AireLimpioYA:

Aplicación móvil para conocer y difundir la calidad del aire de tu entorno

Trabajo de fin de grado

18 de junio de 2014

Autor:
Ismael Peral Álvarez

Director:
Jorge Garcia Vidal

Departament d'Arquitectura de Computadors
Facultat d'Informàtica de Barcelona

Agradecimientos

Este proyecto no hubiera sido posible sin la ayuda de los profesores Jorge García y José María Barceló, que han impulsado desde el principio esta iniciativa.

Tampoco hubiera sido posible sin el equipo de Ecologistas en Acción y las personas que han participado activamente en la lucha contra la polución. A ellos quiero agradecerles toda la ayuda y motivación que me han brindado, especialmente a Javier Martín, María García y Juan Bárcena como coordinadores del movimiento.

Me gustaría agradecer en este trabajo a mis compañeros Óscar Trullols, Joan Salvatella, Farnoosh Farokhmanesh y Enzo Brands por su sentido crítico y consejo diarios.

Por último, y no menos importante, quiero agradecer todo el apoyo que me ha brindado mi familia y mis amigos, no solo en este proyecto, sino en mi vida académica y profesional.

Abstract

Aunque la información sobre la calidad del aire de cualquier región de España es pública, el acceso a esta información es complicado ya que se encuentra distribuido en diversas páginas web y con diversos formatos.

En este trabajo se ha desarrollado una aplicación Android que muestra de forma unificada los datos de diversas fuentes de información sobre la calidad del aire. Esta aplicación añade, además, funcionalidades sociales para poder compartir esa información con otros usuarios a través de las redes sociales.

Although the air quality information of spanish regions are public, the access to this information is complicated because it is distributed on different websites and using different formats.

The goal of this project was the development of an Android application that displays, in a unified way, various sources of air quality information. This application also adds social features to share that information with other social networks users.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

Contents

En plena era de la información, donde cada vez es más sencillo conocer todo tipo de datos, resulta extraño que se siga produciendo un desconocimiento, por parte de la mayor parte de la población, de la existencia de problemas medioambientales. Esta falta de conocimiento provoca, a su vez, que estos problemas no sean tenidos en consideración y sean ignorados.

Con AireLimpoYa, se ofrece una alternativa a los proveedores existentes de información sobre la calidad de aire.

1.1. Formulación del problema

La mayoría de países se ven afectados por múltiples tipos de contaminación medioambiental, en el caso de este proyecto nos centraremos en la contaminación atmosférica o del aire.

La contaminación atmosférica se define como la acumulación, en el aire, de elementos gaseosos que perjudican a la salud humana y al medio ambiente en general. No siempre es fácil identificar este fenómeno. En ocasiones, se manifiesta en forma de niebla tóxica que flota por encima de las ciudades (acumulaciones de diversos gases perjudiciales como, por ejemplo SO_2 y NO_2), pero en muchas otras no es perceptible (como, por ejemplo, el O_3). Esto se puede producir en todo tipo de ambientes, tanto en ciudades como en contextos rurales.

La mayor parte de la contaminación del aire es obra del ser humano y se produce por la combustión ineficiente de combustibles fósiles o de biomasa; por ejemplo, los gases de escape de los automóviles, los hornos o las estufas de leña. Por este motivo, los organismos públicos deben regular y tomar medidas para fomentar la reducción de las emisiones de gases.

Por un lado, la OMS [1], mediante sus estudios, como por ejemplo el *Review of evidence on health aspects of air pollution Project* [12], define los niveles máximos de concentración de los elementos habituales en el aire, por debajo de los cuales no hay condición de riesgo para la salud humana.

Por otro lado, la Unión Europea, en consideración con los estudios de la OMS, regula estos niveles de concentración y establece unos límites legales por los cuales los países miembros pueden ser penalizados en caso de incumplimiento. Estos límites son más laxos que los definidos por la OMS debido

[1] Organización Mundial de la Salud

a la presión que realizan los países miembros con el objetivo de evitar penalizaciones.

España, como país de la UE, controla la calidad del aire con una red de estaciones de medidas distribuida por toda su geografía. Estas medidas son publicadas a través de las diversas páginas web de cada comunidad autónoma (horaria o diariamente, dependiendo de la región). Sin embargo su accesibilidad es muy cuestionable.

Pese a este control, el 94 % de la población española respiró, en 2012, aire contaminado, según los niveles establecidos por la OMS; y un 37 %, si tenemos en cuenta los niveles legales definidos por la UE según indica el estudio *La calidad del aire en el estado español durante 2012* [1]. Esta situación denota que los planes de reducción de gases contaminantes en España tienen unos resultados insuficientes y que existe una falta de información que permita a la ciudadanía prevenirse de una situación de riesgo para su salud como es la contaminación ambiental.

1.2. Objetivos

El objetivo principal de este proyecto es hacer partícipe a la sociedad de este problema, proporcionándole las herramientas adecuadas para conocer, difundir, ilustrar y actuar con respecto a esta información.

Según este objetivo, los puntos más importantes que creemos que debo cumplir este proyecto son:

- **Llegar al mayor número de personas posibles .** Enfocándose, especialmente, en aquellas personas que tienen inquietudes por conocer y tener más accesible la información sobre la contaminación atmosférica.
- **Simplificar y transmitir correctamente el mensaje.** Debe mostrar la información de la mejor forma posible, mejorando el mensaje que ya transmiten las opciones existentes.
- **Ofrecer herramientas para difundir.** importante que, para resolver este problema, se proporcionen herramientas de comunicación entre ciudadanos. Con ello fomentaremos el incremento de la participación del público en todo tipo de acciones contra la contaminación.

A lo largo de este documento se especifica en qué medida se logran estos objetivos, así como las decisiones que se han tomado para cumplirlas a partir de los recursos disponibles.

Esta aplicación beneficiará a todas las personas que se ven afectadas por la polución, especialmente aquellas que solicitan un mejor acceso a la para llevar a cabo acciones de protección. Asimismo, queremos ofrecerles las facilidades para poder difundir el problema al mayor número posible de personas.

Capítulo 2

Gestión de proyecto

Contents

2.1. Estado del arte

En esta sección se describe el contexto del proyecto, las soluciones tecnológicas existentes que tratan este problema y la selección de tecnologías. Con ello se razona el motivo por el cual se han elegido las tecnologías empleadas en el proyecto.

2.1.1. Contexto

Actualmente, la mayoría de empresas y entidades estatales publican mucha información de diversa índole a través de internet en forma de aplicaciones web o tecnologías similares.

En el tema en que nos focalizamos, la contaminación atmosférica, existen multitud de páginas y aplicaciones para conocer la calidad del aire en un periodo de tiempo relativamente corto. Como ya detallamos anteriormente, uno de los problemas de estas utilidades es su opacidad a la hora de transmitir la información y la falta de unión en cuanto a las fuentes de datos (cada comunidad autónoma posee su propia aplicación de calidad del aire).

Otra vertiente de información son los grupos ecologistas, ellos son los únicos que actualmente se encargan de coordinar acciones y difundir conocimientos relacionados con la calidad del aire. Sin embargo, a pesar de estas medidas, las acciones que promueven tienen poca repercusión para llegar a gente que no esta interesada en este tema.

Muchas son las acciones que se llevan a cabo. Por ejemplo, ecologistas en acción realizó una campaña de financiación comunitaria o crowdfunding [10] para AireLimpioYa y, de esta forma, conocer el nivel de demanda de un proyecto de estas características en el público en general.

La campaña de financiación fue un éxito, demostrando así la existencia de una demanda generalizada por la creación de una herramienta con unas características acordes con la actualidad. En este caso se comprometieron en el desarrollo de una aplicación móvil libre y gratuita que cumpliera con el objetivo de resolver el problema.

El proyecto ha sido financiado gracias a esta campaña de crowdfunding, con ello disponemos recursos suficientes para llevar a cabo este proyecto y ademas identificamos una razón de ser evidente, por tratar de resolver un problema real y reconocido.

2.1.2. Soluciones tecnológicas existentes

Como ya explicamos durante la formulación del problema, cada comunidad proporciona aplicaciones web y móviles para conocer el estado actual del aire. Seguidamente entraremos en detalle en las aplicaciones más relevantes con similitudes evidentes con este proyecto.

Este proyecto se nutre en mayor o menor medida de estas soluciones para obtener un resultado más ajustado a nuestras necesidades, replicando aquellas partes que consideramos que se han llevado a cabo adecuadamente y evitando o proponiendo soluciones alternativas para aquellas partes que no se ajustan.

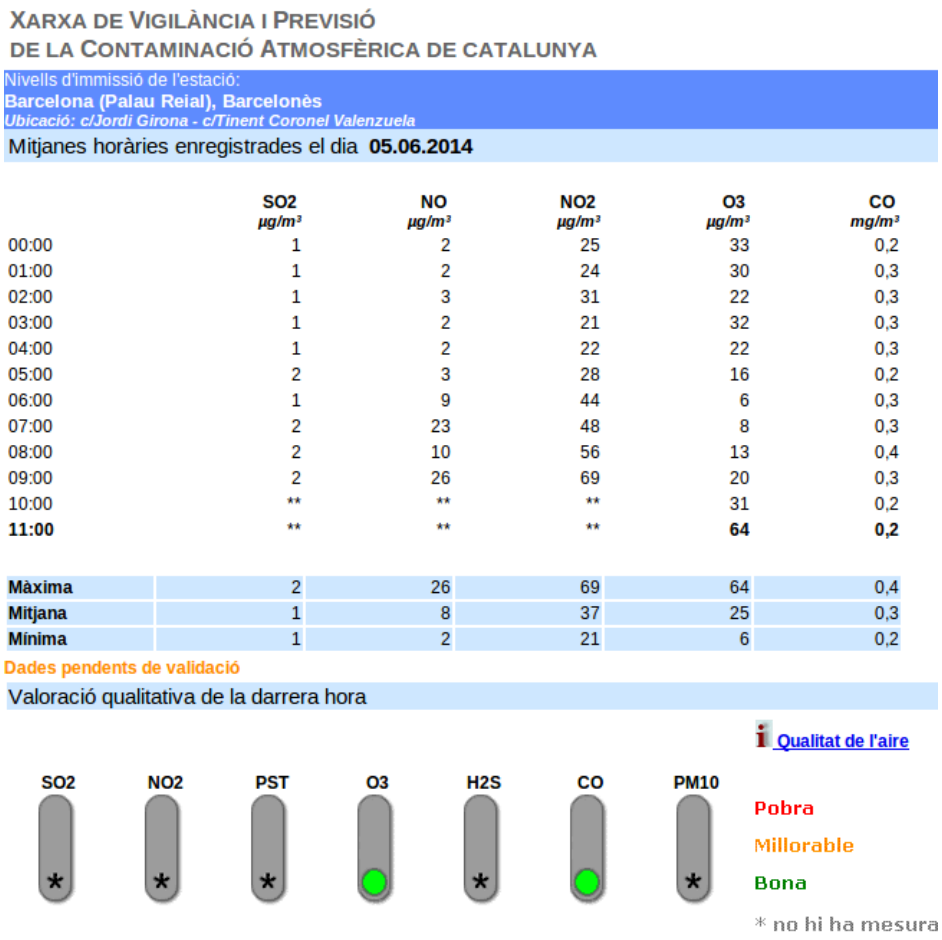
Aplicaciones Web

En este apartado analizaremos las aplicaciones web disponibles relacionadas con la contaminación atmosférica.

En este caso podemos suponer que este tipo de aplicaciones son proporcionadas por servidores web con un software que hace uso de algún lenguaje como, por ejemplo, HTML + CSS + JavaScript o PHP. Estos además posiblemente dispongan de una base de datos relacional como Microsoft SQL o mySQL; o tal vez no dispongan de los datos pero los pueden obtener a través de algún servidor externo que proporcione un servicio web con la información.

- **Web de la Generalitat de Catalunya** La web de la Generalitat de Catalunya dispone de un apartado específico sobre la vigilancia de la contaminación atmosférica [6]. Esta página proporciona acceso a la información de calidad de aire de estaciones situadas en toda Catalunya pudiendo elegir la estación a consultar por ciudad.

  Para acceder a esta página la opción más sencilla es disponer de la URL adecuada, pues acceder desde gentat.cat supone al menos cuatro clics y mucha lectura para abrir los menús adecuados. Otro inconveniente de esta aplicación es que no proporciona más información que los valores de las medidas y la valoración de estas medidas como buena, regular y mala.



XARXA DE VIGILÀNCIA I PREVISIÓ
DE LA CONTAMINACIÓ ATMOSFÈRICA DE CATALUNYA

Nivells d'immissió de l'estació:
Barcelona (Palau Reial), Barcelonès
Ubicació: c/Jordi Girona - c/Tinent Coronel Valenzuela

Mitjanes horàries enregistrades el dia **05.06.2014**

	SO2 µg/m³	NO µg/m³	NO2 µg/m³	O3 µg/m³	CO mg/m³
00:00	1	2	25	33	0,2
01:00	1	2	24	30	0,3
02:00	1	3	31	22	0,3
03:00	1	2	21	32	0,3
04:00	1	2	22	22	0,3
05:00	2	3	28	16	0,2
06:00	1	9	44	6	0,3
07:00	2	23	48	8	0,3
08:00	2	10	56	13	0,4
09:00	2	26	69	20	0,3
10:00	**	**	**	31	0,2
11:00	**	**	**	**64**	**0,2**
Màxima	2	26	69	64	0,4
Mitjana	1	8	37	25	0,3
Mínima	1	2	21	6	0,2

Dades pendents de validació

Valoració qualitativa de la darrera hora

Qualitat de l'aire

SO2 NO2 PST O3 H2S CO PM10

Pobra
Millorable
Bona
* no hi ha mesura

Figura 2.1: Vista de Gencat sobre la calidad del aire en Palau Reial

- **Web de la European Environment Agency** La web de la agencia europea de medio ambiente o EEA [3] proporciona información sobre las medidas de concentración de elementos contaminantes en el aire de toda la UE, con el inconveniente que esta información se publica con hasta 24 horas de retraso. La información está representada en un mapa y se puede identificar las zonas más afectadas, con una escala de valores alta y una leyenda para entender el grado de afectación.

  No es apta para ser visualizada desde dispositivos móviles, está diseñada para especialistas en calidad del aire y no se encuentra disponible en castellano.

Figura 2.2: Visión general que ofrece la EEA

Figura 2.3: Vista de EEA sobre la calidad de aire en Eixample

- **Web de AEMET** Se trata de la web del ministerio de medio ambiente de España sobre el clima [2]. Proporciona información en tiempo real de todo tipo de fenómenos atmosféricos medidos por estaciones situadas en el ámbito estatal. Ofrece información técnica en formato gráfico conjuntada con mapas donde posicionar los focos de medida.

  Como inconvenientes cabe destacar que esta aplicación no proporciona información para ciudadanos sin conocimientos medioambientales y que no es apta para ser visualizada desde dispositivos móviles.

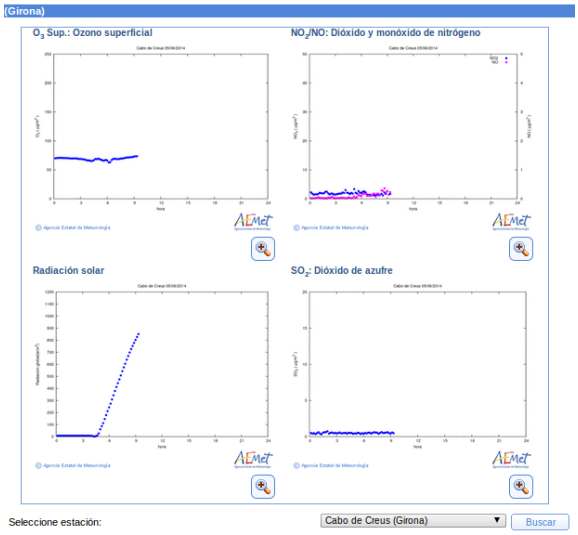


Figura 2.4: Vista de AEMET sobre contaminación de fondo en Cabo de Creus

Aplicaciones Móviles

Otra parte importante de las aplicaciones disponibles se encuentra en el mercado de los smartphones. Estas aplicaciones están diseñadas mayoritariamente para dispositivos con recursos limitados y con resoluciones relativamente pequeñas. La mayoría de aplicaciones no son libres como en el caso anterior, pero podemos aproximar, según la clase de smartphone, el lenguaje que se ha podido utilizar.

En el caso de los dispositivos con sistema operativo Android el lenguaje puede ser Java o C, mientras que, en el caso de dispositivos con iOS, concretamente iPhone y iPad, se debe haber utilizado Objective C. También cabe la posibilidad de que la aplicación sea una visualización de una página web en algún lenguaje con la combinación HTML + CSS + JavaScript.

- **Aire.CAT** Aplicación de la Generalitat, surgida recientemente, esta disponible tanto para Android [4] como para iOS[5]. Actualmente transmite parte de los contenidos de la web de la Generalitat descritos en la web del apartado anterior. Los datos están representados tanto en un mapa como numéricamente.

Como inconveniente, podemos destacar que solo muestra los datos de Ozono, Dióxido de Nitrógeno y Partículas PM_{10} de todos los que podría ofrecer cada estación.

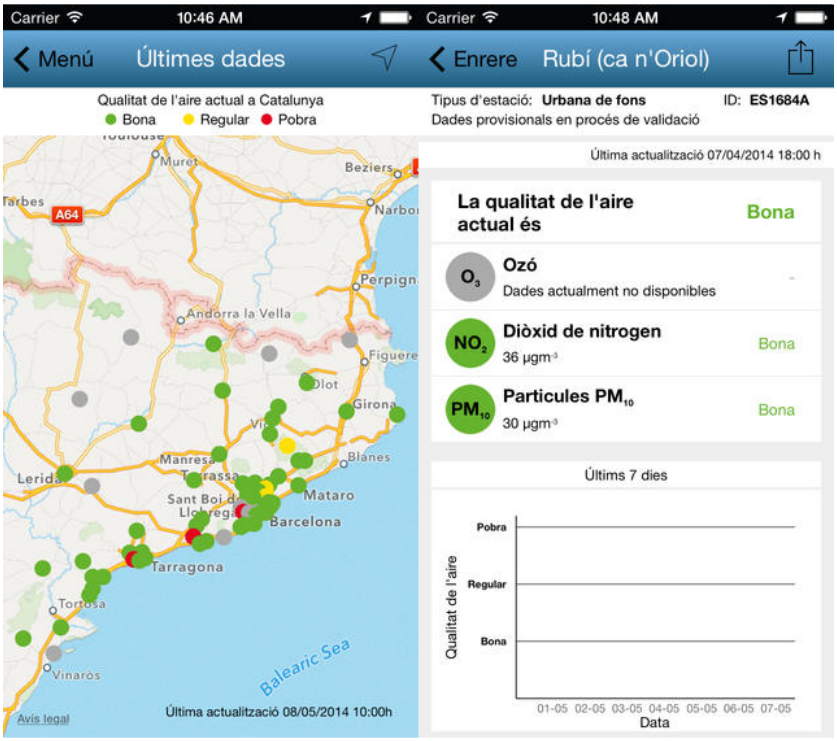


Figura 2.5: Vista de la aplicación Aire.CAT

- **Caliope** Un caso especial a tener en cuenta es el proyecto Caliope [14] desarrollado en el Barcelona Supercomputing Center. Este proyecto dispone de aplicaciones para smartphones con sistema operativo tanto Android como iOS.

  El objetivo de esta aplicación no es mostrar el estado actual de la contaminación atmosférica sino mostrar pronósticos, o predicciones, sobre la calidad del aire en el estado español.

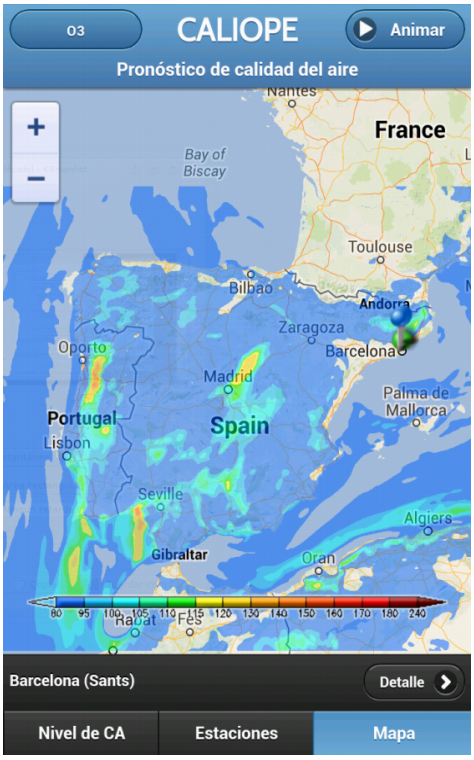


Figura 2.6: Vista de la aplicación CALIOPE

Otros servicios de información

Todos las aplicaciones anteriormente descritas se nutren de información de nodos de medida públicos estatales, no existe ningún servicio público donde toda la información sea pública y accesible y se pueda obtener de forma automática. Mención especial de la EEA [1], la aplicación web anteriormente descrita que dispone de una base de datos consultable a través de parámetros introducidos en la URL siguiendo una API propia. Pero dichos datos sólo pueden ser consultados por investigaciones, no están pensados para ser publicados.

Por este hecho, este proyecto analizará otros servicios de información que dispongan o puedan disponer de información referente a la contaminación atmosférica.

[1] European Enviroment Agency

- **Carriots / Xively** Es una plataforma web creada por la empresa Carriots S.L. [15] para almacenar información generada por nodos distribuidos, lo que conocemos por *Internet of Things*. La información que se pueda obtener de Carriots depende del usuario, es decir, si desplegamos estaciones de medidas de contaminación atmosférica y los configuramos para almacenar los datos en esta página podremos tener una fuente centralizada de datos. Otro ejemplo idéntico a este es la plataforma web Xively [11].

  Ambos casos tienen la posibilidad de mostrar las medidas almacenadas a través de su web pero no es un formato amigable con el usuario y únicamente proporcionan el servicio, pues el código no es libre.

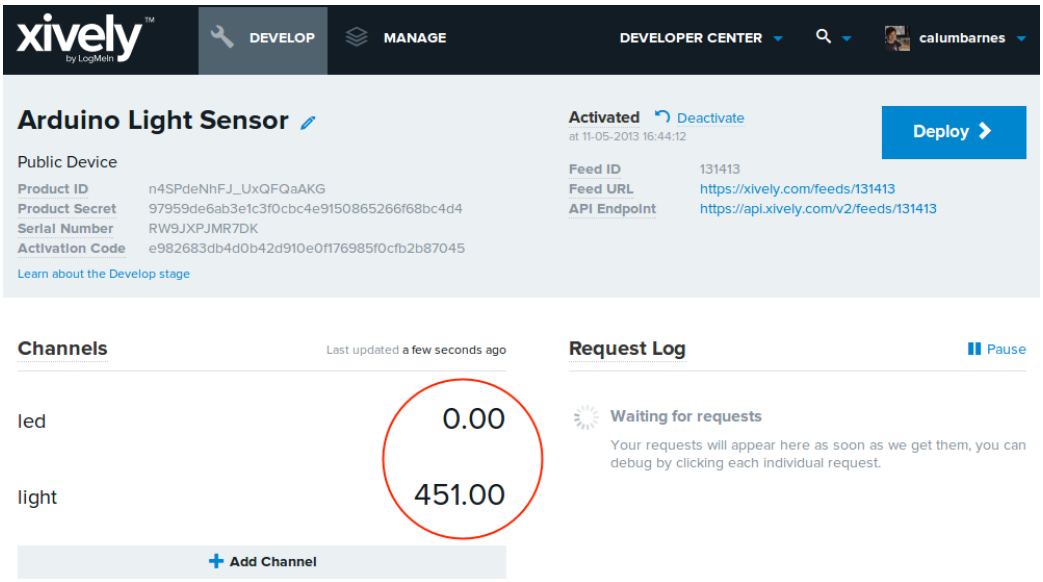


Figura 2.7: Vista del panel de control de Xively

- **CommonSense** Al igual que Carriots o Xively, CommonSense [7] es una plataforma que almacena información generada por nodos distribuidos, la diferencia radica en el tratamiento de la información. CommonSense es una red comunitaria de sensores donde sus usuarios pueden almacenar, publicar y consultar la información guardada en función de la política que decida cada usuario. Esta característica permite que comunidades de usuarios puedan disponer de una fuente de datos suficientemente valiosa como para ser alternativa a las estaciones públicas.

Por otro lado cabe destacar que es un proyecto de la UPC que se encuentra en fase en desarrollo, cuyo código fuente es abierto y está disponible para cualquier usuario que lo desee. Gracias a esta característica es un software mejorable y ajustable a multitud de plataformas.

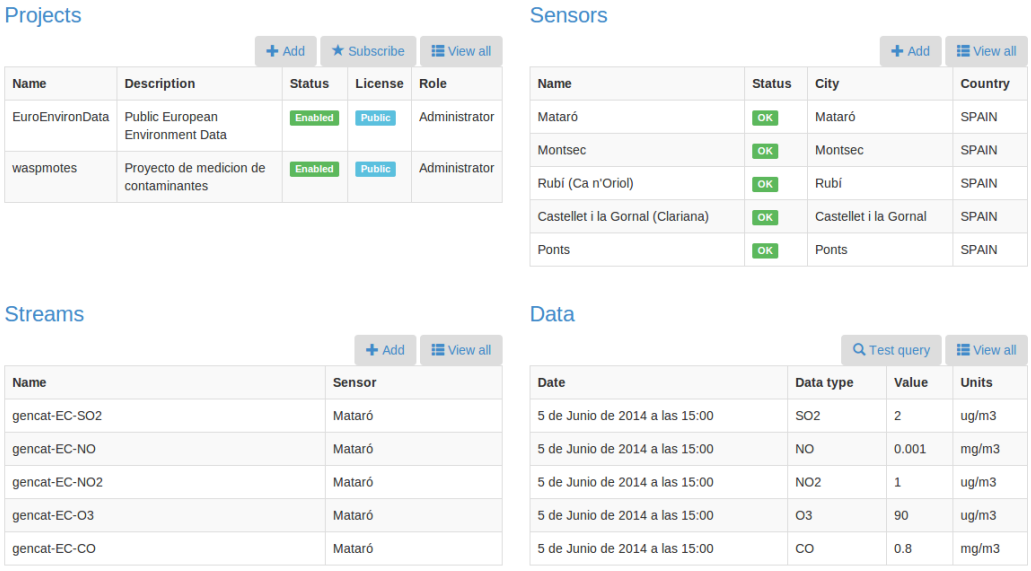


Figura 2.8: Vista del panel de control de CommonSense

2.1.3. Tecnologías seleccionadas

Todas las soluciones tecnológicas existentes no se ajustan a los requisitos que hemos determinado durante la formulación del problema. Podríamos plantearnos modificar o ampliar alguna de las ya existentes pero, por desgracia, no disponemos del código fuente de ninguna de ellas, dado que todas las aplicaciones son de proyectos cerrados.

Es por este motivo que este proyecto implementa el desarrollo de una aplicación completamente nueva para cumplir los objetivos. Esta nueva aplicación deberá ser una aplicación cliente enfocada al usuario final, esto implica un desarrollo de una aplicación móvil en cualquiera de los lenguajes compatibles escogiendo la opción que proporcione mayores funcionalidades, eficiencia y rendimiento.

En cuanto a la fuente de datos, el problema más tedioso, AireLimpioYa se nutrirá de la plataforma de información comunitaria que ofrece CommonSense. Si bien actualmente no dispone de información suficiente aportada por una comunidad, es posible ampliarla con información de las estaciones públicas gracias a su código abierto.

Utilizar CommonSense reduce las necesidades de AireLimpioYa de consultar varias fuentes de información, lo que implica proporcionarle una mayor escalabilidad a la aplicación resultante.

En cuanto al diseño visual, las soluciones analizadas en el apartado anterior sirven como punto de partida para la confección de AireLimpioYa, reduciendo así la fase de diseño que se explica posteriormente. Además se tiene en cuenta proporcionar información didáctica e ilustrativa que ningún otro software de estas características ha sabido dar con el fin de llegar a un mayor número de personas.

Cabe recordar que este proyecto no pretende mejorar ninguna tecnología, sino conjuntar una serie de servicios que actualmente no tienen relación alguna para adaptarlos a nuestras necesidades.

2.2. Alcance

Es importante determinar el alcance de este proyecto para conocer en que medida se cumplirán los objetivos. Esta sección cuantifica los límites que se marcan en este proyecto de la misma forma que más adelante se especificarán las acciones que se pueden realizar en siguientes extensiones del proyecto.

2.2.1. Alcance del proyecto

Recordamos que el objetivo principal es llegar al máximo número de personas, es decir, resolver el problema para el mayor número de personas. Con ese principio, se han tomado dos decisiones importantes de cara al alcance:

- **Ofrecer la información de Catalunya y Madrid.** Como ya describimos en la sección de tecnologías seleccionadas, CommonSense no dispone en principio de información suficiente de calidad del aire de todas las regiones de España. Por este motivo, AireLimpioYa garantizará el acceso a la información correspondiente a las dos regiones con mayor densidad de población de España.

- **Desarrollar la aplicación enfocándose en dispositivos con sistema Android.** Dado que este es un proyecto con recursos limitados, no podemos centrarnos en el desarrollo de una aplicación para todos los dispositivos disponibles. En un principio se planteó la posibilidad de desarrollar una aplicación web que pudiera ser accesible desde cualquier dispositivo, pero este tipo de aplicación requiere de recursos de mantenimiento más elevados, como por ejemplo la necesidad de tener servidores web para distribuir la aplicación. Por otro lado, se ha decidido desarrollar para dispositivos Android por la gran cantidad disponible, con la ventaja de que existe numerosa documentación disponible y el mantenimiento de dicha aplicación requiere un coste menor.

- **Mostrar al menos la información de O_3, NO_2 y PM_{10}** Se dispone de gran variedad de elementos medidos por cada estación, para no mostrarlos todos, en este proyecto nos centramos en mostrar los más importantes en cuanto afectación sobre la contaminación.

- **Ofrecer herramientas para la difusión social integradas con Facebook y Twitter** No es objetivo de este proyecto desarrollar una herramienta de comunicación más teniendo en cuenta la cantidad de servicios que hay disponibles. En este aspecto nos centramos en las dos plataformas sociales más importantes para tratar de difundir al mayor número de personas.

2.2.2. Obstáculos

El mayor obstáculo de AireLimpioYa es no disponer de una fuente de información propia, dado que desde un smartphone no se puede medir los niveles de concentración de elementos gaseosos en el aire. Este hecho provoca que en caso de fallo de los elementos externos que proporcionan la información, en este caso CommonSense, la aplicación se ve afectada de forma que no puede proporcionar el servicio adecuadamente.

Otro obstáculo lo encontramos en el ámbito de la difusión, AireLimpioYa proporciona las herramientas adecuadas para poder difundir pero no existe ningún plan de publicidad sobre la aplicación. Al no tener en cuenta este aspecto importante de la difusión, es probable que el impacto final sobre la población se vea reducido más allá de sus posibilidades. Para resolverlo, delegamos este aspecto a las diferentes asociaciones ecologistas.

2.2.3. Medida en que se resuelve el problema

Con este proyecto, ya con el alcance delimitado, resolvemos parte de los objetivos que nos marcamos inicialmente.

Primeramente en cuanto a la forma en que se muestra la información, mostramos la información de calidad de aire de las estaciones públicas de Catalunya y Madrid en una misma aplicación. Este aspecto deberá verse incrementado en un futuro gracias al aumento de información que pueda surgir en CommonSense (con usuarios que publiquen y compartan su propia información o añadiendo más estaciones públicas)

Ofrecemos la información de los elementos más comunes en cuanto la contaminación atmosférica (O_3, NO_2 y PM_{10}). Pero aún se debe ahondar en la necesidad de transmitir la información de otros elementos que ya recogen algunas de las soluciones que describimos en el apartado de estado del arte, no se han recogido los detalles de los otros elementos por falta de tiempo.

Fomentamos la difusión de la información a partir de las redes sociales más comunes, Facebook y Twitter. Este aspecto puede mejorar de muchas formas, como por ejemplo, ampliando la integración de otras redes sociales de gran impacto, siguiendo la línea de lo decidido en el alcance, como Google+, Whatsapp, entre otras.

Por último, nos focalizamos en los usuarios de dispositivos Android, dejando atrás multitud de usuarios con otro tipo de dispositivos que utilizan otro sistema operativo. Es importante que se incremente la expansión de usuarios desarrollando la aplicación para otras plataformas como podrían ser iOS, Windows Mobile, Blackberry OS, etc... O valorar la opción de realizar el desarrollo de una aplicación web con la mayor compatibilidad de dispositivos aunque eso pueda suponer un mayor coste de mantenimiento.

2.3. Metodología y Rigor

En esta sección se especifica la metodología de desarrollo de software que sigue este proyecto, la forma en que se ha validado y el seguimiento que ha tenido.

2.3.1. Metodología

Durante la realización de la documentación inicial se planteó llevar a cabo seguir una metodología de desarrollo en cascada en un solo ciclo. En el momento de llevarlo a la práctica, observamos que este método era poco ágil y poco flexible en cuanto a modificaciones durante el desarrollo.

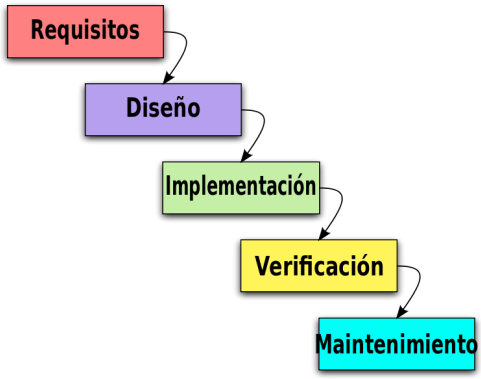


Figura 2.9: Metodología en modelo cascada img:wikipedia.org

El desarrollo de AireLimpioYA es dinámico: inicialmente se recopilaron una serie de requisitos que engloban las necesidades de la aplicación, pero al ser una aplicación para el público y no una empresa con un modelo claramente definido, surgen nuevos requisitos y posibilidades que inicialmente no son planteados.

Para hacer el desarrollo más ágil y flexible, descartamos seguir la metodología con el modelo en cascada y pasamos el desarrollo siguiendo una metodología con modelo iterativo y creciente. De esta forma nos adaptamos con mayor facilidad a posibles cambios y necesidades, aunque ello suponga modificar los requisitos o el diseño planteado inicialmente.

El concepto de este tipo de metodología consiste en realizar el análisis de las necesidades, y el diseño de la arquitectura siguiendo el enfoque de cascada.

Y posteriormente le sigue la realización del desarrollo iterativo de prototipos, que culmina en la verificación del prototipo final. Cada paso iterativo sigue un modelo mini-cascada de desarrollo.

Figura 2.10: Metodología de construcción de prototipos

Por consiguiente, los pasos determinados por la metodología escogida son: análisis de requisitos y diseño preliminar, una serie de iteraciones prototipadas compuestas por análisis, diseño, implementación y verificación.

En cada iteración se desarrolla una o más funcionalidades del producto final, este proceso puede durar entre una y cuatro semanas, dependiendo de los objetivos marcados en ella.

2.3.2. Validación y seguimiento

Para realizar la validación de la metodología es conveniente determinar quienes son los responsables del seguimiento y que roles adquieren.

Por un lado, tenemos el rol de cliente, que se lleva a cabo por los representantes de los usuarios finales, personas de disciplinas variadas no especialmente informáticas. En el caso de este proyecto, este rol lo asume Javier Martín como coordinador de ecologistas en acción.

Por otro lado, tenemos el facilitador, un ente similar al Scrum Master en la metodología especificada en SCRUM, que asegura que en cada iteración se haga el trabajo adecuado y asegura la resolución de posibles conflictos que puedan aparecer en los requisitos del cliente. Este rol lo asume Jorge García, como director del proyecto.

Por último, el equipo de desarrollo, encargado de realizar el trabajo sobre el producto. Este rol será asumido por un servidor como autor del proyecto.

El seguimiento se lleva a cabo mediante reuniones semanales con el facilitador, esto ayuda a mantener la planificación y no desviarse excesivamente de los objetivos de cada iteración. Adicionalmente se realiza un seguimiento mensual con el cliente, para actualizar los requisitos que van surgiendo, intentando que esta reunión coincida con el fin de cada iteración donde el producto final se haya visto incrementado.

2.3.3. Control

Al final de cada iteración se lleva a cabo un control del trabajo realizado en forma de reunión con el facilitador, o director del proyecto.

En esta fase se identifican las partes que han supuesto un coste mayor o menor al esperado. De esta forma, se puede ajustar la planificación final de forma más precisa, haciendo hincapié en las tareas que han supuesto un mayor coste, para evitar que produzcan retrasos en la medida de lo posible.

Otra de las partes a determinar durante la fase de control son los objetivos o tareas que se llevarán a cabo en la siguiente iteración. Para ello, se mantiene una reunión de todas las partes, donde se analiza el producto existente y se programa el desarrollo de la siguiente funcionalidad.

Con estas medidas se consigue mantener el cumplimiento de los plazos estipulados al tiempo que permite que los costes no sobrepasen la estimación. Paralelamente con esta metodología se consigue que el cliente esté informado en todo momento sobre el estado del desarrollo, pudiendo ser consciente del coste de añadir funcionalidades adicionales a las diseñadas inicialmente.

Al final del proyecto se puede verificar el coste total y comprobar que los objetivos planteados se han logrado adecuadamente. En cuanto al control sobre el impacto social y ambiental, también llamado sostenibilidad, será difícil de determinar durante las reuniones. También será difícil determinar este último punto al final del proyecto, es por ello que deberíamos dejar un tiempo prudencial para conocer el impacto social y ambiental que este proyecto ha tenido.

2.4. Planificación

Como consecuencia del cambio de metodología, la planificación se ha visto alterada según la planificación inicial. En esta sección describiremos la planificación final haciendo especial hincapié en los cambios realizados.

El proyecto se inicia en Febrero de 2014 y finaliza en Junio del mismo año. Durante ese periodo se realizaron las actividades que se describen en el siguiente apartado.

2.4.1. Definición de las actividades

Inicialmente se realizó la segmentación del proyecto siguiendo las fases comúnmente definidas en el modelo de cascada. Por ello, las dos primeras actividades, análisis de requisitos y diseño de sistema no se han visto alteradas.

Posteriormente a estas dos actividades, se han realizado cinco iteraciones que consisten en etapas donde se llevan a cabo las fases del modelo de cascada a pequeña escala. Por último se realiza la fase de validación en una vista más global para garantizar que se han cumplido los objetivos.

Estas son las tareas que se llevan a cabo en este proyecto, se encuentran descritas con mayor detalle en las secciones de especificación, diseño e implementación.

- **Análisis de requisitos:** Durante esta fase se determinan los actores, y las necesidades globales del software a desarrollar. Primeramente se realiza un análisis teniendo en cuenta las soluciones ya existentes para el problema. Seguidamente se mantiene una reunión con miembros de Ecologistas en Acción y para finalizar se sintetiza todo el análisis en una documentación adecuada para poder llevar a buen término las siguientes fases.

- **Diseño del sistema:** En el transcurso de esta fase se lleva a cabo un primer diseño global del software siguiendo los requisitos recolectados en la fase anterior. Al finalizar esta fase, se confecciona el documento con las decisiones tomadas respecto al diseño, intentando descomponer al máximo las funcionalidades de cara a la implementación.

- **Iteraciones durante el desarrollo:** Esta fase es la que se ha visto afectada durante la planificación. Al contrario de lo previsto, se segmenta el desarrollo en iteraciones agrupando fases en las que se lleva a cabo la implementación por grupos de funcionalidades. Esto ha permitido un incremento en el control del trabajo realizado y una mayor versatilidad respecto a cambios de última hora.
  Estas han sido las iteraciones que se han realizado definitivamente:

  - **Estructura de la aplicación**
  - **Obtención de datos**
  - **Almacenamiento de la información temporal**
  - **Detalles sobre una observación**
  - **Geolocalización**
  - **Integración con redes sociales**

- **Validación:** Durante esta fase se realiza un conjunto de pruebas para evaluar la integridad del sistema y se comprueba si se han cumplido los objetivos. Adicionalmente al finalizar esta primera etapa, se realiza una prueba con usuarios finales como verificación del trabajo.

2.4.2. Secuenciación de las actividades

Las actividades se han llevado en el siguiente orden: análisis de requisitos, diseño del sistema, las seis iteraciones de implementación de funcionalidades (estructura de la aplicación, obtención de datos, almacenamiento temporal, geolocalización e integración con las redes sociales) y validación.

2.4.3. Planificación final

El proyecto se ha llevado a cabo en el tiempo previsto pese a los cambios realizados en la planificación. En el siguiente diagrama se puede observar la planificación final considerando los tiempos reales dedicados en cada fase. Se incluyen también los contratiempos que se han producido, a pesar de que no han afectado a la consecución del proyecto puesto que se tomaron medidas preventivas durante la planificación inicial.

Es importante destacar que este proyecto se lleva a cabo con una dedicación semanal de 20 horas, normalmente dividido en 4 horas por día laborable (de lunes a viernes).

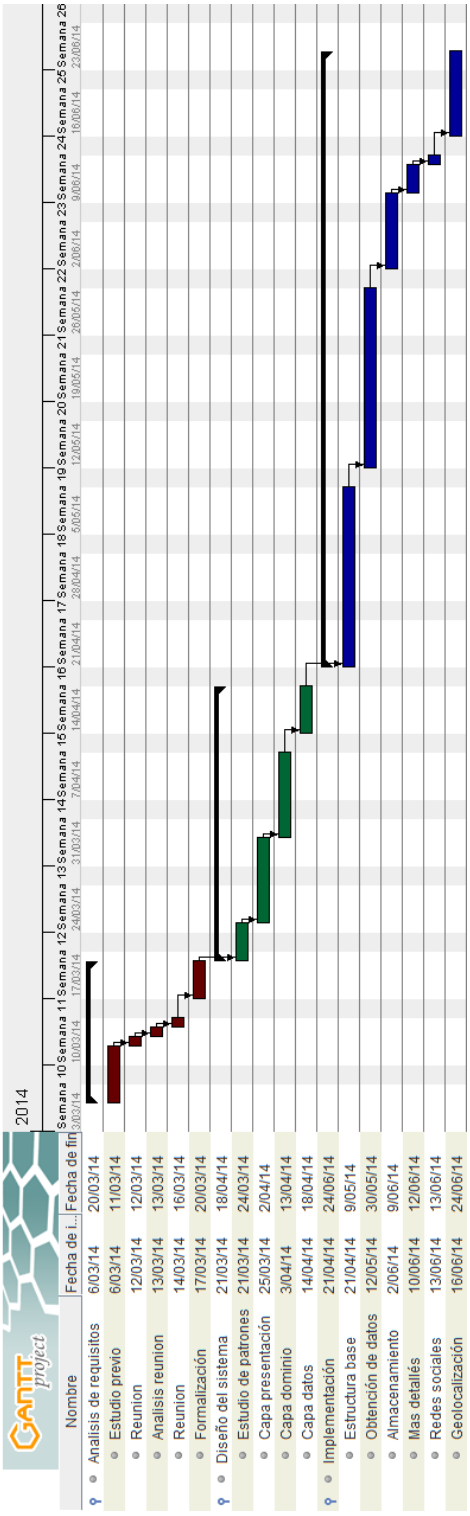


Figura 2.11: Planificación final

2.5. Presupuesto

Gracias a una correcta estimación inicial, el coste final se mantiene dentro de los valores detallados preliminarmente. En esta sección detallaremos todo el presupuesto desglosando los costes por cada recurso implicado y el coste global definitivo del proyecto.

2.5.1. Coste de los recursos humanos

El coste del personal, o coste de los recursos humanos empleados, está relacionado directamente con la planificación anteriormente descrita, concretamente:

Horas Empleadas x *Coste Económico del empleado/Hora*

En este proyecto se realizan las tareas propias de un gestor de proyecto, un analista y un programador. En este caso los tres papeles han sido llevados a cabo por la misma persona: el autor del presente proyecto. Hemos distribuido el tiempo empleado y las tareas en función del tiempo estimado que tendría que dedicarle cada uno de estos profesionales.

Rol	Horas	Participación estimada	Precio/hora	Coste total
Gestor de Proyecto	35h	10 % de total	50€/h	1.750€
Analista	125h	90 % analisis + 90 % diseño	40€/h	5.000€
Programador	190h	90 % desarollo + 90 % validación	35€/h	6.650€
		Coste de recursos humanos:		13.400€

Cuadro 2.1: Tabla coste de recursos humanos

2.5.2. Coste de los recursos materiales

En este apartado se muestra el inventario del material utilizado durante la realización de este proyecto lo desglosaremos por costes individuales.

Tratándose de un proyecto de desarrollo de software, el material consiste en un entorno de desarrollo. De este entorno podemos distinguir dos categorías: hardware y software.

Coste de hardware

El hardware empleado consiste en un computador portátil de gama media donde se ha llevado el desarrollo del proyecto, y cuatro móviles smartphone para realizar las pruebas correspondientes.

En la siguiente tabla se pueden observar los costes reales de los dispositivos, así como su amortización proporcional suponiéndoles una vida útil de 3 años (*tiempo empleado durante el proyecto / tiempo de vida útil*).

Producto	Precio/unidad	Cantidad	Coste Total	Amortización
Portatil Asus F751LD	570€	1	570€	80€
Smartphone Google Nexus 5	360€	2	720€	100€
Smartphone Motorola MotoG	180€	2	360€	50€
		Amortización aplicada al proyecto:		230€

Cuadro 2.2: Tabla de recursos materiales: Hardware

Coste de software

Gran parte del software empleado es de licencia libre y gratuita, no obstante, para la realización de las tareas ofimáticas también hemos hecho uso de software de pago por licencia. como es el caso de Microsoft Office, versión Hogar y Pequeña Empresa; y Adobe Acrobat Professional.

De la misma forma que en el apartado anterior, se ha aplicado un coste de amortización proporcional correspondiente a la duración del proyecto en relación al tiempo de vida útil del software. Hemos considerado 3 años de vida útil para el software, aunque es de esperar que tenga un tiempo de vida más largo.

Producto	Precio/unidad	Cantidad	Coste Total	Amortización
Microsoft Windows 7 Pro.	120€	1	120€	17€
Microsoft Office 2013 HyPE	260€	1	260€	36€
Adobe Acrobat XI Pro.	500€	1	500€	70€
		Amortización aplicada al proyecto:		123€

Cuadro 2.3: Tabla de recursos materiales: Software

2.5.3. Coste final

El coste final de AireLimpioYa es la suma de los costes anteriormente descritos. No se contemplan costes indirectos como puede ser el alquiler de un local, el pago de suministro de agua, luz o conexión a internet, ya que entendemos que no es computable.

La tabla siguiente muestra la suma de costes de forma global:

Concepto	Coste final
Hardware	230€
Software	123€
Personal	13.400€
Coste total del proyecto:	13.753€

Cuadro 2.4: Coste final del proyecto

Según la estimación de costes, este proyecto ha costado una suma de 13.753€. Si este proyecto fuera comercial, a esta cifra se le debería añadir una serie de beneficios esperados y los impuestos sobre el valor añadido correspondientes (un 21 %).

2.5.4. Viabilidad económica

En esta sección describiremos como se han asumido los costes.

Según lo anteriormente descrito en el apartado de Contexto, se realizó una campaña de financiación para hacer partícipe a la ciudadanía de este proyecto [10]. Como resultado de esta iniciativa se obtuvieron 1.250€ para sufragar costes.

Como esta cifra era insuficiente, y debíamos mantener la gratuidad del proyecto, se ha utilizado el entorno de trabajo personal del autor del mismo ha asumido los costes de oportunidad restantes, los correspondientes a los costes de recursos humanos. .

Gracias a todo ello se garantiza la viabilidad del proyecto actual.

2.6. Sostenibilidad y Responsabilidad Social

AireLimpioYa es un proyecto especialmente comprometido con la sociedad y el medio ambiente, siendo su objetivo principal el proporcionar las herramientas que fomenten la reducción de la polución.

En esta sección se identifican las medidas tomadas con respecto a criterios de sostenibilidad y responsabilidad social. Así como el impacto positivo que esas medidas tienen sobre los ámbitos social, ambiental y económico.

2.6.1. Criterios de sostenibilidad

Los criterios de sostenibilidad que enumeramos a continuación se pueden observar a lo largo de la realización del proyecto. Si existen unos criterios principales de mayor relevancia en el proyecto serían los siguientes:

- Fomentar la concienciación sobre la polución atmosférica.
- Ayudar a incrementar la difusión de conocimientos.
- Facilitar la movilización de la población en temas ambientales.

Otros criterios que tienen una gran importancia de la incidencia en la sostenibilidad, pero que no son los criterios que se han fijado principalmente son:

- Garantizar el uso responsable de los recursos.
- Permitir un mejor acceso a la información.
- Elegir una opción tecnológica libre.

Gracias a estos criterios se ha garantizado un impacto positivo en los ámbitos social, ambiental y económico.

2.6.2. Impacto social, ambiental y económico

El impacto del proyecto, como hemos indicado anteriormente, es difícil de determinar, por eso en esta sección realizaremos unas suposiciones para estimarlo.

Supongamos que el software AireLimpioYa es descargado por 50.000 personas que la usan regularmente y se previenen de la contaminación. Gracias a esta prevención, supongamos que un 1 % deja de visitar el centro de salud.

Teniendo en cuenta que el que el gasto medio anual en salud por ciudadano es de 1.211,1€ [13]. Se generaría un ahorro social y económico considerable.

Si suponemos que gracias a esas 50.000 descargas se incrementa la concienciación global sobre la necesidad de reducir la emisión de gases podríamos estar hablando de una reducción del nivel de polución en un 0.05 %, ya sea por el cambio de actitud de la ciudadanía o porque las entidades gubernamentales se sientan más presionadas a reducir ese nivel.

¿Cuales son, por tanto, los beneficios sociales, ambientales y económicos? Esta aplicación mejorará la calidad de vida, lo que supone, no sólo de ciudadanos sino también de seres vivos. Y por consecuencia una mejor calidad de vida supone un impacto económico beneficioso en cuanto a ahorro en medicamentos y visitas médicas.

En cuanto al impacto directo, el trabajo ha seguido una política de ahorro de energía, evitando así un consumo irresponsable. Se han seguido medidas como el uso de lámparas, y dispositivos de bajo consumo, como portátiles y smartphones.

Otro beneficio de esta aplicación es la reusabilidad del código. El proyecto es público y es de código abierto. Además, el código se encuentra rigurosamente comentado. Estas acciones permiten que futuros desarrolladores puedan ahorrar recursos en el desarrollo de herramientas, sirviendo el código aquí presente de ejemplo.

Capítulo 3

Estudio previo

Contents

En este proyecto se explica el desarrollo de una aplicación desde el inicio. En esta sección definiremos a un nivel un poco más técnico qué vamos a hacer, qué herramientas vamos a utilizar y quién esta implicado.

3.1. Descripción de las funcionalidades

Durante todo el proyecto nos centramos en el desarrollo de una aplicación software para dispositivos Android que cumpla los objetivos marcados en el alcance.

Podemos distinguir tres partes diferenciadas del desarrollo:

- **Lógica.** Es la parte donde se realiza la mayor parte del trabajo de las funcionalidades:
  - Comprobación del acceso a internet.
  - Comprobación de acceso al servicio de localización.
  - Comunicación con servicios web de información para la obtención de datos.
  - Evaluación de las observaciones obtenidas.
  - Comunicación con las aplicaciones de redes sociales.
  - Interacción con la funcionalidad de almacenamiento y de interfaz de usuario.
- **Almacenamiento.** Es la parte donde se realiza la gestión de almacenamiento, ya sea para cargar o guardar información. Es importante para guardar el estado actual de la aplicación ante la posibilidad de cierre o apagado. Se utilizan dos tipos de almacenamiento:
  - Información almacenada directamente en ficheros, tanto estática como dinámica. Especialmente para información sobre el estado.
  - Base de datos que sirve como almacenamiento temporal para el servicio de información externo.
- **Interfaz de usuario.** Es la parte donde se gestiona todo el apartado visual y de interacción con el usuario, disparadores, botones, colores, estilos, etc.

3.2. Descripción de las tecnologías empleadas

Las tecnologías que se han estimado oportunas para realizar estas funcionalidades son descritas en esta sección.

3.2.1. Java

Java es el lenguaje oficialmente soportado por Google para la creación de aplicaciones cuyo sistema operativo sea Android [1]. El uso de esta tecnología proporciona más disponibilidad de recursos compatibles y un mayor rendimiento. Además, existe multitud de documentación disponible al ser un lenguaje consolidado.

3.2.2. Android

Existe multitud de librerías para Android que se aplican a este proyecto, normalmente están escritas en Java, y estas permiten, mayoritariamente, interactuar con el sistema operativo. Podríamos destacar las siguientes:

- **android.location** Proporciona acceso a los datos de ubicación del dispositivo y a herramientas que permiten conocer la disponibilidad de los mismos.
- **android.net** Proporciona conocimiento sobre el estado actual de la conexion a internet.

3.2.3. SQLite

SQLite es el sistema de almacenamiento de base de datos integrado en Android, en la librería **android.database**. Este método de almacenamiento permite almacenar grandes cantidades de información de forma organizada, permitiendo así un rápido acceso a esa información. Es un lenguaje consolidado con multitud de librerías y documentación disponibles.

3.2.4. HTTP

Hypertext Transfer Protocol es el protocolo estándar más utilizado en las transacciones World Wide Web. Gracias a esta tecnología se puede hacer uso de servicios web, en este caso siguiendo el modelo **REST**[2]. El modelo REST

[1] Google es la empresa propietaria del sistema operativo Android
[2] Representational State Transfer

consiste en la combinación del protocolo HTTP para transportar mensajes y XML como lenguaje para el contenido de los mensajes.

Esta tecnología está soportada directamente por librerías incluidas en Android y tiene la ventaja de disponer de multitud de documentación dada su importancia.

3.2.5. XML

Extensible Markup Language es un lenguaje de marcas desarrollado por el W3C [3] y utilizado para almacenar datos en forma legible (especialmente entre computadoras). Permite almacenar datos de forma organizada y es comúnmente utilizado para proporcionar servicios web, como por ejemplo los que siguen el modelo REST explicado anteriormente.

3.2.6. RDF

Resource Description Framework es una familia de especificaciones de la W3C originalmente diseñado como un modelo de datos para metadatos. Es un modelo descriptivo que permite que la información en mensajes o ficheros con lenguajes estructurados como XML o JSON puedan representar relaciones y describir recursos.

Un contratiempo que hemos asumido sobre esta tecnología es que los servicios web con RDF no están suficientemente estandarizados en Android y, aunque las comunicaciones se realizan en contenido XML con el contenido modelizado con RDF, no se han podido aprovechar los beneficios tecnológicos que proporciona esta tecnología.

3.3. Tecnologías alternativas descartadas

Se ha contemplado la utilización de otros lenguajes de programación para el desarrollo de la aplicación, aunque finalmente han sido descartados. En esta sección los analizaremos, justificando porque no han sido elegidos.

- **C/C++.** Es un lenguaje muy extendido para la creación de aplicaciones. Google proporciona un paquete llamado NDK (Native Development Kit) que permite implementar las aplicaciones Android con

[3] World Wide Web Consortium

lenguaje C o C++. Siguiendo este lenguaje se consigue un mayor acceso al sistema a la hora de hacer debug, dado que internamente Android esta escrito en C; y proporciona facilidades a la hora de portar la aplicación a dispositivos con sistema operativo iOS, por la similitud de los lenguajes [4]. Su inconveniente, es la falta de documentación que supone un grado más de dificultad que representaría más horas de trabajo y, aunque mantenga mucha similitud con Objective-C, la portabilidad no es trivial.

- **HTML5**[5] Es un lenguaje de marcado utilizado para la elaboración de páginas web. Con la utilización de este estándar es posible elaborar una aplicación web compatible con la mayoría de dispositivos del mercado actual. Posee dos inconvenientes que nos ha hecho descartar esta tecnología: por un lado, ofrece un menor rendimiento frente a las aplicaciones nativas; y, por el otro, este tipo de aplicaciones dependen de un servicio web que proporcione el servicio, lo que supone un incremento en los recursos necesarios.

- **JSON.** JavaScript Object Notation es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que ofrece una alternativa a XML. JSON se ha descartado porque XML ofrece las funcionalidades que JSON podría ofrecer al proyecto y el entorno android recomienda XML para el diseño de la parte estática de interfaces de usuario.

3.4. Entorno de trabajo

3.4.1. Hardware

Para la realización de este trabajo se ha empleado un ordenador personal de características de gama media con el que se ha desarrollado y simulado la aplicación.

También se ha hecho uso de dos dispositivos móviles, también llamados smartphones, con acceso a internet (UMTS/HSPA y Wi-Fi) y que dispongan de componentes que permitan conocer la ubicación del mismo (GPS, o localización inversa) con los que se han realizado las pruebas en un entorno real.

[4]El desarrollo de iOS se lleva a cabo con el lenguaje Objective-C, este lenguaje guarda muchas similitudes con su antecesor C.

[5]Habitualmente se denomina HTML5 al uso conjunto de HTML, CSS y JavaScript.

3.4.2. Software

El software empleado en el desarrollo es de libre acceso y es el comúnmente empleado por los desarrolladores de android:

- **Android Developer Tools.** Es la plataforma de desarrollo oficial para crear aplicaciones compatibles con Android, esta basado en la plataforma de desarrollo de software eclipse y además integra todas las librerías oficiales de Android, así como el código de las diferentes versiones del sistema que pueden ser utilizadas para poder hacer simulaciones y pruebas sin necesidad de un dispositivo real.

- **Java SDK.** Es el kit de desarrollo de Java, necesario para el desarrollo de cualquier aplicativo Java y también necesario para la utilización del ADT[6].

- **SQLite Manager.** Es un gestor para bases de datos SQLite que permite administrar y supervisar los contenidos añadidos en la base de datos. Con el uso de esta herramienta se puede corregir errores que se hayan producido en este ámbito.

- **Git.** Es una herramienta de control de versiones con la que se puede realizar un seguimiento. Ofrece la posibilidad de guardar remotamente dichas versiones en un servidor externo. Habitualmente se utiliza el servicio que ofrece la web Gibhub, especial para el desarrollo comunitario de aplicaciones de código abierto, pero en este caso alojamos el código en un servidor del departamento de arquitectura de computadores de la UPC [7].

3.5. Actores implicados

Si bien hemos identificado personal con diferente rol a la hora de determinar la metodología, AireLimpioYa sólo contempla un actor para llevar a cabo las funcionalidades: el usuario final.

El usuario final puede tener diversos perfiles: usuario novato y usuario experto. Se les puede distinguir por el nivel de conocimiento. Durante este proyecto no hemos distinguido el perfil de los usuarios finales, es por ello que nos centraremos en los usuarios novatos o con menos conocimientos sobre contaminación ambiental.

[6] Android Developer Tools

[7] https://www.ac.upc.edu/projects/commonsense/git/commonsense

Capítulo 4

Especificación

Contents

La primera etapa de este proyecto ha sido la especificación, que se ha realizado justo después del estudio previo. En ella se ha llevado a cabo una serie de reuniones con el cliente o usuario final para determinar los requisitos y comprender exactamente que debería realizar el aplicativo.

La especificación se ha realizado en gran medida como se planificó inicialmente, pero al contrario de lo esperado se han realizado ligeras modificaciones en las diversas etapas de iteración durante el desarrollo, ajustándose mejor así a las necesidades.

4.1. Análisis de requisitos

Se han definido exactamente las funcionalidades y requisitos mediante dos reuniones con los usuarios finales. En esta sección se describe los requisitos tal y como se recogieron entonces. Posteriormente se desglosa en una serie de requisitos divididos por funcionalidades o pequeñas metas para la etapa de desarrollo.

La aplicación AireLimpioYa debe ofrecer la información más reciente disponible del estado de calidad de aire de una estación.

La estación puede ser elegida manualmente, introduciendo una región, ciudad y dirección válidas; o bien automáticamente mediante el uso de el localizador del dispositivo.

Un usuario puede tener una ubicación preferida, y aunque pueda consultar otras estaciones, en el funcionamiento habitual se le facilita esa dirección con más frecuencia.

La información más reciente disponible sobre el estado de calidad del aire puede estar formada por una o más medidas tomadas en una misma fecha y hora.

Una observación (o medida) debe estar formada por un componente químico, un valor y una valoración del valor.

Un componente químico tiene una notación científica (por ejemplo O_3) y un nombre común (Ozono), además tiene dos valores límites máximos en cuanto a recomendable para la salud: El límite ofrecido por la OMS[1] y el limite ofrecido por la regulación europea.

Una observación puede tener una descripción dependiente del componente y su valoración.

Una observación no positiva debe permitir realizar alguna acción. Una actuación puede ser compartir los datos de la observación a través de Twitter o de Facebook una o más veces, otra actuación puede ser acceder a una pagina web para firmar una petición contra la polución, o realizar una donación. Una última actuación puede ser ponerse en contacto directo con otra persona.

Otras características que debe tener la aplicación son, ser directa, evitando mostrar vistas e informaciones innecesarias, debe informar al usuario del estado en todo momento y de las situaciones anómalas que puedan surgir.

Se debe contemplar un diseño visual que permita que el mayor número de personas posible pueda acceder a el sin dificultad.

4.1.1. Requisitos funcionales

En esta sección se definen las funcionalidades del sistema. Se describe los detalles de cada funcionalidad en relación con los requisitos anteriormente descritos, se distingue el nivel de prioridad y las tareas que debe hacer esa funcionalidad.

Es importante destacar que no se ha analizado la carga de trabajo pues esta será analizada durante el desarrollo.

1. Seleccionar zona a consultar	
Como usuario quiero seleccionar la localización correspondiente a la zona que quiero consultar.	
Prioridad	Muy alta
Criterios de aceptación:	
• Quiero elegir una zona a consultar de entre una lista de regiones, ciudades y direcciones disponibles. • El selector debe mostrar la zona que hay seleccionada actualmente. • No debo poder seleccionar una zona donde no hayan disponibles estaciones de medida.	

[1]Organización Mundial de la Salud

2. Seleccionar automáticamente la zona más cercana
Como usuario quiero delegar la selección de la localización a la aplicación de forma que seleccione la mas cercana.
Prioridad | Media
Criterios de aceptación:
• Quiero que seleccione la estación de medida más cercana a mi posición. • La estación de medida seleccionada debe cambiar si cambia la posición y deja de ser correspondiente. • No debe tener en cuenta estaciones que estén a más de 200km.

3. Definir localización a consultar por defecto
Como usuario quiero definir una localización habitual para consultar que me simplifique la tarea de seleccionar esa ubicación.
Prioridad | Baja
Criterios de aceptación:
• Quiero que a la hora de seleccionar una ubicación me aparezca la ubicación por defecto en lugar de la mas reciente. • La ubicación por defecto debe ser una de las localizaciones de estaciones disponibles.

4. Mostrar la última observación disponible
Como usuario quiero visualizar la información correspondiente a las últimas medidas en la ubicación seleccionada.
Prioridad | Muy alta
Criterios de aceptación:
• Quiero poder visualizar las medidas más recientes disponibles de la ubicación seleccionada. • Quiero especial énfasis en los datos correspondientes a los componentes O_3, NO_2 y PM_{10}. • Quiero que se muestre el nombre completo y la notación científica de los elementos. • Quiero poder conocer la hora de la medida. • Quiero poder ver la dirección donde ha sido tomada la medida. • No quiero mostrar datos que tengan mas antigüedad que 24h.

5. Evaluar la observación consultada
Como usuario quiero que al visualizar una medida pueda conocer el significado del valor sin ser experto en polución.
Prioridad | Alta
Criterios de aceptación:
• Quiero conocer si la observación muestra valores recomendables para la salud o no. • Al menos debe poder evaluarse los componentes de O_3, NO_2 y PM_{10}.

6. Mostrar detalles de la observación sobre un componente
Como usuario quiero poder visualizar más detalles sobre una evaluación mostrada.
Prioridad | Alta
Criterios de aceptación:
• Quiero poder ver más detalles sobre el componente y dependiendo de la valoración que se haya hecho sobre la observación. • Al menos se debe obtener información complementaria de los componentes de O_3, NO_2 y PM_{10}.

7. Enviar mensaje a través de las redes sociales
Como usuario quiero publicar un mensaje indicando que se ha producido una medida poco recomendable para la salud.
Prioridad | Media
Criterios de aceptación:
• Quiero poder enviar un mensaje para que los contactos de mis redes sociales puedan conocer que se ha producido una situación negativa.. • Quiero poder redactar el mensaje que envío. • Quiero poder comunicarlo a través de Facebook y de Twitter por separado. • El mensaje no debe superar los 140 caracteres. • No se debe poder enviar un mensaje sobre una medida correcta.

8. Acceder a contenido externo a través del navegador integrado.
Como usuario quiero obtener más información y conocer otras formas de participar alternativas a las redes sociales.
Prioridad Baja
Criterios de aceptación:
• Quiero poder acceder a las firmas web iniciadas por ecologistas en acción. • Quiero poder acceder sitios web que publiciten participar en acciones contra la polución. • Quiero poder acceder a sitios web que permitan donar a la causa.

9. Recomendar la aplicación a otros amigos a través de las redes sociales.
Como usuario quiero enviar un mensaje a mis amigos para que conozcan AireLimpioYA.
Prioridad Baja
Criterios de aceptación:
• Quiero poder enviar un mensaje a mis amigos para que conozcan AireLimpioYa a través de Facebook. • No quiero tener que escribir el mensaje.

4.1.2. Requisitos no funcionales

En esta sección se exponen los requisitos que no cumplen ninguna funcionales pero son necesarios para realizar un aplicativo adecuado.

1. Usable
Descripción:
El sistema debe ser simple de utilizar e intuitivo para los usuarios a los que va dirigido.

2. Robusto
Descripción:
El sistema debe mantener su integridad, por ello debe garantizar que no se producen errores. Además el usuario no debe poder introducir datos incorrectos.

3. Escalable
Descripción:
Las funcionales del sistema pueden ser ampliadas o modificadas sin afectar al resto de funcionalidades.

4. Ampliamente compatible
Descripción:
El sistema debe ser compatible con la mayoría de versiones de sistema, especialmente la versión 4.x.

5. Diseño para todos los públicos
Descripción:
El sistema debe estar diseñado para llegar al mayor número de personas, evitando dificultades.

6. Multi-idioma
Descripción:
El sistema debe estar preparado para mostrar las instrucciones en varios idiomas o traducirlo con facilidad.

4.2. Modelo conceptual

Esta es la especificación del modelo conceptual se ha obtenido a partir de los requisitos, pero se ha ido incrementando durante el desarrollo del proyecto con cada iteración, ampliando y perfeccionando este modelo.

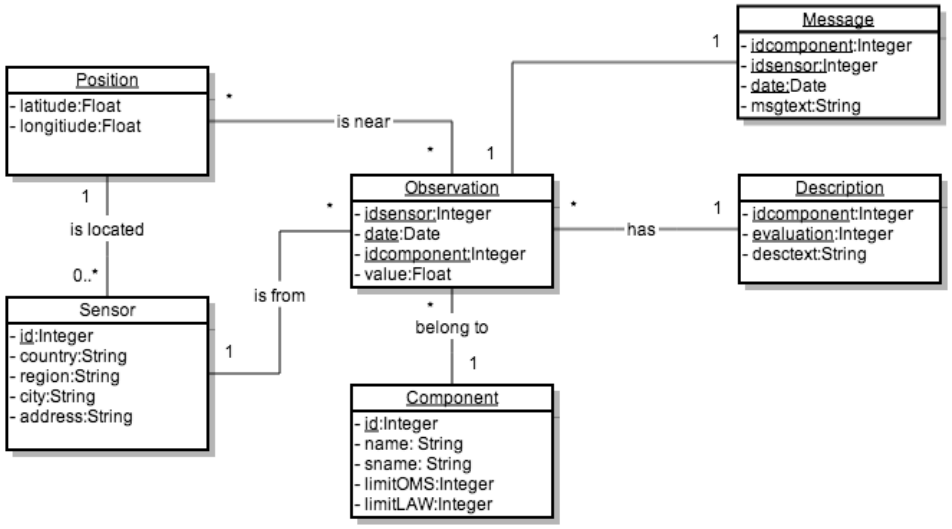


Figura 4.1: Especificación del modelo conceptual de la aplicación

Algunos flecos sobre este modelo son los siguientes:

- A una posición le pueden corresponder varias medidas, esto es porque las estaciones tienen un valor simbólico sobre una zona.
- Una observación dispone de un mensaje por defecto listo para enviar.
- Un observación tiene una descripción de entre 3 descripciones posibles por cada componente segun el resultado de la evaluación que se haya hecho.
- Un sensor, o una estación de medida, puede tener la misma dirección, aunque habitualmente supondremos que solo hay un sensor por posición.

4.3. Modelo de casos de uso

En esta sección se recogen las actividades que puede hacer el actor principal. El actor principal es el usuario final y principalmente puede hacer uso de tres funcionalidades: seleccionar zona, pedir que se muestren las observaciones y enviar un mensaje a las redes sociales.

Cabe destacar que las funcionalidades en este nivel se encuentran descritas de forma muy global, pues durante el diseño y el desarrollo se detallada profundamente estas funcionalidades tanto a nivel interno (dentro del sistema) como externo (enfocado a las funcionalidades concretas hacia el usuario final).

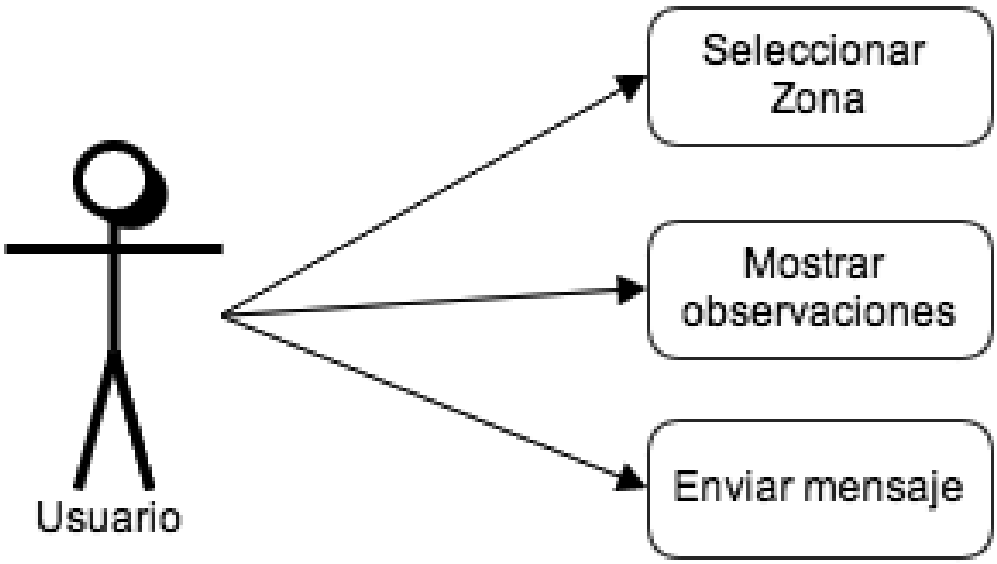


Figura 4.2: Diagrama de casos de uso de las funcionalidades

4.4. Modelo de comportamiento

Según el diagrama de casos de uso tenemos definidos los siguientes diagramas de secuencia para las funcionalidades entre actor y sistema. En la etapa de diseño estos diagramas se especifican con un mayor detalle.

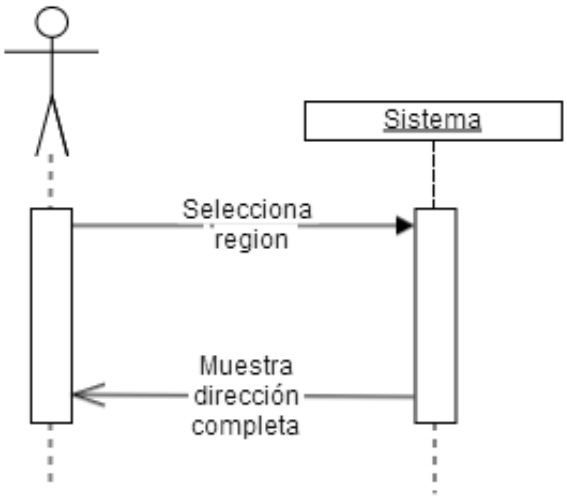


Figura 4.3: Diagrama de secuencia de selección de ubicación

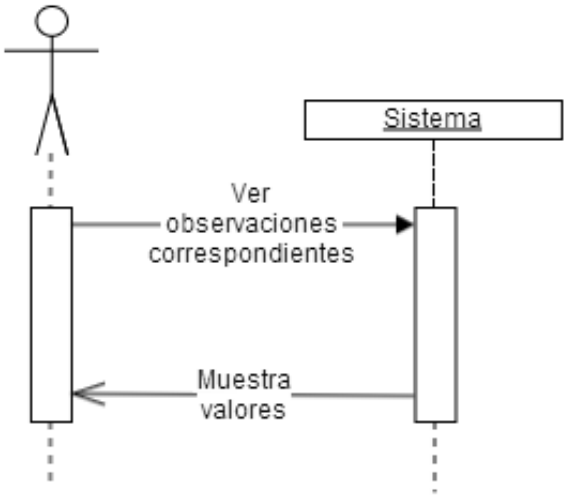


Figura 4.4: Diagrama de secuencia de actualización de observaciones.

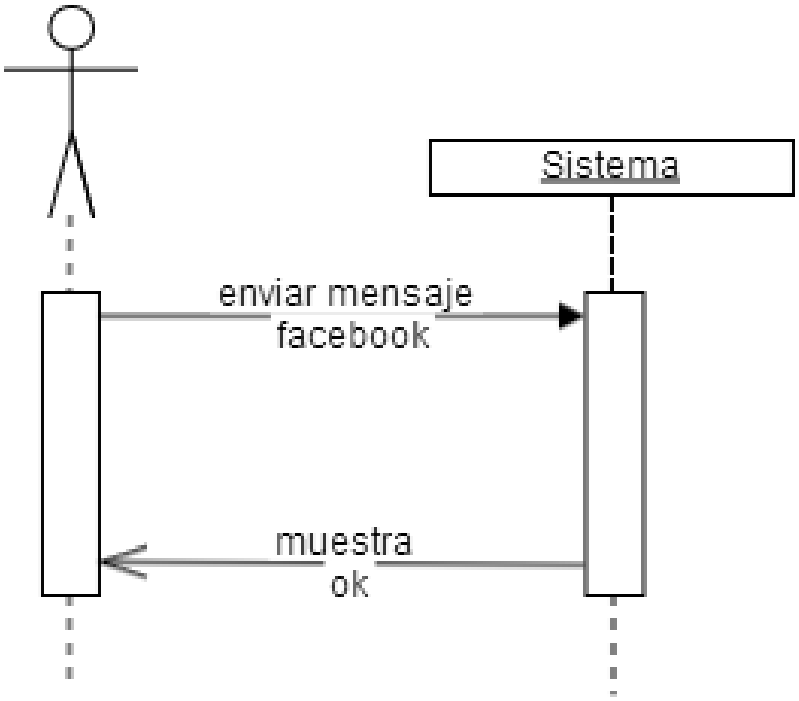


Figura 4.5: Diagrama de secuencia de envío de mensaje en redes sociales

Capítulo 5

Diseño

Contents

El diseño del sistema es la etapa que se ha llevado posteriormente a la especificación. En esta sección se describen los patrones de diseño que se han seguido, y se separa el desarrollo siguiendo los patrones y características utilizadas.

5.1. Patrones de diseño

Para el diseño de AireLimpioYa se ha hecho uso de los patrones de diseño Modelo-Vista-Controlador, Categoría y Singleton. También se han seguido patrones de diseño por Android oficiales [9] y [8].

5.1.1. Modelo-Vista-Controlador

El principal patrón de diseño utilizado en las aplicaciones es el Modelo Vista Controlador (MVC). Este patrón de diseño de software consiste en la separación de la interfaz de usuario, la lógica de control y los datos de la aplicación.

La interfaz de usuario es la parte que se encarga de mostrar información e interactuar con el usuario, normalmente lo podemos encontrar en las Vistas. Los datos de la aplicación es el modelo que se encarga de almacenar y gestionar la información. Mientras que la lógica realiza las operaciones y controla la gestión de lo que debe ser representado y almacenado.

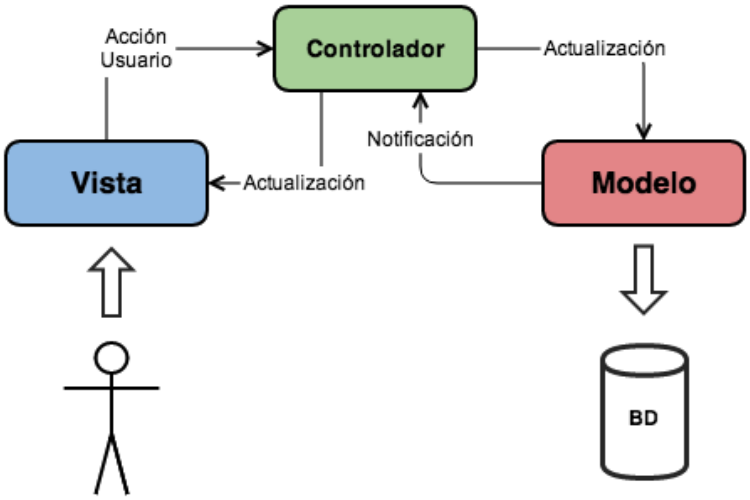


Figura 5.1: Patrón Modelo Vista Controlador

Gracias a este patrón de diseño se consigue mayor flexibilidad, reusabilidad y escalabilidad gracias a tener tres capas bien diferenciadas. En este capitulo veremos en detalle el diseño que se ha llevado en cada capa.

5.1.2. Composite

El patrón de diseño Composite permite la posibilidad de añadir métodos extras a las clases ya existentes. De esta forma se ofrece una soluciones alternativas a situaciones sin la necesidad de rehacer específicamente algún método o clase. Este patrón esta soportado directamente por Java.

Es un método que se utiliza frecuentemente para organizar la implementación de las clases y se ha utilizado en determinadas ocasiones durante todo el proyecto como se podrá observar en la fase de desarrollo.

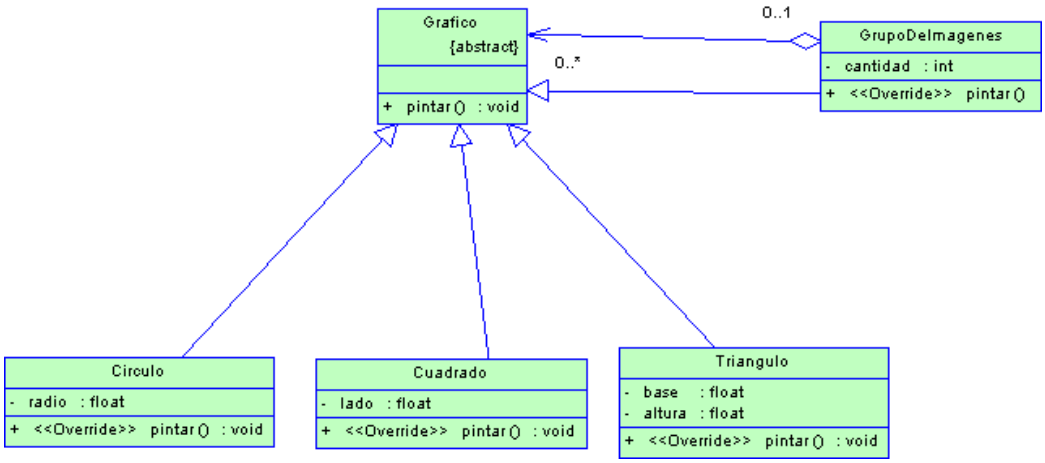


Figura 5.2: Patrón Composite img:wikipedia.org

5.1.3. Singleton

El patrón de diseño Singleton (instancia única) restringe la creación de objetos pertenecientes a una clase en concreto. El objetivo de este patrón es garantizar que esta clase solo tenga una instancia y se pueda acceder a ella globalmente.

El patrón Singleton obtiene la clase método en una instancia y en caso de que esta no exista la crea. El constructor se debe encargar de regular el acceso a la misma para evitar problemas de concurrencia pudiendo hacer uso de atributos de diversos tipos (como permisos de lectura o escritura).

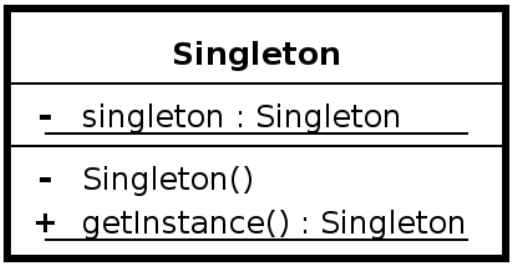


Figura 5.3: Patrón Singleton img:wikipedia.org

5.2. Capa de presentación

Según el patrón Modelo-Vista-Controlador, la capa de presentación es la parte que interactúa con el usuario. Es la parte visible de la aplicación tanto por vista como por navegación.

5.2.1. Diseño visual

El diseño gráfico se ha llevado a cabo en dos fases, una inicial durante la planificación esperada para diseño y una posterior durante las iteraciones de desarrollo. Gracias a esta metodología, el resultado final se ha ajustado mejor enfocándose en un usuario final mas general.

Inicialmente distinguimos tres vistas: Una vista principal que permite elegir el lugar (si no se encuentra activo el acceso a la ubicación del dispositivo que se seleccionaría automáticamente) y permite ver las medidas más recientes de esa ubicación siguiendo los requisitos descritos anteriormente; una segunda vista que permite obtener más información sobre alguna medida; y una tercera vista que permite reaccionar ante una medida perjudicial para la salud.

Esta es la primera versión de la primera vista inicialmente diseñada. Se trata de una vista directa, con un tamaño de fuente suficientemente grande para llegar a un gran número de personas, pero aun podemos ver carencias como por ejemplo el uso de notación científica en lugar del uso del nombre completo.

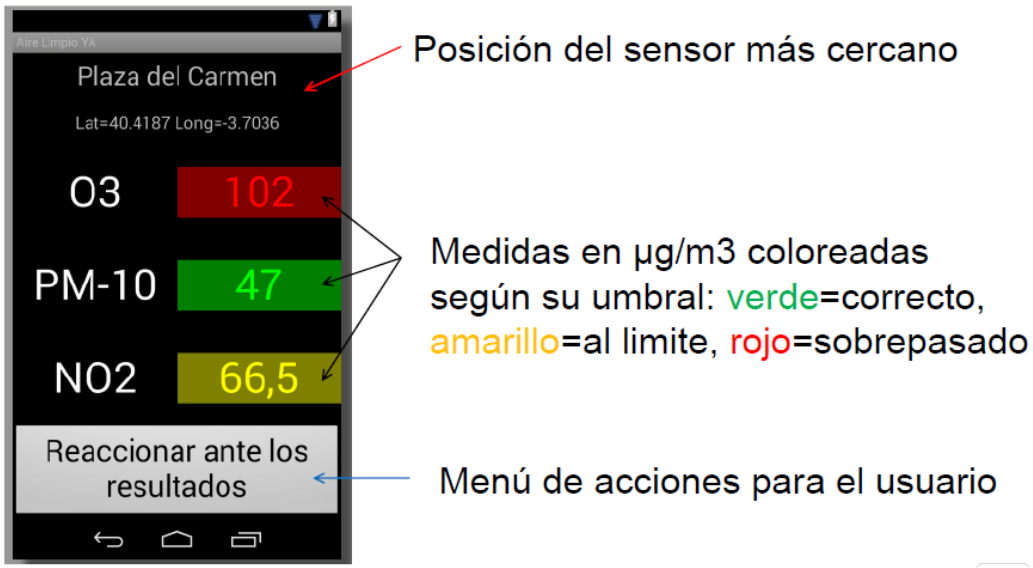


Figura 5.4: Vista prototipo de observaciones más recientes.

En el caso de la segunda vista, se mantiene la misma carencia aunque puede cumplir con el requisito por el cual se confecciona: proporcionar más información).

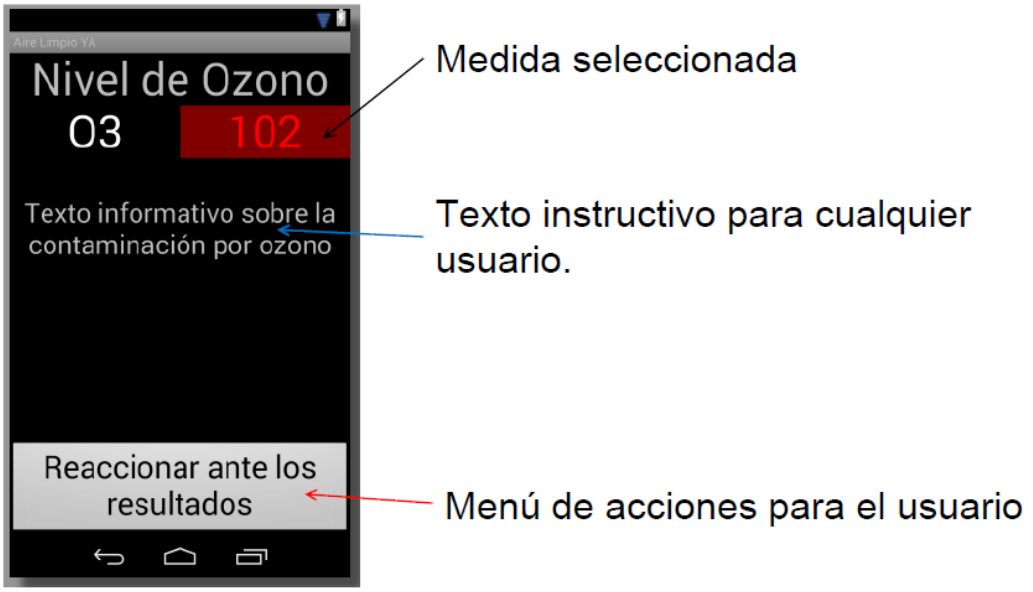


Figura 5.5: Vista prototipo de detalles sobre una observación

Y en el caso de la tercera vista, cumple con muchos de los requisitos aunque la interfaz no es amigable con el usuario con más semejanza a un panel de control más que a una herramienta que permita actuar.

Figura 5.6: Vista prototipo de acciones difusoras.

Vistas actuales

Durante la etapa de desarrollo se ha perfeccionado el diseño para solventar las carencias del diseño inicial. El procedimiento de implementar las funcionalidades ha podido reajustar los requisitos para incrementar los resultados finales.

Adicionalmente hemos mejorado el estilo de la aplicación para seguir el tema principal de las utilidades desarrolladas para smartphones Android. Se ha cambiado tanto textos como iconografias ajustando a las recomendaciones de tanto el director como usuarios finales.

Un ejemplo de modificación es la incorporación de una vista adicional al diseño inicial. Esta vista permite elegir la localización automáticamente o de entre tres listas cerradas con una combinación de la dirección por región, ciudad y dirección.

Este criterio se ha añadido por la necesidad de mostrar más de 50 estaciones de una forma cómoda y distribuida independiente de la cantidad de opciones que aparezca.

Figura 5.7: Vista actual de selección de la ubicación.

Otro ejemplo en el que se pueden destacar grandes cambios es en la vista que proporciona los datos sobre las observaciones actuales. Ahora se constata la hora en que se tomaron las medidas y todos los nombres tienen su formulación cientifica y su nombre completo.

Figura 5.8: Vista actual de observaciones más recientes.

La vista sobre detalles ha sufrido dos cambios destacables, se muestra el nombre completo del elemento y el botón que permite actuar se muestra por delante del texto descriptivo, dando así opción a este texto ser de mayor tamaño que la resolución del dispositivo.

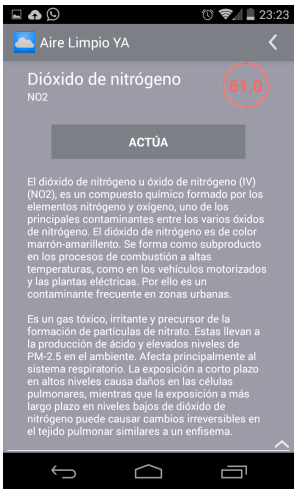


Figura 5.9: Vista actual de detalles sobre una observación.

La vista sobre difusión de una medida se ha ajustado de mejor forma a las necesidades, se ofrece datos sobre la medida que se va a compartir así como el mensaje. Además como anteriormente hemos destacado sigue la el estilo de la iconografía de diseño de aplicaciones Android.

Figura 5.10: Vista actual de medidas posibibles de actuación.

Por último, una vista no tenida en cuenta inicialmente en los diseños, pero si en los requisitos para seguir con la política de diseño de Android es la siguiente. Se trata de una vista de ajustes que permite configurar las opciones en curso de la aplicación.

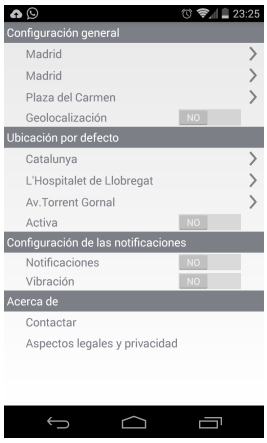


Figura 5.11: Vista de configuración y ajustes.

5.2.2. Mapa de navegación

La aplicación tiene diversas funcionalidades pero el mapa de navegación se puede describir principalmente en una linea horizontal.

Figura 5.12: Storyboard de las vistas.

Este diagrama muestra la relación entre las vistas en el flujo habitual. De funcionamiento.

Cabe destacar que en este Storyboard no esta incluido el flujo en las aplicaciones de terceros. Estas pueden ser Facebook, Twitter o Google Chrome.

Por otro lado también se dispone de la vista de ajustes, que es accesible desde la vista de observaciones. Esta vista puede suplir la necesidad de volver a la vista anterior.

Figura 5.13: Storyboard vistas hacia ajustes.

5.3. Capa de dominio

La capa de dominio según el patrón Modelo-Vista-Controlador es la parte realiza las operaciones que llevan a cabo la funcionalidad operacional de la aplicación. En esta sección explicaremos el modelo conceptual del diseño y el modelo de comportamiento.

5.3.1. Modelo conceptual

El modelo conceptual en la etapa de diseño no difiere especialmente de la versión confeccionada para la especificación. Se han añadido los valores adecuados para el funcionamiento que no se han podido tener en cuenta en la especificación.

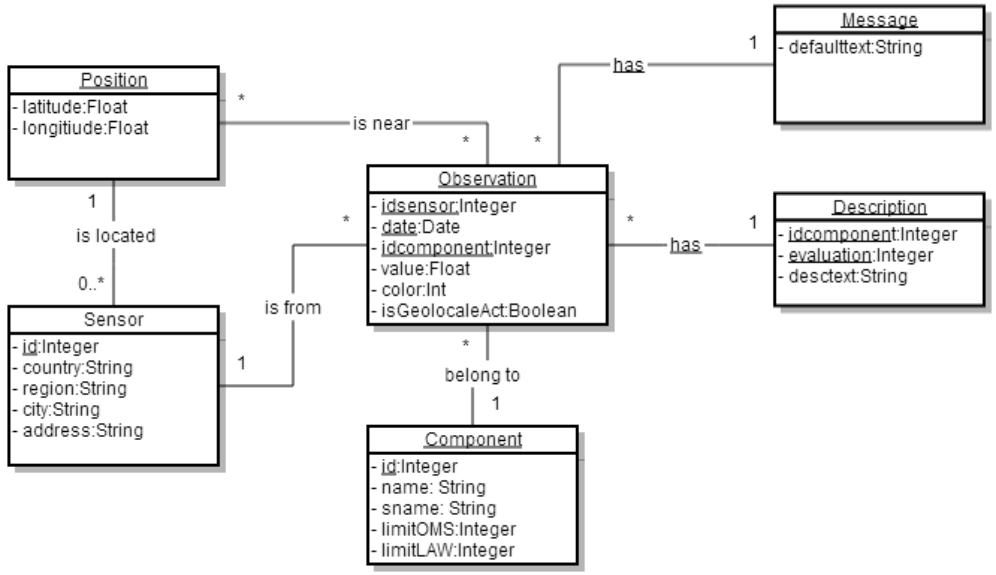


Figura 5.14: Diseño del modelo conceptual.

Cabe destacar de este modelo la inclusión del valor que permite conocer desde las observaciones si la estación se ha encontrado a través de la posición del dispositivo o de una selección natural.

Por otro lado, se han añadido algunas variables para ayudar con la consecución de la lógica y conseguir así una mayor eficiencia como por ejemplo el valor Color.

5.3.2. Modelo de comportamiento

El modelo de comportamiento conforme al diseño entra más detalle en el funcionamiento interno, no tanto en el resultado para el usuario. Para describir este modelo hemos dividido las vistas y las posibles funcionalidades, detallando primeramente la selección de zona, siguiendo con la vista de petición de las últimas observaciones, siguiendo con la vista de los detalles de una observación y por ultimo la escritura de un mensaje para compartir en las redes sociales.

La selección de zona parte de dos posibilidades iniciales, que se trate de la primera selección en cuyo caso se seleccionará la selección automática; y que ya se haya seleccionado alguna opción, en cuyo caso se mostrarán la información obtenida de la ultima elección o la configuración actual.

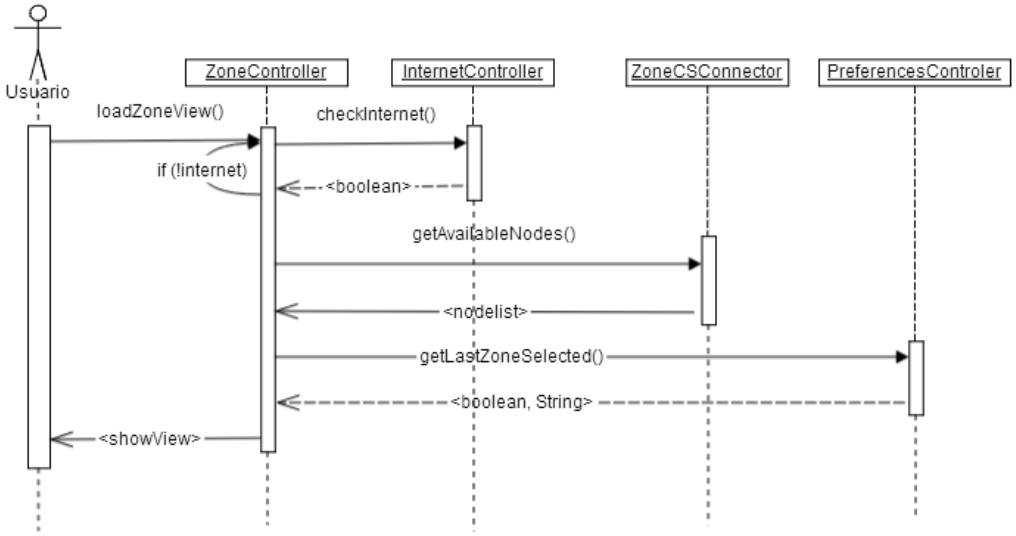


Figura 5.15: Diagrama de secuencia de selección de la ubicación.

En esta vista también tenemos la opción de cambiar la ubicación manual o marcar la opción e selección de la región mas cercana. Esta ultima es un caso más simple que el primero, por eso mostramos el diseño de la selección de una ubicación manualmente (eligiendo region, ciudad o dirección).

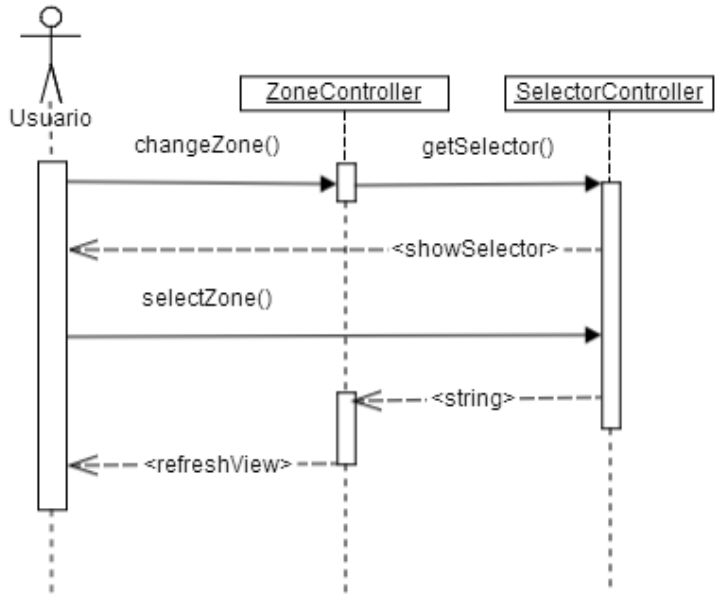


Figura 5.16: Diagrama de secuencia de cambio de ubicación.

En la siguiente vista diferenciada, la vista encargada de mostrar las últimas medidas, se realiza una comprobación inicial de la conexión a internet, posteriormente comprueba en caso de estar activada la selección automática si se encuentra encendido el servicio de ubicación del dispositivo.

El siguiente paso es identificar la estación relacionada con esa posición, se obtienen los datos de la ultima medida provenientes del servicio de CommonSense y por ultimo se evalúan estas medidas para mostrarse adecuadamente al usuario.

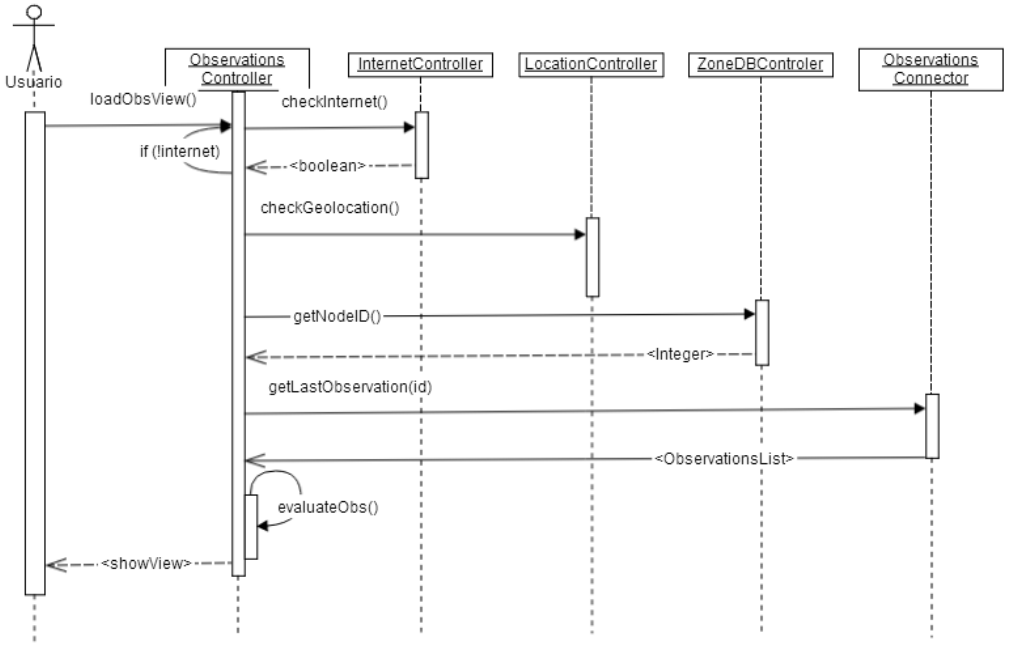


Figura 5.17: Diagrama de secuencia de carga de últimas observaciones.

Cuando se pulsa sobre una observación de estas, se carga una nueva vista con los detalles de la misma y su funcionamiento sigue la siguiente secuencia.

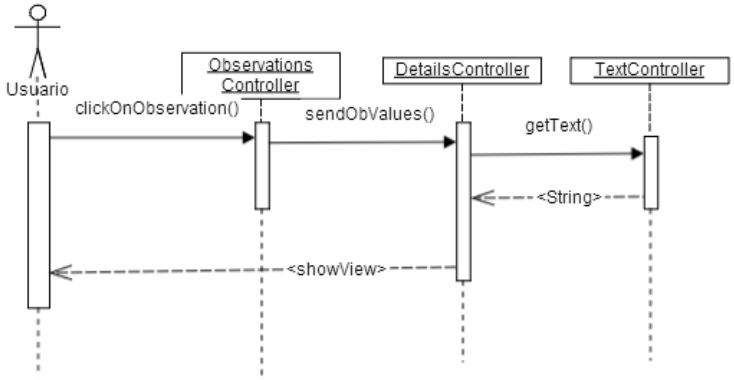


Figura 5.18: Diagrama de secuencia de carga de detalles de la observación.

Por último la funcionalidad que permite compartir o interactuar con estos datos a través de otras aplicaciones del dispositivo. Esta actividad seguiría la siguiente secuencia.

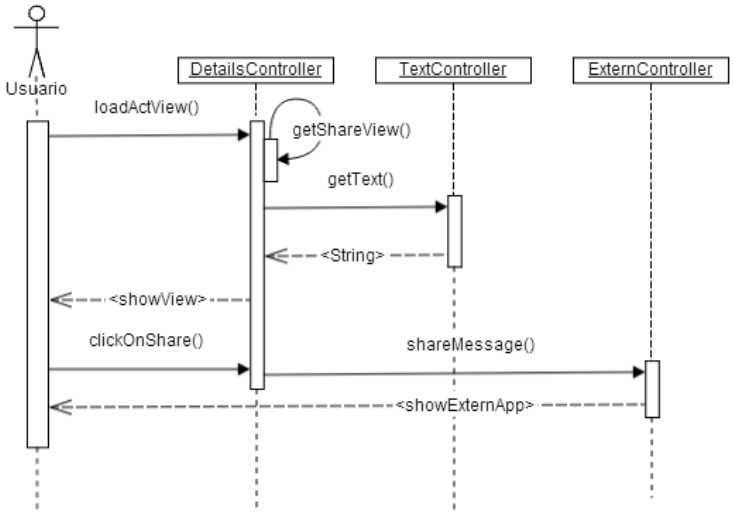


Figura 5.19: Diagrama de secuencia de actúa.

5.4. Capa de datos

La capa de datos gestiona la información de la aplicación. En AireLimpioYA los datos se obtienen del servicio que proporciona CommonSense mediante mensajes con contendido en lenguaje XML siguiendo el modelo RDF.

Estos datos son mostrados y posteriormente almacenados para proporcionar un acceso rápido a esta información sin necesidad de esperar respuesta de CommonSense, que al ser un servicio web proporciona una menor velocidad a dicha información. Los datos que recopilan la cantidad de ubicaciones con medidas se almacenan en una base de datos por la cantidad de información que representa, mientras que los datos con la observación actual se almacenan en preferencias, excepto los valores propiamente dichos de las medidas que se almacenan en BD por comodidad.

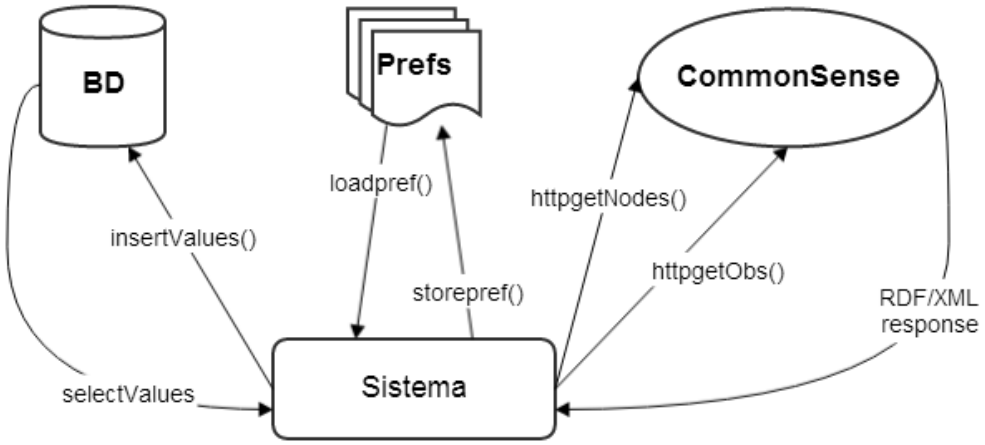


Figura 5.20: Esquema de funcionalidades de la capa de datos.

Capítulo 6

Desarrollo

Contents

Una vez obtenido el diseño inicial, procedimos con la fase de desarrollo. Según la planificación al inicio del proyecto esta se llevaría en una iteración siguiendo el modelo en cascada.

Durante el desarrollo de la estructura de la aplicación, con una semana de trabajo, nos dimos cuenta que los requisitos evolucionaban adaptándose más fielmente a las necesidades de los usuarios finales. Como consecuencia cambiamos la metodología segmentando las iteraciones de desarrollo y la flexibilidad de modificar los requisitos durante del desarrollo.

La segmentación del proyecto en iteraciones se ha llevado a cabo eligiendo funcionalidades claramente separadas para poder validad al final de cada iteración el trabajo realizado.

En esta sección se describen los aspectos mas relevantes el trabajo realizado en cada una de las iteraciones.

6.1. Estructura de la aplicación

La estructura de la aplicación es la primera iteración, y es prerrequisito de la resta de etapas. La implementación de la estructura de la aplicación es necesaria para poder validar y implementar el resto de funcionalidades y poder visualizar el efecto y los resultados de las mismas.

Esta estructura base debe tener las siguientes funcionalidades:

- El sistema debe implementar las vistas diseñadas durante la fase de diseño aunque la información que muestre no sea valida.
- El sistema debe correlación las vistas entre si, según la interacción de usuario.
- El sistema debe evitar que los errores que se puedan producir finalicen la aplicación.

6.1.1. Implementación

Al inicio de la realización partíamos de cero, no existía ningún código ni funcionalidad implementada anteriormente. Por ello se siguió el desarrollo de aplicaciones Android.

El objetivo al final de esta iteración era realizar el desarrollo visual e interactivo de las vistas en las Figuras 5.4, 5.5 y 5.6.

Al cabo de una semana de trabajo en esta etapa, el diseño visual se vio afectado por nuevos requisitos visuales y de satisfacción del cliente. A partir de ese momento se decidió la reestructuración de la planificación de la etapa de desarrollo planificando para esta etapa dos semanas.

Esta fase ha conllevado la implementación de layouts siguiendo los componentes proporcionados en la librería de oficial de Android, con algunas de las clases más utilizadas y básicas como LinearLayout, TextView e ImageView. Se descartaron las vistas Button que producían malformaciones en algunas de las versiones de sistema. Paralelamente se han utilizado iconos de la iconografía que Google pone a disposición de los desarrolladores para seguir la linea de diseño sistema operativo.

Todas las vistas siguen medias proporcionales para ser visibles en cualquier dispositivo de la misma forma independientemente de la resolución que utilicen. Esto se puede apreciar especialmente en los valores que se asocia a layout_width, layout_height y layout_weight.

Los textos estáticos se han almacenado en ficheros en formato XML para facilitar así la compatibilidad multi-idioma. También se ha utilizado XML para guardar información estática sobre estilo y dimensiones. Esto ayuda especialmente a que cualquier cambio sobre el diseño pueda ser llevado a cabo con mayor flexibilidad.

Listing 6.1: Conenido de Strings.xml

```
<resources>
    <string name="app_name">Aire Limpio YA</string>
    <string name="app_title">Aire Limpio YA!</string>

    <!-- Selector View -->
    <string name="select_advice">Seleccione</string>
    <string name="turn_back">Volver</string>
    <string name="title_region">Region</string>
    <string name="title_city">Ciudad</string>
```

```
    <string name="title_address">Direccion</string>

    <!-- Zone View -->
    <string name="button_geolocale">Siempre la mas cercana</
        string>
    <string name="button_manual">Quiero elegir una ubicacion</
        string>
    <string name="button_accept">ACEPTAR</string>
    <string name="localization">Que estacion de medida desea
        consultar?</string>
    <string name="zone_label">Seleccione ubicacion</string>
    <string name="logo_description">Logo</string>
    <string name="underlined_settings"><u>Ajustes</u></string>
    ...
</resources>
```

En tres semanas se implementó el diseño de las vistas y la navegación entre ellas con información estática (o no válida). Se actualizaron los requisitos y el diseño que no había sido alterado aun por el funcionamiento.

6.2. Obtención de información

La etapa que llevó mas complejidad en cuanto al desarrollo de la lógica. En esta fase ya teníamos una estructura inicial donde implementar funcionalidades con la ventaja poder ser probadas y validadas al finalizar la iteración.

Con esta iteración se debía implementar la obtención de información externa de medidas de contaminación del aire proporcionadas por CommonSense y mostrarla en la estructura inicial.

Esta obtención de datos debe proporcionar las siguientes funcionalidades:

- El sistema debe implementar la conexión con CommonSense siguiendo la API para desarrolladores.
- El sistema debe procesar la respuesta de CommonSense para interpretar la información necesaria.
- El sistema debe representar esa información ajustándola a la estructura base.
- El sistema debe conocer si tiene los recursos necesarios para poder hacer las peticiones.

- El sistema debe evitar que se produzcan errores y mantener la integridad del sistema.

6.2.1. Implementación

El objetivo al final de esta iteración era obtener la información deseada para mostrar al usuario y con ello tener gran parte de la funcionalidad acordada durante el alcance del proyecto.

Esta fase ha conllevado la implementación de una tarea asíncrona siguiendo el patrón Composite y extendiendo la clase AsyncTask.

En esta AsyncTask se realiza una petición HTTP siguiendo las características de la API de CommonSense y la respuesta es tratada por un ResponseHandler extendido para resolver correctamente la respuesta obtenida en formato XML con modelo RDF.

Listing 6.2: Gestión de la petición http

```
generatedURL = String.format(URL, sensorID);

HttpGet request = new HttpGet(generatedURL);
request.addHeader("Content-Type", "Application/rdf+xml");

RDFObservationsHandler responseHandler = new
    RDFObservationsHandler();

return mClient.execute(request, responseHandler);
```

Durante la lectura de la respuesta XML se capturan únicamente los datos que son de interés segun los requisitos para el desarrollo de la funcionalidad.

Listing 6.3: Etiquetas RDF/XML filtradas

```
String OBSERVATION_TAG = "ssn:Observation";
String TIME_TAG = "ssn:observationResultTime";
String OBSERVATION_PROPERTY_TAG = "ssn:observedProperty";
String VALUE_TAG = "DUL:hasDataValue";
String PROPERTY_TAG = "ssn:Property";
String PROPERTY_LABEL = "rdfs:label";
String PROPERTY_DETAIL = "rdfs:comment";
```

Al final de esta tarea se obtiene una lista de sensores o observaciones (dependiendo de lo demandado) y se envía al controlador principal para representar estos datos en la estructura base.

Durante esta fase se añadieron requisitos y modificaciones en el diseño. Pri-

meramente, la cantidad de estaciones de medida era tal que resultaba engorroso elegir de entre una lista. Por este motivo se confeccionó una vista para la selección de zona manual y por tanto una modificación en la estructura de la aplicación. Otro cambio que surgió durante esta fase es la necesidad de mostrar la fecha de la medida obtenida dado que la información no es perfectamente en tiempo real.

Otros detalle de implementación se encuentra en la espera necesaria de la finalización de la tarea asíncrona para presentar los datos obtenidos, esto implica pintar una primera versión y actualizarla al finalizar la tarea. O los problemas que pueden surgir ante recoger datos en coma flotante si estos no se han almacenado correctamente.

La obtención de la fecha durante el ResponseHandler no ha sido trivial, se obtiene un formato de fecha que no se ajusta a los requisitos que tiene el cliente, así que se debe convertir ese formato de fecha, el problema es que aunque las librerías de Android permiten cambiar el formato de una fecha, es necesario indicar que modelo de fecha debe convertir para poder traducirlo adecuadamente al formato que deseamos mostrar al usuario. Esto podría haber resultado mas sencillo si existiese algún conversor al que no fuese necesario indicarle el formato de fecha de origen como ya sucede en las librerias de otros lenguajes.

Listing 6.4: Conversion de la fecha leída

```
SimpleDateFormat obsdate = new SimpleDateFormat("EEE␣MMM␣dd␣yyyy
    ␣HH:mm:ss␣ZZZZ", Locale.ENGLISH);
SimpleDateFormat resdate = new SimpleDateFormat("HH:mm␣(dd/MM/
    yyyy)", Locale.ENGLISH);

Date current_date = obsdate.parse(xml.text);
```

En cuatro semanas se implementó la obtención de datos externos, tanto de sensores disponibles como de observaciones de un sensor. Se ha actualizado el diseño de las vistas siguiendo los requisitos surgidos durante esta etapa y con ello se ha llevado a cabo los cambios de implementación en las vistas. Al final de esta etapa tenemos una aplicación sin memoria que utiliza datos volátiles.

6.3. Almacenamiento de la información temporal

Durante la siguiente iteración se lleva a cabo el almacenamiento de datos para conseguir que la información sea menos volátil y la aplicación recuerde selecciones, configuraciones y aumente la velocidad de carga haciendo uso de información caché.

Este almacenamiento de información debe proporcionar las siguientes funcionalidades:

- El sistema debe almacenar la configuración seleccionada para posteriormente poder recuperar.
- El sistema debe almacenar la información de la última consulta realizada.
- El sistema debe almacenar la información de los nodos disponibles.

6.3.1. Implementación

El objetivo conseguido al final de esta iteración es retener la información obtenida para agilizar tanto la interacción por parte del usuario como mejorar el rendimiento en cuanto a la primera petición realizada.

De la implementación de esta iteración es destacable el uso de la librería SharedPreferences de Android que permite una mayor agilidad a la hora de almacenar datos de ajustes en la configuración. Esto ha permitido guardar datos sobre la última configuración válida.

Listing 6.5: Gestión de los datos de estado

```
SharedPreferences settings = getSharedPreferences(AdjustSet.
    stFileName, 0);
SharedPreferences.Editor editor = settings.edit();
editor.putBoolean(AdjustSet.stFirstRun, isFirstRun);
editor.putBoolean(AdjustSet.stGeoactive, geolocationSelected);
editor.putString(AdjustSet.stCRegion, regionName);
editor.putString(AdjustSet.stCCity, cityName);
editor.putString(AdjustSet.stCAddress, addressName);
editor.commit();
```

Por otro lado se ha utilizado la librería SQLite para almacenar datos en forma de base de datos relacional para información recibida sobre estaciones disponibles y observación actual. Se ha elegido este tipo de almacenamiento

por la gran cantidad de sensores disponibles y el tipo de uso que se lleva posteriormente, donde se selecciona la información por características.

Listing 6.6: Comando SQL de creación de la tabla de Sensores

```
final private static String CREATE_SENSORS_CMD =

"CREATE TABLE " + NODES_TABLE_NAME + " ( " + ID + " INTEGER
    PRIMARY KEY, " + COUNTRY_NAME + " TEXT NOT NULL, " +
    REGION_NAME + " TEXT NOT NULL, " + CITY_NAME + " TEXT NOT
    NULL, " + ADDRESS_NAME + " TEXT NOT NULL, " + LATITUDE_NUMBER
     + " REAL NOT NULL, " + LONGITUDE_NUMBER + " REAL NOT NULL) ";
```

Esta implementación se llevó a cabo en una semana, al final de esta iteración la aplicación tiene "memoria"sobre la ultima petición realizada y recupera la información con rapidez y la seguridad de disponer de los datos sobre las estaciones. Respecto a los cambios imprevistos en esta iteración no ha surgido ninguno.

6.4. Más detalles sobre la observación

Durante esta iteración se lleva a cabo la implementación de la vista que detalla los valores de una observación teniendo en cuenta su evaluación. Estos detalles deben ser implementados con las siguientes funcionalidades:

- El sistema debe recuperar la información de la observación seleccionada.
- La información debe ser mostrada de forma detallada.
- Todas las funcionalidades deben ser identificadas facilmente.

6.4.1. Implementación

El objetivo conseguido al final de esta iteración es mostrar la información de forma acorde con los intereses del usuario final teniendo en cuenta la evaluación de la medida, si es un valor saludable, poco recomendable o nocivo para la salud. Esta información es una plantilla de tres textos distintos en los que se ayuda al usuario a entender la situación y le proporciona conocimientos para conocer como se puede proteger o como le puede afectar.

Por otro lado, el objetivo es dar pie a la funcionalidad de mostrar un mensaje relacionado con el estado de la medida preparado para compartir. Esta vista se mostrará cuando se pulse el accionador correspondiente.

El diseño de las vistas se ha visto afectado para adaptarse a los requisitos del usuario, se ha cambiado la forma en que se muestran las medidas y en esta ventana el accionador se ha cambiado de posición.

Esta etapa se ha llevado a cabo en menos de una semana, la mayor complejidad la hemos encontrado en seleccionar los textos adecuados y en la adaptación de las vistas a los nuevos requisitos del usuario final.

6.5. Geolocalización

Con la geolocalización, una funcionalidad que permite conocer la ubicación de un dispositivo a través de sus servicios de localización (Wi-Fi o Internet Móvil), se consigue poder desarrollar la funcionalidad que permite elegir automáticamente la estación más cercana a través de una lista de ubicaciones disponibles.

Esta etapa de desarrollo de la geolocalización requiere implementar las siguientes funcionalidades:

- El sistema debe recuperar la información de la posición actual.
- El sistema debe comparar la información de su posición con la información de la posición de los sensores.
- No se debe tener en cuenta las estaciones que se sitúen a gran distancia.
- Si la información de la posición actual cambia se debe refrescar la busqueda de un nuevo sensor.

6.5.1. Implementación

El objetivo conseguido al final de esta iteración es conseguir que el sistema sea capaz de autoseleccionar una estación dependiendo de la latitud y longitud de la posición actual con el criterio de escoger la mas cercana.

Listing 6.7: Codigo que recupera las estaciones mas cercanas a la latitud-longitud actual

```
public StationNode getNearestNode(double currentLatitude, double
    currentLongitude) throws SQLException {
 StationNode sensor = null;
 String[] FROM = { DBOpenHelper.ID, DBOpenHelper.REGION_NAME,
 DBOpenHelper.CITY_NAME, DBOpenHelper.ADDRESS_NAME,
 DBOpenHelper.LATITUDE_NUMBER, DBOpenHelper.LONGITUDE_NUMBER };
```

```
 String ORDER_BY = "(" + DBOpenHelper.LATITUDE_NUMBER + " - (" +
      currentLatitude + "))*(" + DBOpenHelper.LATITUDE_NUMBER + "
     - (" + currentLatitude + ")) + (" + DBOpenHelper.
     LONGITUDE_NUMBER + " - (" + currentLongitude + "))*(" +
     DBOpenHelper.LONGITUDE_NUMBER + " - (" + currentLongitude +
     "))";
 Cursor result = db.query(false, DBOpenHelper.NODES_TABLE_NAME,
     FROM, null, null, null, null, ORDER_BY, null);
 if (result.moveToFirst()) {
         sensor = StationNode.create(result.getInt(0), result.
             getString(1), result.getString(2), result.getString
             (3), Double.toString(result.getDouble(4)), Double.
             toString(result.getDouble(5)));
 }
 result.close();
 return sensor;
}
```

Otra parte importante del desarrollo es la petición para conocer la posición actual. Esta información viene dada por el servicio Android.Location, el código principal únicamente debe iniciar la escucha bajo un cierto criterio y a partir de ese momento empieza a recibir la información de la posición. Los criterios pueden ser el tiempo de refresco o la distancia minima a tener en cuenta en caso de cambio.

Listing 6.8: Código de escucha al servicio de localización

```
locationManager.requestLocationUpdates( LocationManager.
    NETWORK_PROVIDER, MINIMUM_TIME, MINIMUM_DISTANCE,
    locationListener );
locationManager.requestLocationUpdates( LocationManager.
    GPS_PROVIDER, MINIMUM_TIME, MINIMUM_DISTANCE,
    locationListener );
```

Por último el desarrollo debe comprobar que la distancia entre el punto actual y la estación no sea mayor a un MAX_DISTANCE. Esto se puede conseguir con el uso del metodo Location1.distanceTo(Location2).

El desarrollo de esta iteración se ha realizado en una semana y no ha surgido ningun cambio en el diseño de las vistas ni los requisitos.

6.6. Integración con redes sociales

La última iteración de este proyecto supone la integración de la aplicación con las redes sociales, para así poder difuncir la información obtenida en esta información a los amigos conectados a las redes sociales.

En esta etapa de desarrollo es necesario implementar las siguientes funcionalidades:

- El sistema poder enviar un mensaje a la aplicación de Facebook o a la de Twitter.
- El mensaje no puede superar los 140 caracteres que permite como máximo Twitter.
- El sistema debe disponer de opciones para navegar a una pagina web a traves de la app de navegacion.

6.6.1. Implementación

El objetivo logrado al final de esta iteración es la posibilidad de enviar el mensaje creado en la etapa de mas información sobre obseravaciones en alguna de las dos redes sociales más utilizadas.

Para implementar esta funcionalidad se ha hecho uso de las llamadas entre aplicaciones o Intents, utilizando el método setText con el contenido del mensaje a enviar de forma que el usuario únicamente tiene que aceptar el envío del mensaje.

Listing 6.9: Código de envío de mensaje a otra App

```
share.putExtra(Intent.EXTRA_SUBJECT, getResources().getString(R.
    string.app_name));
share.putExtra(Intent.EXTRA_TEXT, ui_message.getText().toString
    ());
startActivity(Intent.createChooser(share, "Share It!"));
```

El diseño de la vista ha cambiado por los requisitos incrementados por el cliente, tanto la iconografía, como las funcionalidades ha ido cambiandose dejando ya los requisitos y diseño final.

Esta iteración se ha llevado a cabo en una semana con todas las funcionalidades, aunque la integración con las redes sociales no ha quedado completa al no utilizar las librerias proporcionadas por Facebook y Twitter.

Capítulo 7

Trabajo futuro

Una vez finalizada la etapa de desarrollo, queda pendiente el lanzamiento o puesta en marcha de la aplicación en un entorno controlado para comprobar que el usuario final esté satisfecho con la aplicación. Esta etapa se llevará a cabo durante las próximas semanas gracias a la colaboración de un colectivo de ecologistas voluntario que pondrá a prueba y dará su opinión critica sobre la aplicación.

Una vez esta etapa haya finalizado, incluido el trabajo de acondicionamiento de la aplicación, se debe proceder al lanzamiento público de la aplicación en el mercado de aplicaciones de Android (o Google Play).

Hacer pública la aplicación no significa el fin del proyecto. Probablemente los requisitos irán evolucionando en función de las necesidades de los usuarios como en la mayoría de proyectos software.

En ese aspecto, cabe esperar la necesidad de realizar un mantenimiento regular que permita reacondicionar el diseño de la aplicación al gusto del público en general. De esta forma se mantendrá como una herramienta viva y útil para los usuarios el mayor tiempo posible.

Por otro lado, los objetivos que planteamos inicialmente en este proyecto tienen una gran envergadura. Y aunque el alcance de este proyecto ha conseguido cumplir en gran medida con ellos, aún tenemos mucho recorrido en cuanto ampliaciones y mejoras posibles. Algunos ejemplos son las siguientes funcionalidades:

- **Ofrecer la información de más regiones.** CommonSense no dispone de más información que de Catalunya y Madrid. Un trabajo de futuro podría ser obtener más fuentes que permitan conocer el estado de calidad del aire en más lugares.

- **Desarrollar la aplicación para más dispositivos.** Se ha desarrollado una aplicación para dispositivos Android, podría ser interesante plantear un desarrollo para dispositivos iOS.
  Si los recursos en el futuro y el desarrollo de las tecnologías lo permiten, se debería plantear el desarrollo de una aplicación web con compatibilidad para cualquier tipo de dispositivos (con diversos sistemas operativos y resoluciones).

- **Mostrar más contaminantes.** Actualmente ya se proporciona y se muestra información sobre O_3, NO_2 y PM_{10}. Gracias a las fuentes de CommonSense, se consigue obtener información sobre otros contaminantes como NO, CO, SO_2, entre otros. El trabajo futuro en este ámbito sería mostrar y difundir información sobre otros contaminantes.

- **Proporcionar más herramientas para la difusión social.** En este proyecto nos hemos centrado en Facebook y Twitter por ser las más famosas, pero no podemos olvidar que existen otras redes sociales disponibles (como Google+, WhatsApp...) y otras formas de comunicación (como email, SMS...) con las cuales podríamos ampliar el alcance de la difusión.

- **Traducir la aplicación a más idiomas.** La aplicación se ha realizado en castellano, pero se ha proporcionando facilidades que permiten la inclusión de los textos en otros idiomas y así la aplicación podría proporcionas más lenguajes sin tener que realizar cambios en la implementación actual. Una tarea para el futuro podría ser traducir los textos de la aplicación.

Capítulo 8

Conclusiones

Antes de iniciar este proyecto, el objetivo primordial era conseguir reducir de forma notable la contaminación atmosférica, aunque este objetivo pudiera parecer imposible a través del desarrollo de un trabajo de fin de grado en la especialidad de ingeniería informática.

Entonces, pasamos a analizar las causas por las cuales se producía el fenómeno de contaminación atmosférica. Una de ellas nos llamó especialmente la atención *El problema de la contaminación se ignora en gran medida por el desconocimiento de la existencia de los controles y las regulaciones que se realizan sobre la calidad del aire*. En conclusión, si el público desconoce que hay un problema, nadie toma medidas contra la contaminación.

A partir de entonces focalizamos este proyecto en la resolución del problema de difusión. Mantuvimos contacto con especialistas de la disciplina del estudio atmosférico que nos ayudaron a conocer los requisitos, y nos transmitieron conocimientos de como se debía realizar de una buena difusión. Como, por ejemplo, el contenido **qué** se debe transmitir, y **cómo** debe hacerse.

En ese momento comenzó el proyecto AireLimpioYa, cuando se llevó a cabo la identificación del problema y la definición de los objetivos. Después de ello, analizamos todas las tecnologías y servicios existentes que intentaban dar solución a este problema, conociendo así las partes positivas y las negativas que estas tecnologías tenían.

En mi opinión personal, con el análisis de las tecnologías existentes he aprendido que un proyecto nunca empieza de cero, que en la mayoría de ocasiones nos basamos en el trabajo que otras personas han realizado anteriormente para, posteriormente, aportar algo más. En especial, me ha sorprendido la aparición de nuevas aplicaciones similares mientras desarrollábamos este proyecto, lo que me hace ver la riqueza en cuanto a ideas y formas de resolver un mismo problema.

Otra parte que debo destacar de la realización de este proyecto ha sido el aprendizaje durante la confección de la planificación y la importancia de seguir la metodología adecuada a las necesidades del proyecto. Por falta de experiencia, inicialmente las previsiones no fueron ajustadas debidamente, pero a lo largo del proyecto, gracias a la experiencia, este aspecto se ha visto mejorado.

Con el presupuesto se ha identificado el coste empleado aproximado y la viabilidad del proyecto. Estoy seguro que existen muchos factores alrededor de este coste que se podrían que contemplar no tanto en el presupuesto sino al coste final de realización del producto.

La sostenibilidad y el impacto social es algo que siempre ha estado presente durante todo el proyecto, pues el objetivo es concienciar a los usuarios de la importancia de cuidar del medio ambiente. Es difícil conocer el impacto que tendrá este proyecto en el ámbito social. Tengo la sensación que solo ese apartado podría suponer meses de trabajo.

En cuanto al desarrollo, ha sido un placer hacer uso del proyecto libre CommonSense, un proyecto desarrollado en la Universitat Politècnica de Catalunya, especialmente por su forma de entender la información como algo comunitario. Asimismo ha sido especialmente interesante crear una herramienta que ofreciera una solución a un problema común como es el problema de la polución.

Por último, estoy satisfecho de que este proyecto aún tiene mucho recorrido. Aún es posible ampliar sus funcionalidades y perfeccionar sus requisitos con el objetivo de ayudar primeramente con la difusión de la información de la calidad del aire. En definitiva, para que con la concienciación ciudadana se esté ayudando a la reducción de la emisión de gases contaminantes. Solo el tiempo dirá el impacto que este proyecto ha tenido sobre la sociedad.

Apéndice I

Glosario

- **Android:** es un sistema operativo para dispositivos móviles creado por Google.

- **Cliente:** Hay dos aceptaciones a la cual nos podemos referir en este proyecto para este termino: 1. es el usuario que utilizará la aplicación o 2. es una aplicación que consume un servicio remoto.

- **CommonSense:** es una plataforma de comunidad de sensores desarrollada por el grupo Compnet del departamento de Arquitectura de computadores. Su objetivo es recopilar y ofrecer la información proveniente de sensores que cualquier usuario o comunidad de usuarios desee centralizar.

- **Comunidad:** es un grupo de usuarios que mantienen algo en común, en el caso de este proyecto, una afición o una red de sensores.

- **Concentración de O_3:** Es el nivel de O_3 en el aire, habitualmente medido como .

- **Contaminación atmosférica:** Es la presencia en el aire de materias o formas de energía que impliquen riesgo, daño o molestia grave para las personas y bienes de cualquier naturaleza.

- **Desarrollo/Modelo en Cascada:** es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

- **Desarrollo iterativo y creativo:** este modelo de desarrollo de software es un conjunto de tareas agrupadas en pequeñas etapas repetitivas.

- **EEA - European Environment Agency:** O también llamada Agencia Europea de Medio Ambiente, es un órgano descentralizado -agencia- de la Unión Europea (UE), cuya misión es la recogida, elaboración y difusión de información sobre la situación y la evolución del medio ambiente a escala europea.

- **Fuente:** es un medio de difusión de información, normalmente disponible a través de internet.

- **Funcionalidad:** es un conjunto de características que hacen que algo sea práctico y utilitario.

- **Estación:** elemento de alta precisión que mide el nivel de calidad del aire. Normalmente es un elemento de la red de control atmosférico del estado español por su alto coste.
- **Google Play:** es una plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android.
- **iOS:** es un sistema operativo para dispositivos móviles creado por Google.
- **Medida:** o medición, es un proceso básico de la ciencia que consiste en comparar un patrón seleccionado con el objeto o fenómeno cuya magnitud física se desea medir para ver cuántas veces el patrón está contenido en esa magnitud.
- **Nodo:** es un punto de intersección o unión de varios elementos que confluyen en el mismo lugar. Por ejemplo: en una red de ordenadores cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.
- **O_3, NO_2, PM_{10}, SO_2, CO, etc:** denominación técnica de algunos de los elementos que se pueden encontrar en el aire: Ozono, Dióxido de nitrógeno, Micropartículas cuyo diámetro es menor a 10 micrómetros, Dióxido de azufre, Monóxido de Carbono.
- **Observación:** en este contexto es sinónimo de medida / medición.
- **OMS - Organización Mundial de la Salud:** es el organismo de la Organización de las Naciones Unidas (ONU) especializado en gestionar políticas de prevención, promoción e intervención en salud a nivel mundial.
- **Plataforma:** es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible.
- **Proveedor:** es una entidad que presta servicios a otras entidades.
- **Red Social:** es un medio de comunicación social que se centra en encontrar gente para relacionarse en línea. Normalmente nos referimos a ella con Twitter o Facebook.
- **REST:** es una técnica de arquitectura software para sistemas hypermedia distribuidos como la World Wide Web, normalmente se usa para describir cualquier interfaz web simple que utiliza XML y HTTP.

- **Sensor:** es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. En este proyecto nos referiremos a sensores de calidad del aire.
- **Servicio web:** es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
- **Servidor:** es un nodo que, formando parte de una red, provee servicios a otros nodos denominados clientes.
- **Smartphone:** es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades semejantes a una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional.
- **Storyboard:** es un conjunto de ilustraciones mostradas en secuencia con el objetivo de servir de guía para entender una historia.
- **URL:** es un identificador de recursos uniforme (URI) cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo.
- **Usuario final:** es la persona o personas que van a manipular de manera directa un producto de software.
- **Vista:** es la parte visible de una aplicación.

Bibliografía

[1] Ecologistas en Acción con la colaboración de la Fundación Biodiversidad. *Informe de la calidad del aire en 2012*. URL: https://www.ecologistasenaccion.org/IMG/pdf/informe_calidad_aire_2012.pdf.

[2] AEMET. *Agencia estatal de meteorología*. URL: http://www.aemet.es/es/portada.

[3] European Environment Agency. *Air Quality Map*. URL: http://www.eea.europa.eu/themes/air/air-quality/map/real-time-map.

[4] Generalitat de Catalunya. *Aire.CAT para Android*. URL: https://play.google.com/store/apps/details?id=cat.gencat.mobi.airecat.

[5] Generalitat de Catalunya. *Aire.CAT para iOS*. URL: https://itunes.apple.com/us/app/aire.cat/id877506570?mt=8.

[6] Generalitat de Catalunya. *Xarxa de vigilancia i previsió de la contaminació atmosferica*. URL: http://www.gencat.net:8000/oicqa/owa/b01.consulta?estacio=00&contaminant=99&dades=1.

[7] Grupo Compnet del Departament Arquitectura de Computadors. *Comm onsense - Una red de sensores comunitaria*. URL: http://commonsense.pc.ac.upc.edu/.

[8] Interaction Designers. *Android Patterns*. URL: http://www.androidpatterns.com/.

[9] Google Developers. *Android Patterns*. URL: http://developer.android.com/design/patterns/index.html.

[10] Hazloposible.org. *Microdonaciones para AireLimpioYa*. URL: http://microdonaciones.hazloposible.org/proyectos/detalle/?idProyecto=112.

[11] LogMeIn. *Xively*. URL: https://xively.com/.

[12] World Healt Organization. *Review of evidence on health aspects of air pollution*. URL: http://www.euro.who.int/__data/assets/pdf_file/0004/193108/REVIHAAP-Final-technical-report-final-version.pdf.

[13] EUROPA PRESS. *Diferencias entre coste por habitante en los presupuestos sanitarios*. URL: http://www.infosalus.com/actualidad/noticia-diferencias-ccaa-mas-500-euros-habitante-presupuestos-sanitarios-2014-20140107133032.html.

[14] Barcelona Supercomputing Center - Centro Nacional de Supercomputación (BSC-CNS). *CALIOPE - Sistema de pronóstico de la CALIdad del aire Operacional*. URL: http://www.bsc.es/caliope/es.

[15] Carrio ts. *Internet of Things Platform*. URL: https://www.carriots.com/.