además se ofrece información adicional de interés como la salida/puesta del sol y la fase lunar.

También se pueden consultar mapas de predicción con animaciones que ofrecen mapas de lluvia, temperatura, presión, nieve, etc. de España (península y Canarias) de Europa y otros países, además de imágenes actualizadas de varios satélites y el radar de lluvia de la AEMET.

Dispone de mapa de alertas de riesgos meteorológicos a nivel de España fácilmente interpretables según el nivel de alerta.

Cuenta con Widget para ver directamente en la pantalla del móvil la predicción de su ciudad hasta 5 días, con opción de widget de hora y predicción meteorológica. [9]

3.5.3 El tiempo de AEMET

Permite conocer información de manera precisa y detallada de donde lloverá, si los vientos van a ser fuertes o si se prevén fenómenos meteorológicos adversos, las últimas predicciones y avisos oficiales de la Agencia Estatal de Meteorología (AEMET).

Principales características:

- Predicciones y avisos del Servicio Meteorológico Nacional de España, AEMET.
- Predicciones a 7 días para más de 8.000 municipios, de hora en hora hasta el tercer día.
- Avisos de fenómenos adversos, día a día y hora a hora.
- Imágenes radar de la red de AEMET.

Además la aplicación identifica de manera automática el lugar donde está ubicado el dispositivo desde donde se consulta y muestra el tiempo previsto para esa localidad. También es posible obtener la predicción en cualquier otro lugar de España a través de un sencillo y rápido buscador. [10]

3.5.4 Conclusiones

Entre las características de estas aplicaciones, podemos diferenciar varias comunes a ellas: los mapas de predicción (muestran una evolución temporal de las variables) y la consulta de la predicción según geolocalización o selección. Estas son las funcionalidades que queremos implementar.

Además, nos hemos basado en la forma en que estas aplicaciones muestran los mapas para el desarrollo de nuestra aplicación.

3.6 Selección de la tecnología a utilizar:

Las diferentes opciones a la hora de desarrollar aplicaciones móviles multiplataforma son tres. A continuación se presentan las ventajas y desventajas de cada una, ya que esto supone un punto de inflexión a la hora de tomar la decisión de qué tipo de desarrollo se va a utilizar:

3.6.1 Aplicaciones nativas (tabla 3.1.)

Ventajas	Inconvenientes
Acceso completo al dispositivo.	Diferentes habilidades/ lenguajes/ herramientas para cada plataforma de destino.
Mejor experiencia del usuario.	Tienden a ser más caras de desarrollar.
Visibilidad en tiendas virtuales.	El código del cliente no es reutilizable entre las diferentes plataformas.
La actualización de la app es constante.	

Tabla 3.1. Aplicaciones nativas

Las aplicaciones nativas son las que se desarrollan de manera específica para un determinado sistema operativo utilizando un software de desarrollo específico, kit de desarrollo de software (SDK). Cada plataforma Android, IOS o Windows Phone, tiene su propio sistema, por lo que si quieres que la aplicación esté disponible en cada una de estas plataformas, deberás desarrollar una aplicación diferente para cada sistema operativo.

- Las aplicaciones para iOS se desarrollan con lenguaje Objective-C.
- Las aplicaciones para Android se desarrollan con lenguaje Java.
- Las aplicaciones en Windows Phone se desarrollan con lenguaje .Net.

La ventaja principal con respecto a los otros dos modelos, es que disponemos de la posibilidad de acceder a todas las características hardware del dispositivo: cámara, GPS y dispositivo de almacenamiento, entre otras. De esta manera se consigue que la experiencia del usuario sea mejor que en las demás.

Además, las aplicaciones nativas no requieren de conexión a internet para trabajar, pueden trabajar en local sin ésta. Una página web, necesitaría de un servidor al que consultar.

El proceso de marketing y promoción se agiliza debido a que la descarga e instalación se realiza a través de las tiendas de aplicaciones, disponibles en los sistemas operativos de los dispositivos, facilitando la búsqueda e instalación a usuarios.

Si el coste no es un problema y disponemos del suficiente tiempo, la mejor opción es la de desarrollar una aplicación nativa, pero el presupuesto y el tiempo de desarrollo es limitado, las aplicaciones web pueden proporcionar grandes ventajas.

3.6.2 Aplicaciones Web (tabla 3.2.)

Ventajas	Inconvenientes
El mismo código base es reutilizable en múltiples plataformas.	Requiere conexión a internet.
Proceso de desarrollo más sencillo y económico.	Acceso muy limitado a los elementos y características del hardware del dispositivo
No necesitan ninguna aprobación externa para publicarse(a diferencia de las nativas para estar visibles en app store).	La experiencia del usuario (navegación, interacción...) y el tiempo de respuesta es menor que en una app nativa.
El usuario siempre dispone de la última versión.	Requiere de mayor esfuerzo en promoción y visibilidad.
Pueden reutilizarse sitios “responsive” ya diseñados.	

Tabla 3.2. Aplicaciones Web

Una aplicación web suele desarrollarse en lenguajes como HTML, JavaScript y CSS, muy conocidos y utilizados en el diseño de páginas web. La principal ventaja es que podemos programar la aplicación en un único lenguaje independientemente del sistema operativo que vayamos a utilizar, permitiéndonos ejecutar una misma aplicación en diferentes dispositivos.

El contenido de éstas se adapta a la pantalla, adquiriendo el aspecto de la navegación de una aplicación nativa gracias a la utilización de plantillas

adaptativas. Esto se ejecuta en el navegador web del que dispone el dispositivo a través de una URL.

Estas aplicaciones no requieren instalación y, aunque también es una ventaja, genera un inconveniente al no estar visible en los mercados de aplicaciones. La promoción y comercialización se realiza de manera independiente.

Si nuestro objetivo es adaptar la web a formato móvil, las aplicaciones web son siempre una buena opción.

3.6.3 Aplicación híbrida (tabla 3.3.)

Ventajas	Inconvenientes
Es posible distribuirla en las tiendas de aplicaciones, como iOS y Android.	Experiencia del usuario más propia de la aplicación web que de la aplicación nativa.
Instalación nativa pero construida con JavaScript, HTML y CSS.	Diseño no siempre relacionado con el sistema operativo en el que se muestre.
Acceso a parte del hardware del dispositivo.	La experiencia del usuario (navegación, interacción...) y el tiempo de respuesta es menor que en una aplicación nativa.
El usuario siempre dispone de la última versión.	

Tabla 3.3. Aplicación híbrida.

Una aplicación híbrida es una combinación de las dos anteriores, se podría decir que recoge lo mejor de cada una de ellas. Se desarrollan con lenguajes propias del desarrollo web, HTML, JavaScript y CSS, permitiendo de ésta manera su uso en diferentes plataformas. Sin embargo, también presenta la posibilidad de acceder a las características hardware del dispositivo. La principal ventaja que aporta es que permite desarrollar un único código, compilado en los diferentes sistemas operativos y pudiendo así distribuir las aplicaciones en las respectivas tiendas. [11]

Tras realizar esta comparativa seleccionamos las aplicaciones híbridas para el desarrollo de este proyecto por las razones que se explican a continuación:

Desde un principio buscábamos una aplicación multiplataforma disponible para varios sistemas operativos. Dado que una aplicación nativa supone el

desarrollo de una aplicación en el lenguaje propio de cada sistema operativo, con los costes que esto supone, descartamos esta propuesta para evitar tener que diseñar, desarrollar y probar cada aplicación, ya que no contamos con el tiempo ni los medios necesarios para ello.

La aplicación web solucionaría los problemas presentados en la aplicación nativa, ya que permite que se pueda reutilizar el código y acceder a ella desde cualquier navegador disponible en los dispositivos móviles. No obstante, la limitación del acceso a los elementos hardware del dispositivo (tales como el GPS, necesario para la geolocalización), supone una razón de peso para rechazar esta propuesta.

La solución elegida, por tanto será de un diseño híbrido que comparta las características necesarias anteriores: diseño de una sola aplicación, en un único lenguaje, compilada para cada sistema operativo y con acceso al hardware.

Hay varios ejemplos de frameworks para el desarrollo multiplataforma de aplicaciones híbridas. Entre los más utilizados se encuentran PhoneGap y Cordova. Ambos parten del mismo núcleo, pero el primero da la posibilidad de compilar en la nube para un variado tipo de plataforma (tales como Android, IOS y Windows Phone) que, como se ha comentado anteriormente, son las que lideran el mercado. De esta forma, no sería necesaria la instalación de los drivers de cada una (necesarios para compilar la aplicación en local), el kit de desarrollo de software (necesario para poder ejecutar las pruebas como es el caso de Cordova) y un equipo específico (como en el caso de IOS, necesitaríamos un MAC para poder compilar, ejecutar y probar la aplicación en el sistema operativo de Apple).

3.6.4 PhoneGap

PhoneGap es un framework para el desarrollo de aplicaciones móviles desarrollado por Nitobi, y posteriormente comprado por Adobe Systems. Permite a los programadores desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3. El resultado no es una aplicación nativa al dispositivo, pero tampoco se trata de una aplicación web. En este caso hablamos de aplicaciones híbridas.

PhoneGap maneja la API que permite tener acceso a elementos como el acelerómetro, la cámara, los contactos en el dispositivo, la red, el almacenamiento, las notificaciones, etc. Estas API se conectan al sistema operativo usando el código nativo del sistema huésped a través de una Interfaz de funciones en Javascript.

PhoneGap no necesita de un simulador dedicado, permitiendo su desarrollo ejecutando las aplicaciones en nuestro navegador web y nos da la posibilidad de utilizar frameworks como Sencha Touch o JQuery Mobile.

PhoneGap es una distribución de Apache Cordova. La diferencia principal entre Apache Cordova y Phonegap es que el segundo tiene acceso a servicios de compilación en la nube proporcionados por Adobe Creative Cloud.

Apache Cordova es un software de código abierto y, tanto éste como PhoneGap, pueden ser utilizados libremente en cualquier aplicación sin necesidad de atribución o licencias de ningún tipo. [12]

3.6.5 JavaScript

JavaScript es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML. [13]

3.6.6 HTML5

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. [14]

HTML, siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), es un estándar a cargo de la W3C, que sirve para la elaboración de páginas web. Define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, videos, entre otros. [15]

3.6.7 CSS3

Hoja de estilo en cascada o CSS es un lenguaje formulado por el World Wide Web Consortium (W3C), utilizado para definir y crear la presentación de un documento estructurado escrito en HTML. CSS3 es la última versión y más actual para CSS. [16]

3.6.8 jQuery

jQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

jQuery es software libre y de código abierto. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código. [17]

JQuery Mobile

Framework optimizado para dispositivos táctiles que está siendo desarrollado actualmente por el equipo de proyectos de jQuery. El desarrollo se centra en la creación de un Framework compatible con la gran variedad de smartphones y tablets. Es compatible con otros frameworks móviles y plataformas como PhoneGap. [18] Entre sus características, las más importantes son: desarrollo rápido de interfaces, sintaxis sencilla, efectos prediseñados, manejo de eventos y una baja curva de aprendizaje que facilitan un desarrollo fácil y ágil del diseño de la aplicación. [19]

Capítulo 4.

Análisis de requisitos

Hay que tener muy presente a la hora de realizar un trabajo final de grado cuál es la finalidad de éste. Este análisis genera los requisitos del sistema, que son el comienzo de todo proyecto.

4.1 Análisis de requisitos

Los requisitos de la aplicación son el punto de referencia que nos permitirá alcanzar, de la forma más satisfactoria posible, el resultado deseado. Esto juega un papel muy importante en el resto del trabajo.

Se ha de mencionar que, durante el desarrollo de este trabajo, se ha gozado de una gran libertad. Algunos requerimientos han ido surgiendo durante las fases de desarrollo del trabajo y se han visto añadidos, modificados o adaptados en el proceso.

4.1.1 Requisitos funcionales

Los requisitos funcionales definen una función del software o de sus componentes que el sistema debe cumplir, estableciendo el comportamiento del sistema.

Como requisitos se plantean los siguientes: la aplicación debe poder ser utilizada desde cualquier dispositivo móvil y mostrar los resultados de las simulaciones, permitiendo al usuario visualizar los mapas de predicción para las múltiples variables meteorológicas.

Además, esta aplicación será capaz de mostrar al usuario cómo va a ser la evolución temporal de las principales variables para una localización concreta dentro de la región de interés.

A continuación se listan los requisitos funcionales:

- La aplicación debe estar disponible para los distintos tipos de dispositivos móviles.
- En el menú principal se puede seleccionar:
 - Mostrar los mapas de predicción para los distintos dominios de Canarias, simuladas por el GOTA.
 - Consultar la evolución temporal de las variables para un punto determinado (en cualquiera de las zonas de las simulaciones anteriores).
- En la galería de mapas habrá tres opciones a seleccionar:
 - Fecha: Para las cuales se han generado los mapas y están disponibles para visualizar.
 - Región: Canarias, Canarias Occidental y Tenerife.
 - Variable: Lluvia, temperatura y viento.
- Los mapas en la galería se muestran a pantalla completa y se podrán seleccionar en un rango desde la hora actual hasta las 18:00 horas del día siguiente.
- Podremos consultar la evolución temporal diaria de las variables a través de:
 - Nuestra localización actual.
 - Seleccionado la localización en un mapa.
- Visualización de la evolución temporal diaria del tiempo, para la ubicación seleccionada, desde la hora actual hasta las 18:00 horas del día posterior. Se deben mostrar las siguientes variables:
 - Temperatura.
 - Precipitación.
 - Viento: Intensidad y dirección.
 - Nubosidad.

4.1.2 Requisitos no funcionales

Los requisitos no funcionales, en cambio, se enfocan al diseño y la implementación. Explican cómo opera el sistema en lugar de juzgar las funciones a realizar, que corresponden a los requisitos funcionales.

En ellos se especifican: funciones a implementar, Phonegap como la tecnología que vamos a utilizar, los lenguajes en los que vamos a desarrollar, (HTML, CSS y JavaScript) y cómo recoger la información (a través de AJAX estructurada en JSON).

A continuación se listan los requisitos no funcionales:

- La aplicación se implementa con Phonegap, un framework multiplataforma.
- Los menús y las vistas de la aplicación se desarrollan en HTML y CSS, agilizando este proceso con los estilos predefinidos que proporciona jQuery Mobile.
- La información en el apartado de la galería se extrae de un fichero JSON que se encuentra en el servidor de GOTA y que recuperamos utilizando AJAX. La fecha de la predicción se muestra ordenada, primero las más recientes, y por defecto se selecciona la última predicción.
- Utilizaremos JavaScript para conocer el tamaño de pantalla, redimensionándola al girarla. Para seleccionar las distintas imágenes haremos uso de un slider y jQuery, obteniendo su valor cada vez que cambie y modificando la URI de cada imagen a mostrar.
- La API de Google Maps nos proporciona nuestra localización actual además de los mapas de donde podremos obtener las coordenadas de la posición seleccionada.
- Haremos una consulta mediante AJAX al servidor de GOTA, pasando la latitud y la longitud. Éste nos devuelve una respuesta JSON con los datos necesarios a mostrar.
- Los datos de la predicción se muestran con un decimal. Las variables se muestran con iconos representativos: el viento varía con su dirección y la nubosidad en función del estado.

Capítulo 5.

Fases y desarrollo del proyecto

Las fases de desarrollo de este proyecto se dividen en dos. La fase de diseño, donde se toman decisiones sobre la arquitectura de la que se compone la aplicación, y la segunda fase, donde se ejecuta la implementación de ésta.

5.1 Diseño

El diseño de la aplicación parte de la filosofía de desarrollo modelo-vista-controlador (figura 5.1), muy utilizado en la implementación de aplicaciones web cliente-servidor. Se desarrollan multitud de frameworks en los principales lenguajes de programación que implementan este patrón:

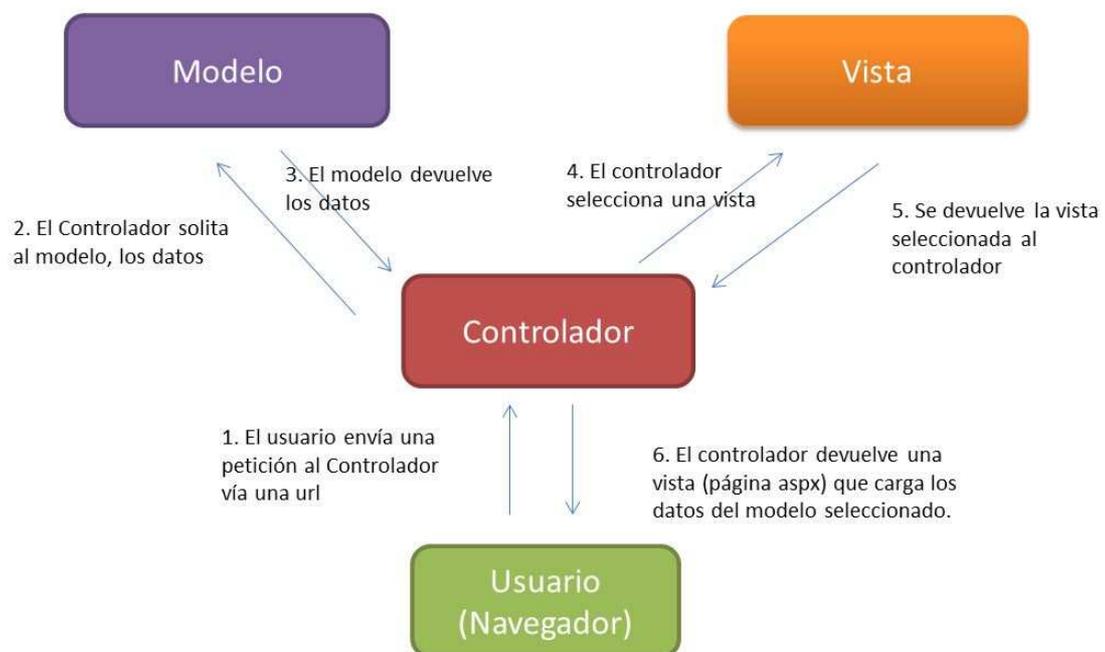


Figura 5.1. Modelo vista controlador.

Tal y como se muestra en la figura 5.1, el modelo-vista-controlador se divide en las siguientes partes:

- El Modelo: Es la representación de la información. Gestiona los accesos a la información, como consultas o actualizaciones y los privilegios de acceso a ésta. El “controlador” manda las peticiones de acceso al “modelo” y éste envía la información a la “vista”. El modelo en la aplicación lo proporciona Phonegap y su API en JavaScript aunque, en nuestro caso, no es importante ya que solo almacenamos información relevante a la sesión. La información que vamos a mostrar la recibimos de la aplicación web, que tiene su propio modelo y gestión de la información.
- El Controlador: Hace de intermediario entre la “vista” y el “modelo”. Invoca las solicitudes de información al modelo, que pueden ser accesos a la base de datos local o, en nuestro caso, hacer las peticiones al servidor. Éste responde a eventos (normalmente realizados por el usuario o por el propio controlador) y genera una interfaz de usuario con los datos recibidos del modelo y la vista seleccionada. El lenguaje que utiliza es JavaScript, dado que estamos hablando de programación web en el lado del cliente.
- La Vista: Recibe los datos del controlador y presenta el modelo en un formato adecuado con el que pueda interactuar e interpretar el usuario. Phonegap utiliza para las vistas HTML y CSS. Nos hemos ayudado de otro framework, JQuery Mobile, compatible con una gran variedad de smartphones y tablets.[20]

5.1.1 Obtención de datos del servidor

Los datos e imágenes para las predicciones están almacenados en un servidor externo denominado “portal web para información meteorológica”, otro Trabajo Fin de Grado realizado por una compañera del Grado de Ingeniería Informática de la Universidad de La Laguna, Ángela Hernández Delgado.

La dirección IP del servidor es “10.209.2.98” de acceso interno desde la red de la ULL. Desde el exterior se puede acceder a ella a través de una red privada virtual o VPN.

Esto supone que, para acceder a los datos, tenemos que estar conectados a la red de la ULL o estar conectados a ésta a través de una VPN. Esto limita el uso y el acceso de la aplicación a usuarios registrados en la Universidad de La Laguna, dado que es necesario poseer un nombre de usuario y una clave para acceder a ésta.

La solución consta de utilizar un servidor a modo de “puente” (figura 5.2.), como se puede ver en la siguiente figura:



Figura 5.2. Servidor puente.

La aplicación hace peticiones al servidor puente y éste, a su vez, al servidor interno de la universidad. De esta manera se devuelven los datos a la aplicación resolviendo el problema de la limitación de acceso.

Este servidor se aloja en el Centro de Cálculo de la Escuela Técnica Superior de Ingeniería y Tecnología. Entre sus servicios, ofrece a sus alumnos el alojamiento de pequeñas páginas webs escritas en PHP. Se accede a este espacio a través de la dirección FTP, “banot.etsii.ull.es”, con el nombre de usuario y contraseña.

5.2 Implementación

Durante la implementación del proyecto es importante llevar a cabo un control constante de los requisitos. Este control ha sido llevado a cabo

durante las reuniones, comprobando en qué estado se encontraban, cuáles se estaban cumpliendo o están pendientes.

Se han tomado decisiones en cuanto a la actividad a realizar, qué requisitos eran los más importantes, qué dificultad tenían y sus posibles soluciones.

En el apartado de implementación se comentará las herramientas utilizadas, las decisiones que se han tomado y cómo se han resuelto los problemas encontrados durante el desarrollo de la aplicación.

5.2.1 Vistas

La vista es una parte importante de la aplicación ya que es la manera en la que el usuario recibe la información. Es por ello por lo que ésta debe ser intuitiva y de fácil comprensión. Es vital que se adapte al mayor número de dispositivos, dado que existe una gran variedad y lo que se pretende es abarcar el mayor número de ellos, para llegar así al mayor número de usuarios posibles.

JQuery Mobile (figura 5.3.) es un framework compatible con plataformas móviles como Phonegap. Permite el desarrollo de interfaces optimizadas, compatible con un gran número de dispositivos móviles, facilitándonos este trabajo. [21]



Figura 5.3. JQuery Mobile.

Para su utilización hay que descargar las librerías JavaScript de jQuery y jQuery Mobile, así como sus hojas de estilo y guardarlas junto a la aplicación. Las librerías también se pueden enlazar directamente desde la web a través de los enlaces que proporciona JQuery pero, no es recomendable para trabajar con Phonegap dado que el dispositivo no siempre tendrá conexión a internet y no podríamos ejecutar la aplicación. [22]

jQuery Mobile dispone de un potente Framework que permite a los desarrolladores personalizar sus temas. La aplicación ThemeRoller (disponible vía web) permite, de manera sencilla y con drag and drop, la personalización y descarga de un CSS personalizado, pudiendo emplearlo directamente en el proyecto. [23]



Figura 5.4. Descarga de JQuery Mobile.

Una de las características de jQuery es la de no pasar de un documento HTML a otro, como haría un sitio web. Las páginas en las que se desplaza la aplicación se manejan en un solo documento HTML. En nuestro caso, el archivo se va a llamar “index.html”, nombre necesario para que pueda trabajar con Phonegap.

jQuery Mobile trabaja las páginas en un solo archivo de esta manera se acelera el tiempo de carga, dando imagen de una aplicación más fluida. Funciona mostrando la parte del documento que corresponde a la pantalla actual, mientras que el resto permanece oculto. De esta forma permite eventos y animaciones al cambiar de una pantalla a otra.

Los atributos personalizados de jQuery Mobile se definen con el prefijo “data-“. “Data-role” es un ejemplo de los atributos más utilizados, permitiendo crear, entre otros elementos, el tipo página.

El documento puede contener más de un elemento de tipo “page”, que representa el contenido de una pantalla completa. A su vez, una “page” puede tener otros divs, definidos por un data role “header”, content o “footer”.

Entre los atributos más comunes de jQuery Mobile podemos encontrar los siguientes:

- data-role: Especifica el rol de cada elemento, como “header”, “content”, “footer”, etc.
- data-theme: Especifica qué tema se va a utilizar con los elementos dentro del contenedor.
- data-transition: Especifica una de las posibles animaciones que se pueden emplear al pasar entre páginas.
- data-icon: Especifica uno de los posibles iconos que pueden ser añadidos a un elemento.

La aplicación cuenta con siete vistas. Las cuatro primeras se componen de un menú de selección, la quinta de un mapa donde poder seleccionar la localización y las dos últimas muestran la información.

Menú principal (figura 5.5)

El menú principal consta de los siguientes contenidos:



- La cabecera: Es un div, con el atributo data-role "header". Esto lo posiciona en la cabecera y contiene el nombre de la aplicación, “GOTA”.

- El cuerpo: Se define con el atributo data-role "content". Contiene una lista con los enlaces a otras dos páginas, “Mapas” y “Predicción”. Los enlaces tienen el atributo “href” típico de HTML, con la diferencia de que para enlazar una página interna con jQuery Mobile, hay que añadir “#” al nombre de ésta. Al atributo data-transition de los enlaces se le asigna el tipo de transición "slide", que genera un efecto de deslizamiento al cambiar de página.

Figura 5.5. Menú principal.

- El pie: Es un div, data-role "footer", que se posiciona en el pie de la página. Tiene una barra de navegación con tres accesos directos a diferentes páginas de la aplicación: “Mapa”, “Home” y “Tiempo”.

Mapas (figura 5.6.)

La página de mapas contiene un menú donde se puede seleccionar: la fecha de la predicción, la región y la variable que queremos mostrar.

El estilo de la cabecera y el pie de la página se mantienen iguales durante toda la aplicación. A diferencia que en la cabecera aparece un botón del tipo data-role "button" y el atributo data-rel "back", que nos permite regresar a la página anterior.

El contenido del cuerpo es un formulario, donde podemos seleccionar la fecha, región y variable que queremos mostrar. En la parte superior, aparece un enlace que nos lleva a la página donde se muestra el mapa.

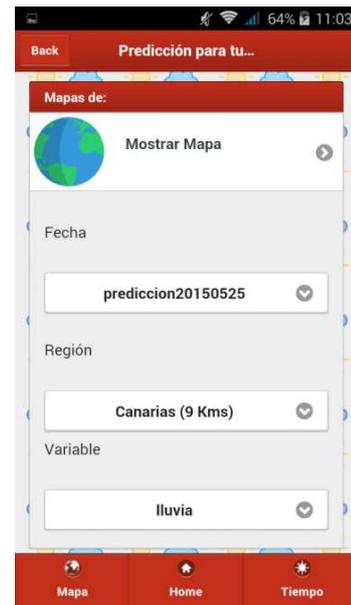
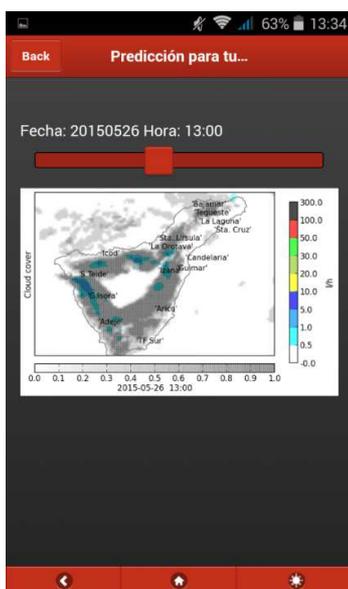


Figura 5.6. Mapa selección.



Mostrar mapa

En esta página se muestra el mapa correspondiente con las opciones seleccionadas en la página anterior.

El cuerpo lo componen dos elementos: un elemento de selección (donde podemos elegir la hora que queremos visualizar) y otro donde se muestra la imagen del mapa. (Figura 5.7)

Para ver la imagen a pantalla completa tendremos que girar el dispositivo. (Figura 5.8).

Figura 5.7. Mostrar mapa.

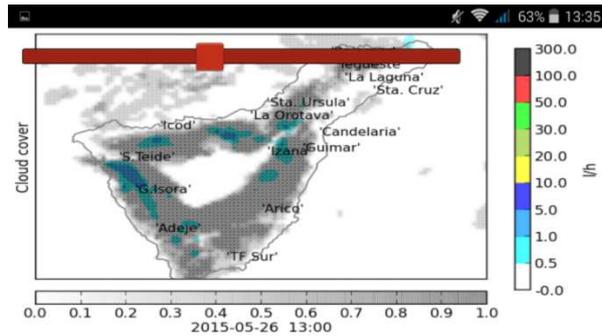


Figura 5.8. Mapa horizontal.

Método de consulta para la predicción

En el cuerpo de esta página los dos primeros enlaces que nos dirigen a los métodos de consulta posible para la predicción (Figura 5.9 a).

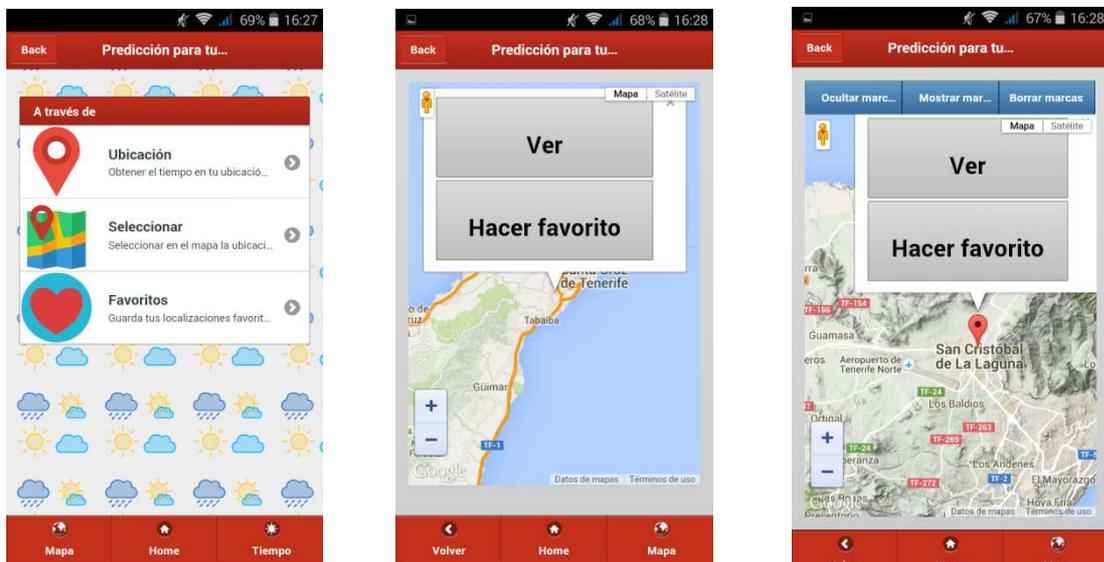


Figura 5.9. a) Predicción. b) Ubicación. c) Seleccionar.

El primer método calcula nuestra posición gracias al GPS y la muestra en el mapa, pudiendo seleccionarla para ver la información (Figura 5.9 b).

En el segundo, seleccionamos en el mapa la localización que queremos consultar, pudiendo ocultar, mostrar y borrar las últimas marcas seleccionadas (Figura 5.9 c).

En el tercer enlace (5.9 a) se añade la funcionalidad extra “Favoritos” que no fue planteada inicialmente en la realización de este trabajo. Además de poder seleccionar una localización, también podemos guardar hasta tres

favoritas, consiguiendo de esta manera acceder a las localizaciones que utilizamos más frecuentemente, ahorrando el trabajo de volver a buscarlas.

Al hacer selección “Hacer favorito” (5.9 b, 5.9 c), se abre una ventana con una entrada de texto donde el usuario puede añadir un nombre a la localización (Figura 5.10 a). Estas localizaciones se pueden consultar en la vista de favoritos (Figura 5.10 b), que muestra las localizaciones que previamente hemos guardado, donde haciendo selección nos lleva a la vista para la predicción.

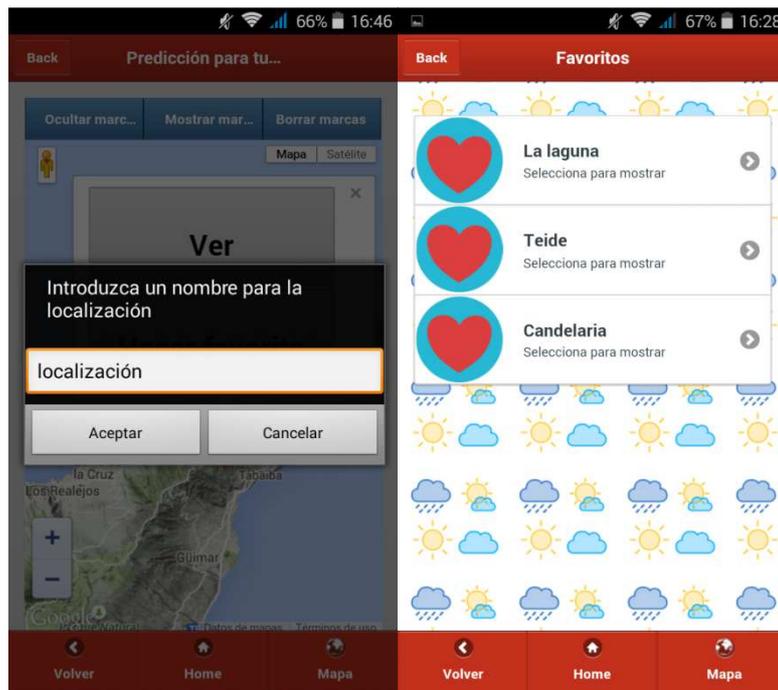
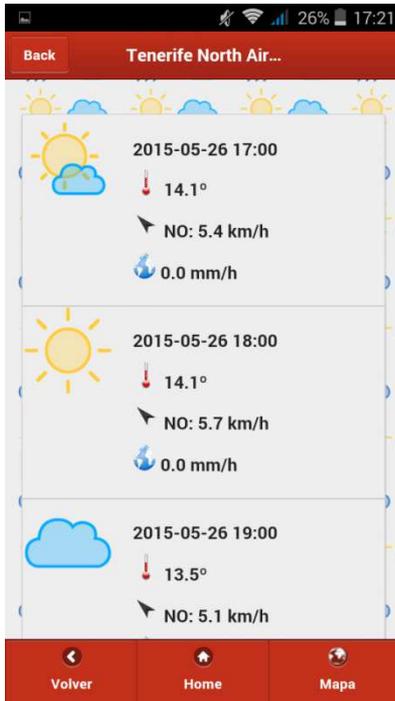


Figura 5.10. a) Hacer favorito. b) Favoritos.

Predicción del tiempo (figura 5.11.)

- La cabecera muestra el nombre de la localización para la predicción que acabamos de realizar.
- En el cuerpo se muestra una lista de las horas, partiendo de la hora actual, hasta las 18:00 horas del día siguiente.
 - Las variables que se muestran en cada elemento son:
 - Temperatura: En grados centígrados, representada por el icono de termómetro.



o Viento: El icono en forma de diagrama de barbas muestra su dirección y su velocidad se representa en km/h.

o Precipitación: En mm/h, representada por el icono de una gota de lluvia.

o Nubosidad: La nubosidad se representa con el icono de mayor tamaño (que se encuentra arriba a la izquierda) y puede representar tres posibles valores (despejado, parcialmente nublado y nublado).

Figura 5.11. Predicción.

Iconos e imágenes de la aplicación

JQuery Mobile dispone de un variado grupo de iconos de uso normal en una aplicación, pero si buscas un icono más específico hay que añadirlo manualmente.

Para las imágenes e iconos nos hemos ayudado de paquetes de iconos gratuitos que hemos descargado de internet, en especial estos tres:

- Iconfinder: Es una web con un buscador donde se pueden descargar iconos de manera gratuita o pagando. [24]
- WIND-PACK: De este paquete hemos sacado las flechas necesarias para marcar la dirección del viento. [25]
- Vector-stylish-weather-icons: Las imágenes que representan la variable de nubosidad han sido sacadas de este paquete. [26]

5.2.2 Controlador

El controlador es el encargado de responder a las acciones que realiza el usuario, de forma que sean transparentes a éste. El framework de Phonegap maneja el controlador de la aplicación y se encarga del renderizado de la aplicación, mediante vistas web y su API, que permiten el acceso a elementos hardware del dispositivo.

Para poder desarrollar en Phonegap, existen dos vías: desarrollar en local, (con la interfaz de línea de comandos, compilando y ejecutando en nuestro ordenador) o trabajar directamente en la nube con PhoneGap Build.

Para poder desarrollar en local hay que seguir los siguientes pasos:

- Primero instalaremos Phonegap vía node.js.
- Con los comandos de Phonegap crearemos un proyecto.
- Luego se añaden las plataformas en las que vamos a trabajar (iOS, Android, Windows phone)
- Es necesario tener instalado el SDK de cada plataforma, pero esto no es siempre posible, iOS solo funciona en MAC y Windows Phone en Windows.
- Teniendo el SDK instalado ahora podremos construir la aplicación.

Esto genera el directorio que vemos a continuación (figura 5.12):



Nombre	Fecha de modifica...
.cordova	05/02/2015 17:54
hooks	05/02/2015 17:54
platforms	05/02/2015 17:54
plugins	05/02/2015 17:54
www	05/02/2015 19:32
config.xml	05/02/2015 17:54

Figura 5.12. Directorios de Phonegap.

- En la carpeta “.cordova”, se encuentra el núcleo del framework de Phonegap.
- “Platforms” contiene las diferentes plataformas en las que vamos a trabajar.
- Los diferentes plugins se instalan en su carpeta correspondiente.
- En “www” está contenida nuestra aplicación, con los archivos y carpetas necesarias para implementar una aplicación web.
- El fichero “config.xml” contiene los ajustes para la configuración de Phonegap.

Aquí tenemos un ejemplo de la carpeta “www” (figura 5.13.). Éste es el ejemplo que proporciona Phonegap al crear la primera aplicación.

Nombre	Fecha de modifica.
css	05/02/2015 17:54
img	05/02/2015 19:32
js	05/02/2015 17:54
res	05/02/2015 17:54
spec	05/02/2015 17:54
icon.png	05/02/2015 17:54
index.html	05/02/2015 17:54
spec.html	05/02/2015 17:54

Figura 5.13. Carpeta “www”.

El contenido de ésta es el de los ficheros necesarios de cualquier página web, con la única diferencia de que el archivo que carga Phonegap es el “index.html”. [27]

La desventaja de trabajar de esta manera es tener que realizar todas las configuraciones (para cada plataforma) para hacer uso de Phonegap. Para ello sería necesario obtener los SDKs de cada una de ellas y sus entornos de desarrollo. Esto puede llegar a ser muy complejo, debido a las configuraciones que lleva cada una, implicando un gran coste de tiempo y de memoria por el tamaño de los paquetes que tenemos que descargar. Otro inconveniente que nos surge es el sistema operativo que usamos para desarrollar. Esto es debido a que para el entorno de desarrollo de Windows Phone necesitamos del sistema operativo de Windows y para IOS, nos hará falta el sistema operativo de Apple.

La segunda opción es utilizar Phonegap Build (5.14.). Entre sus ventajas están: la posibilidad de trabajar en la nube y la posibilidad de desarrollo para un amplio número de plataformas (Android, iOS, Windows Phone), ahorrando de esta manera el tiempo que dedicamos a la instalación y configuración de cada una de estas.

Para trabajar con esta aplicación web sólo es necesario subir en un fichero zip los archivos web (HTML, CSS y Javascript) que corresponden a la carpeta “www” antes mencionada. Phonegap Build se encarga del empaquetado y compilación de la aplicación para cada uno de los sistemas operativos, posibilitando la descarga directa en el dispositivo móvil a través de un código QR o una URL. [28]

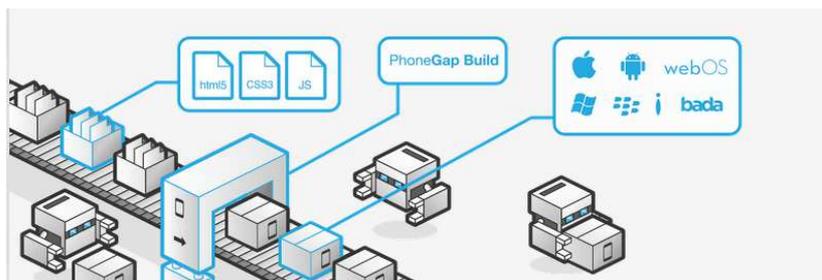


Figura 5.14. Phonegap Build.

Phonegap Build nos permite crear un número ilimitado de aplicaciones de manera totalmente gratuita, siempre que estén vinculadas a un repositorio público en GitHub. También nos permite crear de manera gratuita una aplicación privada. Si queremos desarrollar más aplicaciones privadas, tendríamos que consultar el coste en su página web. [29]

Los datos que se encuentran alojados en el portal web de meteorología se cargan utilizando la función “getJSON()”. [30]

Esta función, que pertenece a la librería de JQuery, carga datos en JSON mediante una petición HTTP GET. El objeto devuelto es una matriz de JavaScript con la estructura que define JSON, permitiendo trabajar de una manera más sencilla.

JSON es un formato ligero en el intercambio de datos. Su simplicidad ha generalizado su uso entre la comunidad de desarrolladores, estando disponible en cualquier navegador web. [31]

Las funciones encargadas de controlar la aplicación se encuentran en la carpeta “js” del proyecto. Entre ellas, las más importantes son: la encargada de mostrar la evolución temporal, los mapas de la predicción y la que obtiene la localización.

“dia(Latitud,Longitud)” muestra la evolución temporal para una localización dada. A la función se le pasan dos variables (latitud,longitud) y mediante una petición a la función “getJSON()” (que devuelve el nombre del lugar, un listado de horas y las variables), se muestra la lista a partir de la hora actual. Las variables se representan con un icono que ayuda al usuario a su identificación. Estos son fijos, a excepción del viento y la nubosidad.

“mapas()” carga del servidor el fichero “predicción.json”, que contiene información de la fecha de las simulaciones ejecutadas, de las regiones

comprendidas y de las variables estudiadas. Esta información es necesaria para la carga del formulario de selección, debido a que las imágenes que queremos mostrar corresponden a una fecha de simulación, región y variable. Una vez seleccionadas estas variables por el usuario, se cargan las imágenes a través de una URI del tipo de “lugar donde se almacena + fecha de la simulación + zona + variable + año + mes + día + hora”, y mediante un slider y una función en JavaScript se cambia la hora, refrescando la imagen mostrada. (Figura 5.6.)

Para la localización hay dos funciones que usan la API en JavaScript de google maps que provee de todo lo necesario para trabajar con mapas: clases, métodos y vistas. La primera nos devuelve nuestra geolocalización y la segunda nos pinta un mapa donde la podemos seleccionar. Esta información es la que usaremos luego para mostrar la evolución temporal en una localización dada. [32] (Figuras 5.9. b, c)

Por último para poder almacenar los datos necesarios para guardar nuestras localizaciones favoritas Phonegap, permite el uso de un almacenamiento local a través de la API de LocalStorage definida en HTML5.

LocalStorage tiene las siguientes características:

- Es un almacenamiento de tipo clave/valor.
- La información queda guardada en el dispositivo y sobrevive a reinicios de la aplicación.
- Está limitado a unos 5MB.

El inconveniente que presenta LocalStorage es que las claves y valores han de ser tipo String pero convirtiendo estas cadenas a JSON podemos almacenar estructuras más complejas.

Para el almacenamiento de favoritos utilizaremos dos funciones:

- “favoritos()” accede al LocalStorage y muestra las localizaciones ya almacenadas en una lista.
- “favoritos_add” (latitud,longitud) esta función se llama cuando almacenamos una localización favorita. Se le pasa la latitud y la longitud de la localización seleccionada y se guarda en el LocalStorage con el nombre que se solicita al usuario.

Capítulo 6. Evaluación y resultados.

La forma de evaluar la aplicación ha sido mediante el uso de herramientas de desarrollo web, de las que disponen los exploradores. Éstas permiten comprobar los elementos de una página web, ver su código fuente, seleccionar elementos y comprobar el correcto funcionamiento del código en JavaScript.

Se ha llevado un control de versiones de la aplicación, comprobando y añadiendo las funcionalidades necesarias. Los archivos se encuentran disponibles en el siguiente enlace de GitHub “<https://github.com/alu0100611298/GOTA>”, una plataforma que usa un control de versiones Git.

También se ha creado una página web donde el tutor y alumno han podido comprobar el desarrollo y los avances de la aplicación de manera inmediata. Primero se realizan las pruebas en la web, comprobando el correcto funcionamiento con las herramientas de desarrollo, y luego se prueba en el dispositivo móvil, descargando e instalando la aplicación. La web está alojada en el espacio que el Centro de Cálculo dispone para esto, a la que se accede mediante este enlace “<http://banot.etsii.ull.es/alu4213/gota/index.html>”.

La aplicación ha sido probada en varios dispositivos con sistema operativo Android y se puede descargar desde un código QR (figura 6.1) o esta URL <https://build.phonegap.com/apps/1301030/install/3gEz3rz1xYG1yd9gz3Cf>, estando disponibles tanto para dispositivos Android como Windows Phone. La aplicación también está preparada para IOS, no estando disponible en estos momentos, dado que necesitamos de unas claves para desarrollar, de las que no disponemos. [33]



Figura 6.1. Códigp QR.

6.1 Resultados

A continuación se mostrará un ejemplo de uso de la aplicación desarrollada durante el trabajo fin de grado que también sirve como guía para el usuario. En este caso se ha realizado sobre un dispositivo Android pero también se puede realizar desde la Web dado que utilizan el mismo código.

En primer lugar aparece la pantalla de inicio (figura 6.2), donde se puede seleccionar ver los mapas de predicción o la evolución temporal para una localización concreta.

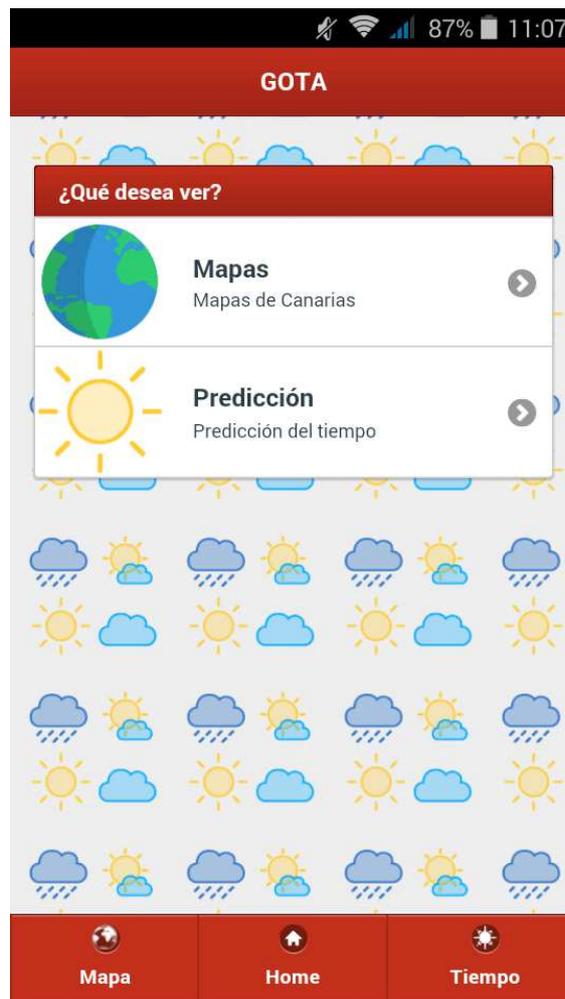


Figura 6.2. Pantalla de inicio.

Si seleccionamos ver los mapas de predicción (figura 6.3.) nos aparecerá la siguiente pantalla donde podemos seleccionar la fecha para la predicción, la región y la variable que queremos mostrar.

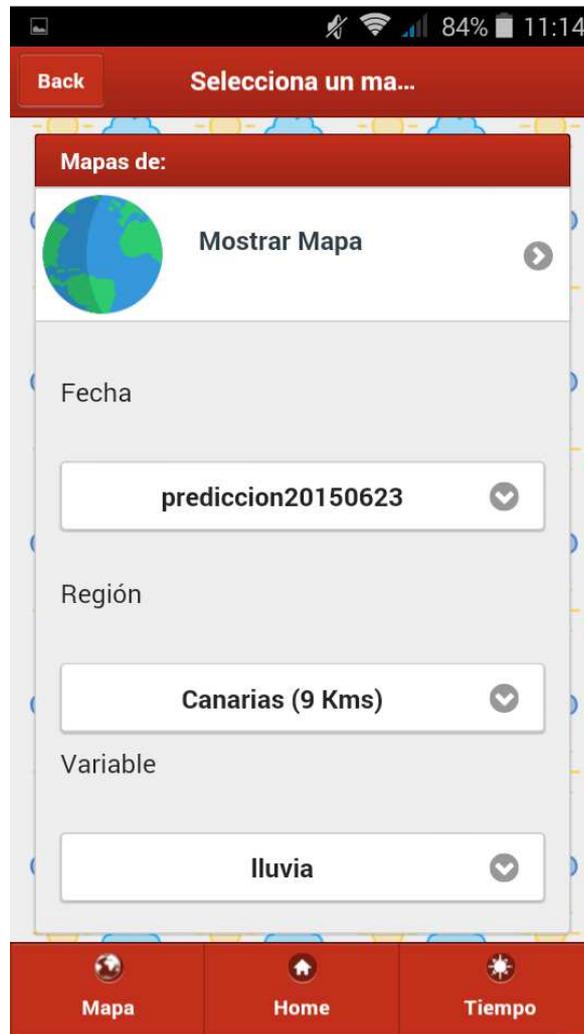


Figura 6.3. Selección de mapas de predicción.

Los mapas de predicción se muestran en las dos siguientes imágenes. En este caso hemos seleccionado la última predicción disponible, la región de Tenerife (1km) y la variable temperatura. El primer mapa como podemos ver corresponde a la hora actual y a través del slider se puede seleccionar las diferentes imágenes para las horas disponibles.

En la figura 6.4. vemos como se muestra la pantalla con el dispositivo en vertical y en la figura 6.5. podemos ver el mapa a pantalla completa girando el dispositivo de manera horizontal.

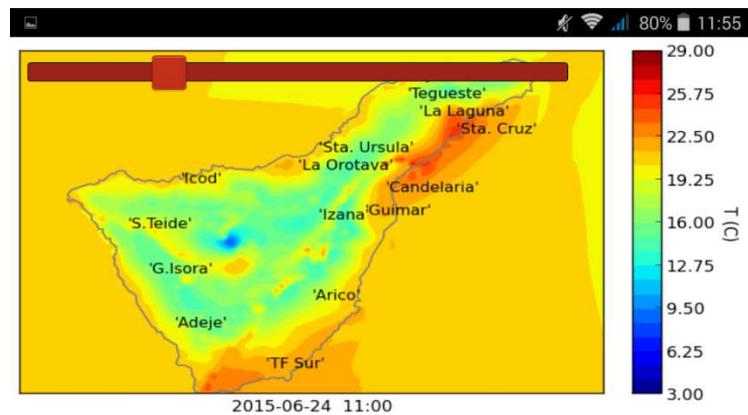
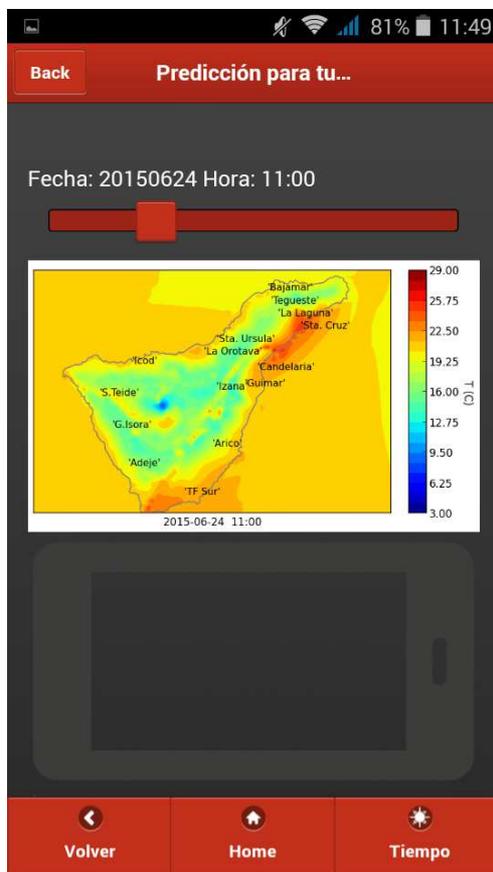


Figura 6.4. Mapa de predicción horizontal.

Figura 6.5. Mapa de predicción.

Para la evolución temporal de una localización disponemos de las siguientes opciones: por geolocalización, seleccionado la localización en un mapa o la lista de localizaciones favoritas (figura 6.6).



Figura 6.6. Menú evolución temporal.

La geolocalización y la selección de la localización se muestran en el mapa (Figura 6.7) que proporciona la API de Google Maps, donde tendremos la opción de ver la evolución temporal de las variables o guardar la localización en las favoritas.



Figura 6.7. Geolocalización.

La pantalla de favoritos (figura 6.8) muestra una lista de hasta tres localizaciones favoritas que previamente hemos guardado. Al hacer selección nos llevará directamente a la pantalla donde se muestra la evolución temporal de las variables.



Figura 6.8. Favoritos.

Y por último la pantalla donde se muestra la evolución temporal (figura 6.9) de las principales variables para una localización dada. En la parte superior se puede ver el nombre de la localización que estamos consultando y en el cuerpo una lista de la evolución temporal desde la hora actual hasta las 18:00 horas del día siguiente.

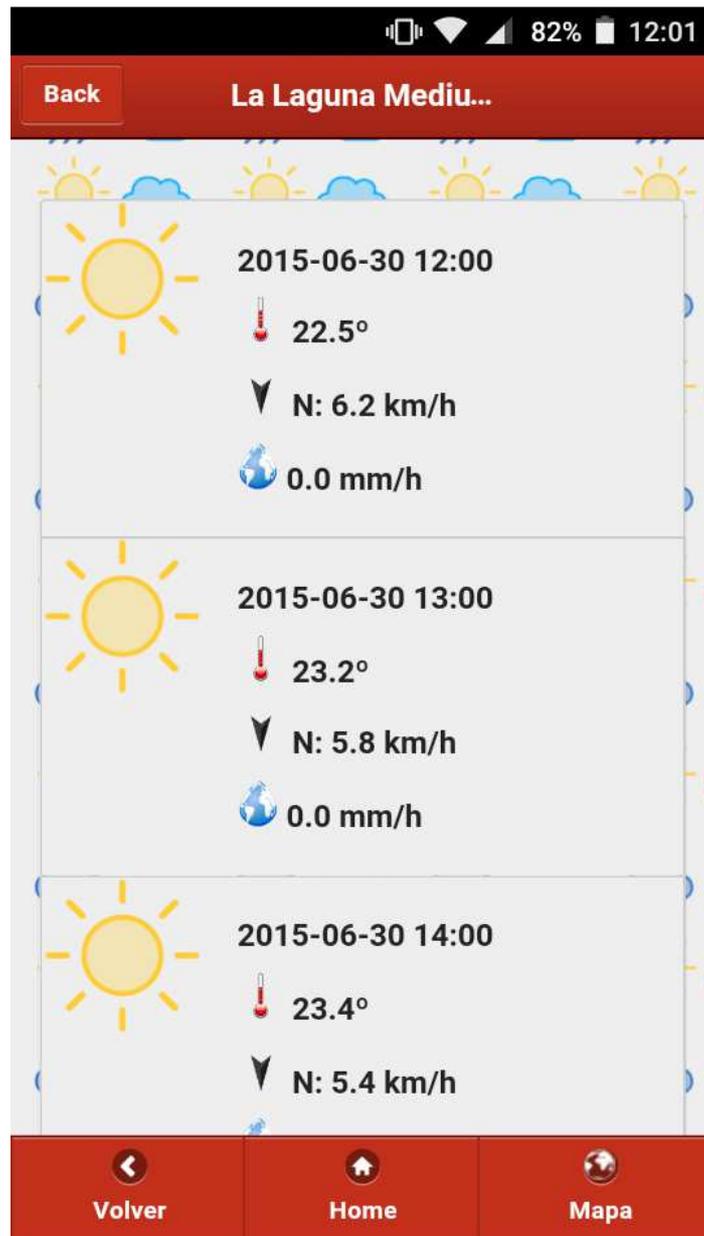


Figura 6.9. Evolución temporal.

Capítulo 7. Conclusiones y líneas futuras

En este último apartado se describen las conclusiones posteriores a la realización del Trabajo de Fin de Grado, a fin de revisar los objetivos que fueron planteados en los distintos apartados durante su realización y para comprobar el grado de satisfacción de cada uno de ellos. Tras este análisis se introducen algunas líneas futuras sobre un posible Trabajo Fin de Grado donde se podrían seguir avanzando funcionalidades del proyecto, en este caso, de la aplicación.

El presente Trabajo de Final de Grado ha tenido como resultado la creación de una aplicación móvil amigable y sencilla disponible para diversos tipos de dispositivos móviles. Ésta permite obtener información de las predicciones meteorológicas realizadas por el grupo GOTA, permitiendo visualizar mapas de predicción meteorológicas y mostrar una evolución temporal para una localización concreta.

7.1 Conclusiones

En este apartado se describen una serie de objetivos que se plantearon para la realización del Trabajo Fin de Grado y se exponen las conclusiones sobre su cumplimiento:

- Se ha llevado a cabo una evaluación de soluciones existentes, donde hemos analizado el desarrollo de aplicaciones (nativas, web e híbridas), la selección de la tecnología a utilizar (en este caso Phonegap, una tecnología híbrida de desarrollo multiplataforma) y el desarrollo de un análisis de requisitos para la aplicación.
- Se ha conseguido la creación de una interfaz con capacidad de mostrar la información necesaria para la selección de dominios y variables.
- Se ha conseguido que la aplicación muestre los mapas de predicción y se puedan seleccionar, tanto la fecha, como zona, variable y hora a visualizar.

- Se ha conseguido la presentación de la evolución temporal de las variables seleccionadas para un punto elegido sobre el mapa.
- Y por último, gracias a todos los objetivos planteados, se ha logrado el desarrollo de una aplicación multiplataforma, robusta y eficiente, disponible para la descarga mediante un enlace web.

Tras haber cumplido los objetivos principales planteados en un principio, se puede llegar a la conclusión de que se ha alcanzado satisfactoriamente los objetivos del Trabajo de Fin de Grado.

La realización de este Trabajo de Fin de Grado ha conllevado un amplio estudio sobre meteorología y nuevas herramientas como JQuery Mobile, Phonegap, PHP, entre otras. Además de ampliar los conocimientos en herramientas ya conocidas como, JQuery, HTML, CSS y JavaScript.

Durante el desarrollo de la aplicación se han presentado diversas dificultades a las que nos hemos tenido que enfrentar, como el desconocimiento total de algunas tecnologías como Phonegap y JQuery Mobile y dificultades en las consultas y transferencias de archivos, que hemos tenido que solventar (usando otro servidor a modo de puente).

Cabe decir que el presente Trabajo de Fin de Grado ha sido una labor enriquecedora, proporcionando y reforzando conocimientos. Gracias a éste hemos adquirido experiencia de cómo enfrentarnos a un problema real, como es el desarrollo de una aplicación.

7.2 Líneas futuras

Para finalizar, en este apartado se presentarán posibles líneas de futuro para esta aplicación. Éstas pueden ser estudiadas y desarrolladas con gran probabilidad, continuando con el trabajo ya realizado.

- Notificaciones: Se dispondrá de un módulo de notificaciones donde se le enviarán al usuario alertas meteorológicas sobre el estado del tiempo.
- Distribución: En los diferente mercados de aplicaciones, facilitando de esta manera la instalación y actualización.

Capítulo 8.

Summary and Conclusions

This section describes the conclusions of this Final Degree Project, in order to review the objectives raised in different sections and check the satisfaction of each one.

This Final Degree Project results in the creation of a friendly and easy mobile application available for different types of mobile devices. It provides information about the weather predictions made by the group GOTA, it allows displaying maps which show the weather prediction and temporal evolution from a particular location.

8.1 Summary and Conclusions

This section describes the objectives and conclusions raised to carry out the Final Degree Project:

- Different existing solutions have been evaluated, including the development of (native, web and hybrid) applications, the selection of the technology to be used (in this case Phonegap, a hybrid technology platform development) and development an analysis of requirements for the application.
- We have created an interface capable of displaying the necessary information for the selection of domains and variables.
- The application displays the prediction maps and you can select the date, the location, the variable display and time.
- The temporal evolution of the variables selected for a chosen point on the map is presented.
- And finally, thanks to all these objectives, we have succeeded in developing a multiplatform application, robust and efficient, available for download through a web link.

The main objectives that were raised at first have been fulfilled; we can conclude that it has successfully achieved the objectives of the Final Degree Project.

The realization of this Final Degree Project has led a comprehensive study of meteorology and new tools such as JQuery Mobile, Phonegap, PHP, among others. Besides, we had expanded the knowledge about tools known as JQuery, HTML, CSS and Javascript.

During the development of the application, various difficulties have been presented and we have faced them, as the unknown of some technologies such as Phonegap and JQuery Mobile and difficulties in queries and file transfers. We have had to overcome them in different ways, as using another server as a bridge.

We can say that this Final Degree Project has been an enriching work, providing and reinforcing knowledge. Thanks to this we have gained experience of how to deal with a real problem, as the development of an application.

Capítulo 9.

Presupuesto

En este capítulo se va a detallar el presupuesto para la realización de este Trabajo de Fin de Grado.

9.1 Presupuesto

A continuación, en la siguiente tabla (9.1.) se puede observar las distintas actividades realizadas durante el presente Trabajo de Fin de Grado indicando la duración en horas de cada una de ellas, con un total de 286 horas.

Para la estimación del cálculo hemos supuesto el sueldo que cobra un analista/programador freelance en una media de 25€ la hora. El resultado estimando del coste total de la aplicación es de 7150€.

Actividad	Duración (horas)	Coste (€)
Evaluación de soluciones existentes	20	500
Selección de la tecnología a utilizar	15	375
Análisis de requisitos	40	1000
Diseño	40	1000
Implementación	70	1750
Evaluación y pruebas	20	500
Elaboración de la memoria	70	1750
Revisión	11	275
Total	286	7150

Tabla 9.1. Presupuesto

Bibliografía

- [1] <http://es.wikipedia.org/wiki/Meteorolog%C3%ADa>
- [2] http://es.wikipedia.org/wiki/Modelo_num%C3%A9rico_de_predicci%C3%B3n_meteorol%C3%B3gica
- [3] <http://wrf-model.org/index.php>
- [4] http://es.wikipedia.org/wiki/Dispositivo_m%C3%B3vil
- [5] http://es.wikipedia.org/wiki/Tel%C3%A9fono_inteligente
- [6] http://es.wikipedia.org/wiki/Sistema_operativo_m%C3%B3vil
- [7] http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil
- [8] <https://play.google.com/store/apps/details?id=es.eltiempo.weatherapp>
- [9] <https://play.google.com/store/apps/details?id=aplicacion.tiempo>
- [10] <https://play.google.com/store/apps/details?id=es.aemet>
- [11] <http://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/>
- [12] <http://es.wikipedia.org/wiki/PhoneGap>
- [13] <http://es.wikipedia.org/wiki/JavaScript>
- [14] <http://es.wikipedia.org/wiki/HTML5>
- [15] <http://es.wikipedia.org/wiki/HTML>
- [16] http://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada#CSS3
- [17] <http://es.wikipedia.org/wiki/JQuery>
- [18] http://es.wikipedia.org/wiki/JQuery_Mobile
- [19] <http://revolucion.mobi/2013/05/03/aplicaciones-moviles-con-jquery-mobile/>
- [20] <http://es.wikipedia.org/wiki/Modelo%20%93vista%20%93controlador>
- [21] http://es.wikipedia.org/wiki/JQuery_Mobile

- [22] <http://jquerymobile.com/>
- [23] <http://themroller.jquerymobile.com/>
- [24] <https://www.iconfinder.com/search/?q=map&price=free>
- [25] <http://www.mikeafford.com/buy/free-icons-wind.html>
- [26] <http://www.softicons.com/web-icons/vector-stylish-weather-icons-by-bartosz-kaszubowski>
- [27] http://cordova.apache.org/docs/en/5.0.0/guide_cli_index.md.html#The%20Command-Line%20Interface
- [28] <https://build.phonegap.com/faq>
- [29] <https://build.phonegap.com/>
- [30] <http://api.jquery.com/jquery.getjson/>
- [31] <http://es.wikipedia.org/wiki/JSON>
- [32] <https://developers.google.com/maps/?hl=es>
- [33] <https://build.phonegap.com/apps/1301030/builds>