

Escuela de Ingeniería Informática

Aplicación Web de Información Pluviométrica

Crístofer Martel Miranda
Las Palmas de Gran Canaria



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática



Proyecto fin de carrera de Ingeniería en Informática de la
Universidad de Las Palmas de Gran Canaria presentado por el alumno:

Crístofer Martel Miranda

Título del Proyecto:

Aplicación web de información pluviométrica

Tutores:

Francisca Quintana Domínguez

Carlos Hernández Medina

A mi familia y amigos.

AGRADECIMIENTOS

Me gustaría agradecer en primer lugar a mis tutores por toda la ayuda que me han ofrecido dándome su apoyo y consejos para llevar este proyecto a cabo.

A mi familia por estar encima de mí con sus ánimos para poder finalizar esta última etapa en la universidad.

Y a todos mis amigos que tantos buenos momentos hemos pasado a lo largo de toda la carrera.

Muchas gracias a todos.

Crístofer.

Índice

1. Introducción	2
2. Objetivos	2
3. Estado del arte.....	3
3.1. Consejos insulares	3
3.1.1. <i>Situación actual</i>	6
3.1.2. <i>Resumen</i>	9
3.2. Aplicaciones de ámbito nacional.....	10
3.2.1. <i>Pluviómetro de Andalucía</i>	10
3.2.2. <i>SAIH</i>	16
3.2.3. <i>AEMET</i>	20
3.3. Aplicaciones de ámbito internacional	25
3.3.1. <i>Servicio de meteorología Irlandés</i>	25
3.3.2. <i>Rain rauge</i>	29
3.4. Resumen.....	34
4. Metodología	36
5. Recursos Utilizados	40
5.1. Recursos Hardware.....	40
5.2. Recursos Software	42
5.2.1. <i>Herramientas</i>	44
5.2.2. <i>Tecnologías</i>	46
6. Análisis	50
6.1. Requisitos de la aplicación Web.....	50
6.1.1. <i>Requisitos del software</i>	50
6.2. Modelo de Interacción.....	71
6.2.1. <i>Interfaz para dispositivos móviles</i>	73
6.2.2. <i>Interfaz para tabletas</i>	76
6.2.3. <i>Interfaz para escritorio</i>	81

6.3. Requisitos no funcionales	85
6.4. Modelo de dominio	87
7. Diseño.....	88
7.1. Diagrama de despliegue	88
7.2. Patrón de diseño – Modelo Vista Controlador (MVC)	89
7.3. Ruby on Rails	90
7.4. Diagramas de clases de diseño	93
7.5. Modelo de datos.....	99
8. Desarrollo	100
8.1. Estructura de ficheros del proyecto Rails.....	100
8.2. Recursos externos	104
8.2.1. <i>Cliente – Front-End</i>	104
8.2.2. <i>Servidor – Back-End</i>	105
8.3. Despliegue en Heroku.....	107
9. Pruebas.....	114
10. Resultados y conclusiones	116
11. Trabajo Futuro	117
12. Bibliografía.....	118
Anexo I: Manual de uso de la aplicación.....	120
Anexo II: Detalles sobre la creación del mapa del proyecto.....	127

1. Introducción

El Consejo Insular de Aguas es una institución perteneciente al Cabildo de Gran Canaria y es el encargado entre otras cosas, de la medición y recogida de datos de las lluvias de la isla de Gran Canaria.

Para ello, hay una serie de dispositivos repartidos por diferentes puntos de la isla que son usados para llevar el control de las infraestructuras como pueden ser presas, pozos, estaciones meteorológicas o pluviómetros, que devuelven una serie de parámetros que son usados por el consejo para realizar una serie de estudios e investigaciones.

El objetivo de este proyecto es usar la información recogida de los pluviómetros para realizar una aplicación web adaptativa a diferentes dispositivos móviles que permita acceder y entender estos datos de una forma sencilla y rápida para el uso público.

En este documento se presentan cada una de las fases llevadas a cabo durante el proceso de elaboración del proyecto fin de carrera, en el que se reflejan todos los aspectos relevantes del mismo y se justifican cada una de las decisiones tomadas durante su desarrollo.

2. Objetivos

El objetivo del proyecto es desarrollar una aplicación web adaptativa a diferentes dispositivos como son los móviles, tabletas y pcs de escritorio, que permita a los ciudadanos acceder a la información de los pluviómetros repartidos por toda la isla de una forma sencilla y desde cualquier dispositivo.

El motivo de que nuestra aplicación siga la filosofía del diseño web adaptable (RWD), cuyo objetivo es adaptar su apariencia al dispositivo web que la esté usando, es que el uso de dispositivos móviles, está

creciendo a un ritmo increíble, incrementando las ventas y el uso de éstos en los últimos años. Tanto es así que la navegación en Internet mediante estos dispositivos cada vez más común, superando al ordenador de sobremesa.

Desde todos los tamaños se podrá acceder a la información de los pluviómetros clasificados por municipios y estaciones, visualizándola de diferentes formas e intervalos de tiempo hasta 30 días de antigüedad.

Solo la versión de escritorio poseerá un buscador, con diferentes parámetros para encontrar información específica de toda la base de datos de precipitaciones. Esto es, debido al espacio del que podemos hacer uso, ya que será más cómodo para el usuario.

3. Estado del arte

En este apartado se presenta un estudio del estado actual de las aplicaciones pluviométricas, comenzando por el estado que presentan hoy en día los consejos insulares de aguas, para luego pasar a las aplicaciones que cubren el ámbito nacional e internacional.

3.1. Consejos insulares

Los Consejos Insulares de Aguas [\[1\]](#) se crearon como consecuencia de las disposiciones de la Ley 12/1990, de 9 de julio, de Aguas, asumiendo el control y la administración de forma autónoma así como la participación en la dirección, ordenación, planificación y gestión unitaria de las aguas de las Islas Canarias con personalidad jurídica propia y plena autonomía.

Esto es debido a que el agua es un recurso básico que cobra una mayor importancia en nuestro ámbito insular, donde el territorio es limitado y la presencia de lluvias escasa.

Cada isla constituye una cuenca hidrográfica, con significativas diferencias entre unas y otras. Por tanto, se estableció una administración insular donde participan los sectores, públicos y privados, para que arbitren en su ordenación, aprovechamiento, uso y gestión para cada una de las siete islas que componen el archipiélago canario.

Las funciones que desarrollan son las siguientes:

1. La elaboración de su presupuesto y la administración de su patrimonio.
2. La elaboración y aprobación de ordenanzas que el desarrollo de su actividad pueda precisar.
3. La elaboración y aprobación inicial de los Planes y Actuaciones Hidrológicas.
4. El control de la ejecución del planeamiento hidrológico y, en su caso, la revisión de este.
5. El otorgamiento de las concesiones, autorizaciones, certificaciones y demás actos relativos a las aguas, así como la inspección, realización de aforos y vigilancia en las condiciones en ellas impuestas.
6. La custodia del Registro y Catálogo de Aguas Insulares y la realización de las inscripciones o rectificaciones oportunas.
7. La gestión y control del dominio público hidráulico, así como de los servicios públicos regulados en la Ley.
8. La policía de aguas y sus cauces encargados de vigilar y proteger los barrancos e instalaciones de los consejos.

9. La instrucción de todos los expedientes sancionadores y la resolución de los sustanciados por faltas leves y menos graves.
10. La ejecución de los programas de calidad de las aguas, así como su control.
11. La realización de las obras hidráulicas de responsabilidad de la Comunidad Autónoma de cada isla.
12. La fijación de los precios del agua y su transporte, en aplicación de lo que reglamentariamente establezca el Gobierno de Canarias.
13. La participación en la preparación de los planes de ordenación territorial, económicos y demás que puedan estar relacionados con las aguas de cada isla.
14. La explotación, en su caso, de aprovechamientos de aguas y realización de estudios de hidrología.
15. La prestación de toda clase de servicios técnicos relacionados con el cumplimiento de sus fines y, cuando proceda, el asesoramiento a las Administraciones Públicas, así como a los particulares.
16. Las que se deriven de los Convenios con la Comunidad Autónoma de Canarias, Cabildo Insulares, Corporaciones Locales y otras entidades públicas o privadas, o de los suscritos con los particulares.
17. En general, todas las labores relativas a la administración de las aguas insulares no reservadas a otros organismos por la Ley de Aguas o por las normas generales atributivas de competencias.

3.1.1. Situación actual

Tras estudiar cada uno de los sitios web de los consejos insulares de todas a las islas, se ha podido determinar que la situación de acceso a los datos pluviométricos está obsoleta en todos ellos, debido a que son datos que no están actualizados a día de hoy o directamente no se muestran.

Los consejos de Gran Canaria y Lanzarote son los únicos que poseen un apartado específico y de fácil acceso para los pluviómetros, donde se muestran, información general y los datos completos de lluvias entre los años 1924 y 2009.

En Gran Canaria para poder ver y manipular los datos de pluviometría, se usa un programa de escritorio (Ver Figura 3.1), que es únicamente compatible para sistemas Windows hasta la versión XP.



Figura 3.1 Vista principal del programa de pluviometría del CIAGC

En Lanzarote, se usan documentos de Excel donde puedes elegir descargar los datos de todas las estaciones en un solo documento o en distintos, divididos por estación (Ver Figura 3.2)

LISTADO DE PLUVIÓMETROS		
Código	Nombre de la estación	Descarga
001-L	Pechiguera	
002-L	Puerto Mármoles	
003-L	Guacimeta - Aeropuerto	
005-L	Haría	
006-L	Mácher	
007-L	Femés	
008-L	Guatiza	
009-L	Soo	

Figura 3.2 Vista principal del programa de pluviometría del CIALT

Por otro lado, en los consejos insulares restantes no se muestra ningún dato, como es el caso del Hierro, La Gomera y La Palma, o se dan muy pocos, como ocurre en los casos de Tenerife y Fuerteventura.

En Tenerife, en el apartado de la página web del CIATF, llamado “Tenerife y el agua”, se muestran aspectos generales relacionados con las precipitaciones en la isla, donde solo hemos podido encontrar la imagen de la Figura 3.3 en la que se puede ver datos genéricos de precipitaciones entre 1921 y 2004.

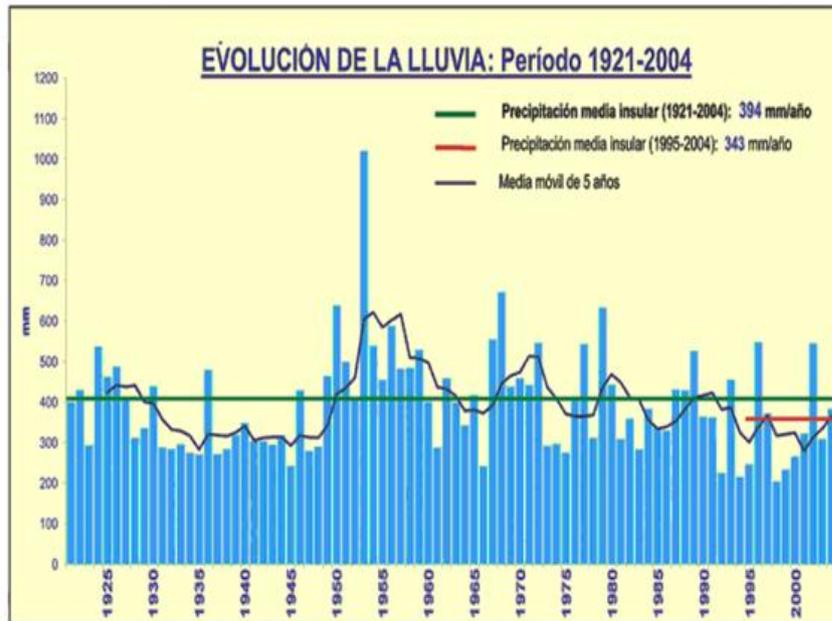


Figura 3.3 Gráfica del CIATF

En el caso de Fuerteventura, se muestran datos de unos días concretos. En la Figura 3.4 podemos ver un día al azar que hemos seleccionado, en el que se muestran el nombre de la estación, las coordenadas y la precipitación recogida.





Inicio

Descripción de la Isla
Descargas
Tablón de Anuncios
Buzón de Sugerencias
Noticias

El Consejo como Organismo

Presentación
Estatutos
Organigrama
Organos de Gobierno
Auxilios Y Subvenciones
Obras Hidráulicas

Directiva Marco de Aguas

DMA
Transposición de la DMA

Planificación Hidrológica

Plan Hidrológico
Revisión del Plan Hidrológico
Participación Pública
Demarcación Hidrográfica de Fuerteventura

Perfil del contratante

Perfil del contratante

6 de Mayo de 2013 19:48
Jornadas de Lluvia | Datos pluviométricos observados

Datos pluviométricos observados

Atendiendo a los datos de precipitación remitidos a este Organismo por los observadores de los pluviómetros que conforman la actual red pluviométrica de Fuerteventura, el pasado día 25 de Enero de 2011 se tomaron las siguientes medidas en las estaciones que se relacionan:

ESTACIÓN		U.T.M. 1998				MEDIDA DE PRECIPITACIÓN (MM)
Nº	NOMBRE	X	Y	Z	HOJA 5.000	
010	Vega del Río Palmas	589834	3141217	263	26-C	45,50
014	Gran Tarajal	595887	3121276	23	43-D	35,40
018	Agua de Bueyes	595473	3139338	248	30-B	22,50
036	Tuineje	593981	3133057	168	34-B	53,30
039	Tarajalejo	585845	3119272	72	46-B	24,80

[Volver](#)

Figura 3.4 Datos de precipitaciones de CIATF

3.1.2. Resumen

A día de hoy ninguna de las instituciones ha puesto en línea datos actualizados de las lluvias, que puedan ser accedidos por los ciudadanos u otro tipo de institución o empresa.

Con respecto a la adaptación a las nuevas tecnologías, como puede ser una versión web adaptada a dispositivos móviles o una aplicación móvil específica, ninguno de ellos tiene estas opciones disponibles.

Por tanto, se puede concluir que “Los Consejos Insulares de Aguas” no están al día, tanto en la actualización de la información de los pluviómetros como en la puesta en marcha de nuevas tecnologías.

3.2. Aplicaciones de ámbito nacional

En este apartado, se estudiarán diversas aplicaciones asociadas a la pluviometría en el ámbito nacional para poder ver cómo se encuentra la situación en nuestro país.

3.2.1. Pluviómetro de Andalucía

La junta de Andalucía ha creado una aplicación para los móviles de la plataforma Android que permite la consulta de los datos de lluvia de esa comunidad.

Desde esta aplicación se pueden consultar los registros de las diferentes estaciones meteorológicas de Andalucía. En el detalle de cada estación, el usuario dispone de información sobre la fecha y cuantía de las precipitaciones, la precipitación acumulada a lo largo del año hidrológico y gráficas de evolución.

Cuando abrimos la aplicación (Ver Figura 3.5) lo primero que vemos es una presentación con los siguientes datos:

1. Responsable encargado de la aplicación.
2. Nombre de la aplicación.
3. Subtítulo informativo.
4. Imagen de la Unión Europea de los fondos Feder.
5. Departamento responsable de la aplicación.

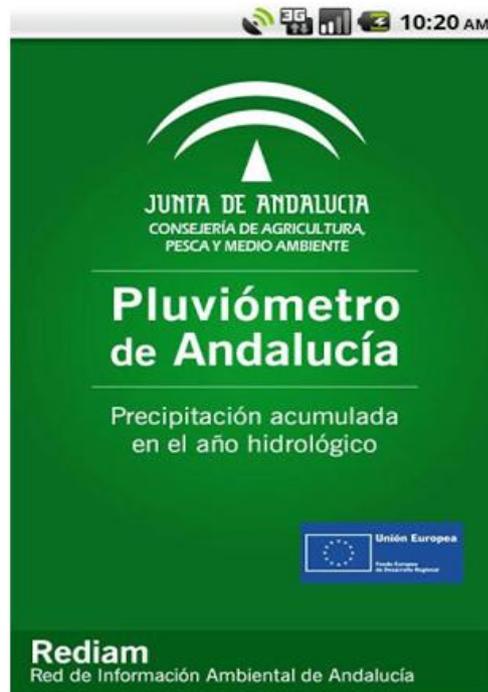


Figura 3.5 Portal del programa de Andalucía

El primer paso es seleccionar la provincia de la comunidad para la muestra de los datos (Ver Figura 3.6).



Figura 3.6 Vista intermedia del Pluviómetro de Andalucía

Una vez seleccionada, aparece una vista general (Ver Figura 3.7) que muestra una serie de datos, de los cuales los más importantes son:

1. Lugar de la estación.
2. Cantidad de la última precipitación y día.
3. Dibujo de representación de la cantidad de lluvia.
4. Precipitación acumulada.

Después de estos datos aparece una explicación que indica el porcentaje de la precipitación acumulada con respecto a la media y una gráfica comparativa de la precipitación acumulada y media.



Figura 3.7 Vista de una estación concreta

Si se quiere ver la lluvia caída en meses anteriores se selecciona el botón “Ver gráfica” y aparece una imagen con una gráfica de barras (Ver Figura 3.8) con la evolución de la lluvia caída en los meses anteriores al último dato recogido.

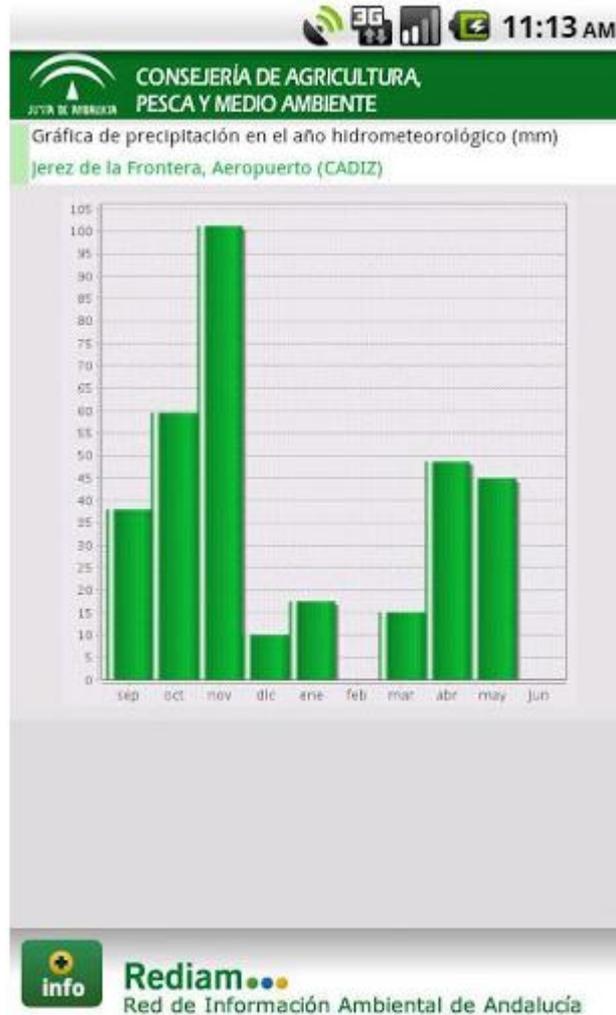


Figura 3.8 Vista Gráfica de los datos

Ventajas y desventajas

En la Tabla 3-1 se muestran las ventajas y desventajas de la aplicación “Pluviómetro de Andalucía”, la característica de geolocalización es un punto fuerte ya que te permite acceder a datos más precisos de tu zona pero se ve penalizado por la gran cantidad de desventajas como pueden ser que no sea multiplataforma obligándote a tener un móvil con el sistema operativo Android para poder acceder a la información.

Además de no tener una clasificación de las últimas provincias donde más ha llovido o un mapa donde poder ver la situación general de Andalucía, así no tener que estar comprobando provincia por provincia en cual ha llovido y en cual no.

En resumen, estos detalles hacen que la experiencia del usuario disminuya notablemente y perjudique el uso de la aplicación.

Ventajas	Desventajas
Es gratuita	No tiene la posibilidad de seleccionar estaciones favoritas
Geolocalización de la estación más cercana	No permite compartir los datos
	No es multiplataforma
	No posee un mapa o una tabla donde aparezca la situación general de Andalucía.

Tabla 3-1 Ventajas y desventajas de la aplicación del Pluviómetro de Andalucía

SAIH Ebro

De todas las confederaciones hidrográficas de la red SAIH hemos elegido la del Ebro porque es la única que además de tener versión web, posee una aplicación móvil, llamada “SAIH Ebro” [3], tanto para las plataformas de iOS como Android.

Proporciona datos sobre niveles y caudales de los ríos de la cuenca, embalses (nivel, volumen, % de llenado), estaciones meteorológicas, precipitaciones y otros usos del agua (Ver Figura 3.10).

La información se actualiza en tiempo real (datos horarios para público en general y datos cada quince minutos para usuarios registrados), se presenta sobre mapas geográficos (de toda la cuenca o de una subcuenca concreta) y en formato numérico mediante tablas. También se puede usar un buscador de estaciones para acceder de forma rápida a la información introduciendo el código o el nombre de la estación.



Figura 3.10 Portada del SAIH Ebro

La aplicación nos da la posibilidad de ver los mapas pluviométricos en diferentes intervalos de tiempo (15min, 1h, 12h, 24h) que muestran la cantidad de lluvia que ha caído de forma gráfica en el mapa. Además, también permite visualizar las estaciones que han medido más lluvia en las últimas veinticuatro horas (Ver Figura 3.11).

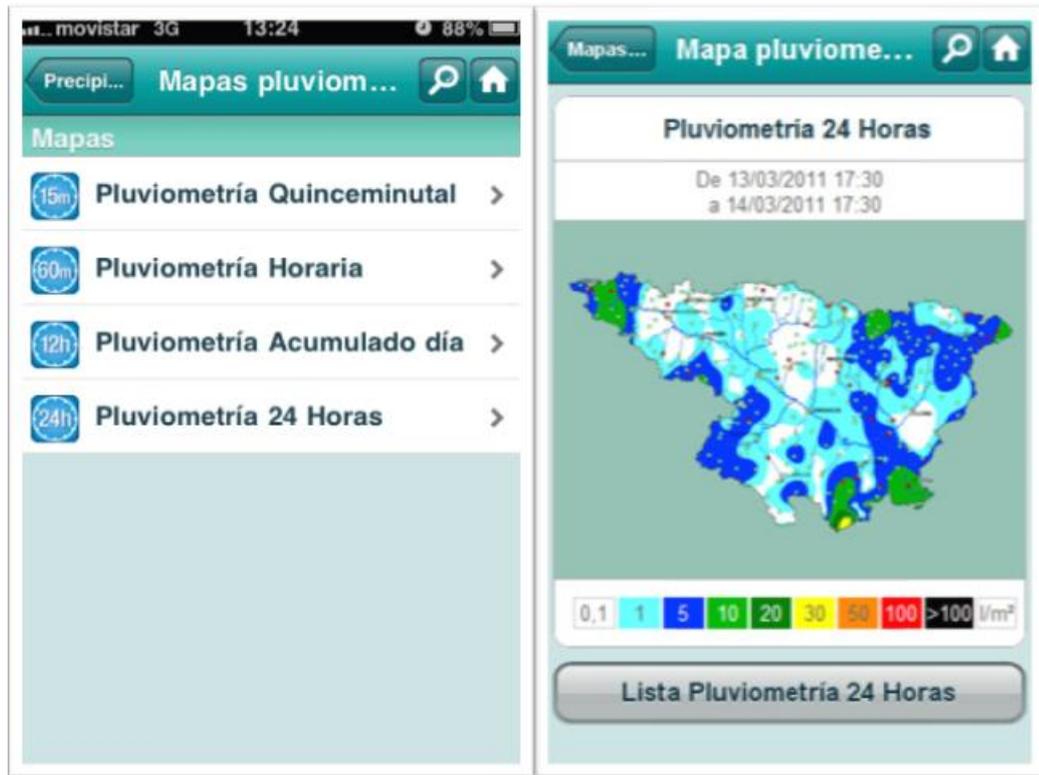


Figura 3.11 Vista intermedia de SAIH Ebro

Ventajas y desventajas

En la Tabla 3-2 se muestran las ventajas y desventajas de la aplicación SAIH Ebro, que es una de las más completas que he podido encontrar a nivel nacional ya que posee varios puntos fuertes como pueden ser que sea multiplataforma permitiendo desde diferentes puntos el acceso a los datos de hasta los últimos 15 días de precipitaciones.

Estos se muestran mediante un mapa de forma gráfica o en tablas, con un formato numérico en diferentes intervalos de tiempo o clasificados según la cantidad de precipitación. Además posee un buscador de estaciones para ir directamente a la estación deseada.

Por otro lado, tener todas esas características para ver los datos y después no poder compartirlos con otras personas o enviártelos a ti mismo para poder hacer uso de ellos en una desventaja que no conllevaría gran trabajo de implementar.

La falta de la característica de la geolocalización o poder seleccionar como favoritos las estaciones de interés en una cuenca donde la superficie es de unos 85.000 km² estaría bien para simplificar la búsqueda de los datos.

En resumen, la aplicación SAIH Ebro es una de las aplicaciones más completas a nivel nacional y con algunos cambios mínimos sería una de las mejores.

Ventajas	Desventajas
Multiplataforma	No permite compartir los datos
Posee buscador de estaciones	No permite tener estaciones favoritas
Te muestra los datos tanto de forma gráfica como de forma numérica	No permite usar geolocalización
Es gratuita	
Te permite ver datos según intervalos de tiempo	
Puedes acceder a datos antiguos	

Tabla 3-2 Ventajas y desventajas de la aplicación SAIH Ebro

3.2.3. AEMET

La Agencia Española de Meteorología [4] posee una aplicación móvil tanto en plataforma de iOS como Android, además de una web móvil adaptada para que sea visible en los demás dispositivos. Por tanto, ofrece una solución multiplataforma para acceder a sus datos.

Mediante la aplicación móvil puedes conocer las últimas predicciones y avisos oficiales en tiempo real, detectando automáticamente donde te encuentras o usando un buscador para encontrar la localidad deseada, pudiéndola seleccionar como favorita y así no tener que buscarla nuevamente (Ver Figura 3.12).

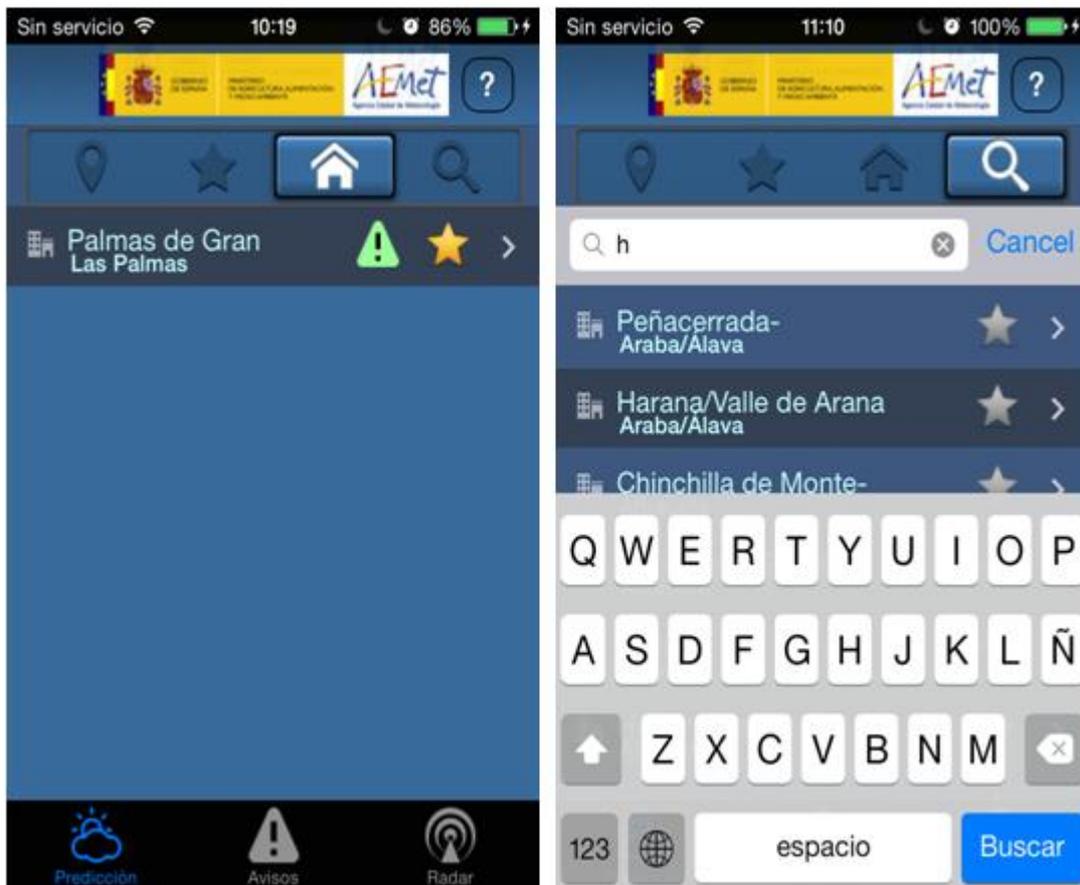


Figura 3.12 Vista de localidades favoritas y buscador de la app de AEMET

Además podrás ver imágenes del radar de diferentes puntos de España, así como lo avisos meteorológicos, en el caso de que los hubiera (Ver Figura 3.13).



Figura 3.13 Vista de zona de avisos meteorológicos y radar de la aplicación de AEMET

En la versión web normal y adaptada, es donde único se pueden acceder a datos meteorológicos de estaciones con valores en tiempo real, de las observaciones horarias de las últimas 24 horas. También se podrá acceder a los resúmenes diarios de estas estaciones en los últimos 7 días (Ver Figura 3.14).

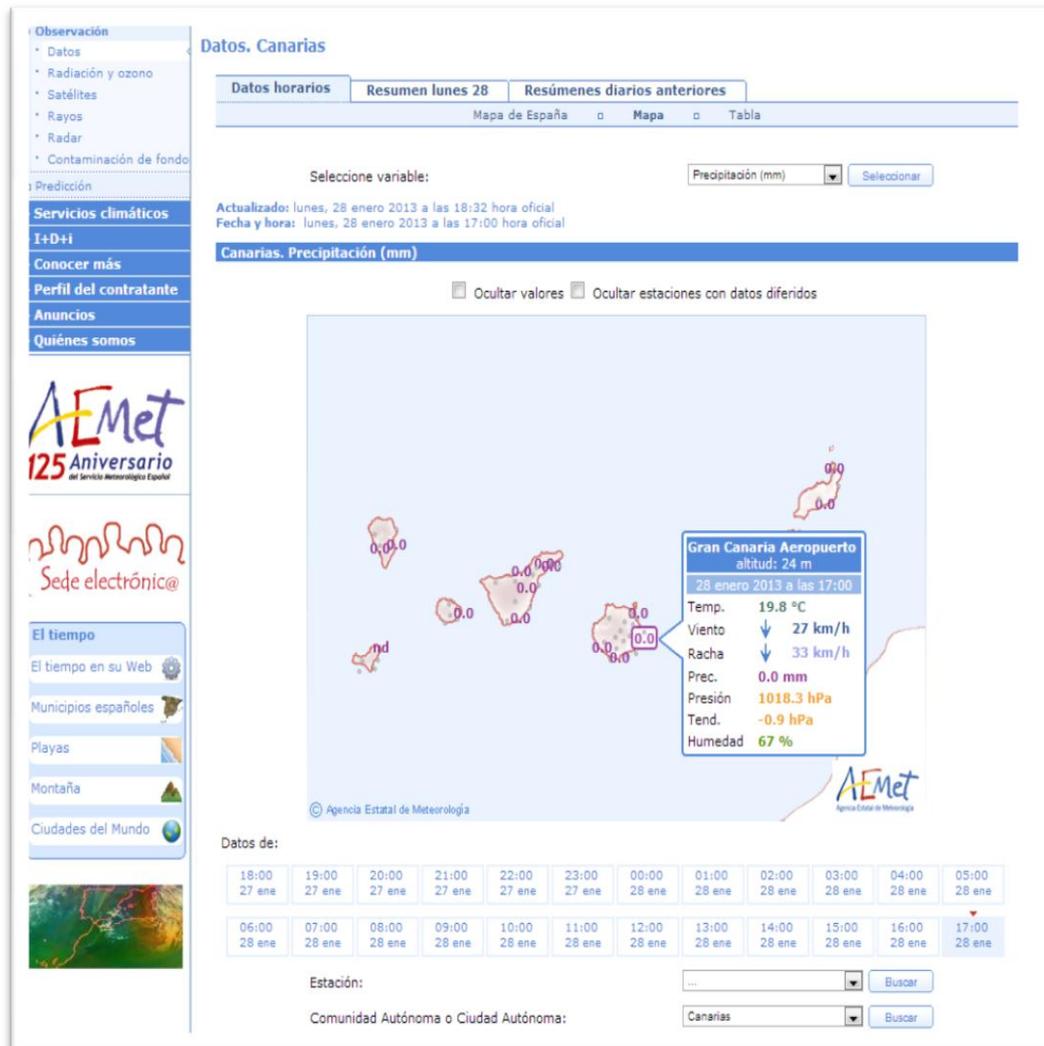


Figura 3.14 Vista de los datos de las estaciones en la web de AEMET

Cabe decir que en la versión web adaptada visualmente se pueden realizar las principales funciones, aunque los datos de las estaciones mostradas anteriormente solo se podrán ver en una tabla, pero no se visualizará correctamente (Ver Figura 3.15).

Fecha y hora oficial	Temp. (°C)	V. vien. (km/h)	Dir. viento	Racha (km/h)
19/02/2013 17:00				
19/02/2013 16:00				
19/02/2013 15:00	16.4	24	↓	40
19/02/2013 14:00	17.2	17	↙	33
19/02/2013 13:00	16.4	22	↙	43

Dir. viento	Racha (km/h)	Dir. racha	Prec. (mm)	Presión (hPa)	Tend (hPa)
↓	40	↓	0.0		
↙	33	↙	0.0		
↙	43	↙	0.0		

Figura 3.15 Vista de los datos de las estaciones en la versión web adaptada

Ventajas y desventajas

En la Tabla 3-3, mostramos los puntos fuertes y débiles de la aplicación. Como se puede apreciar predominan más en este caso las ventajas que las desventajas.

La aplicación de AEMET posee bastantes puntos fuertes como son el poder usarla los diferentes tipos de dispositivos, geolocalización y poder seleccionar tus localidades favoritas para acceder más rápidamente la información de su interés.

Por otro lado, solo podemos ver los datos del día y las probabilidades de lluvia pero no datos verdaderos como se pueden ver en la web, que te permite visualizar hasta los últimos 7 días de precipitaciones.

Además, no se puede compartir estos datos mediante las redes sociales o enviarlos por email para hacer uso de los mismos.

En resumen, la aplicación de AEMET posee las características esenciales para facilitar la interacción del usuario pero la no integración de las redes sociales y no poder acceder a datos sino a probabilidades del día la penaliza en gran medida.

Ventajas	Desventajas
Multiplataforma	No puedes compartir datos
Gratuita	No puedes acceder a datos antiguos
Posee buscador de localidades	
Puedes seleccionar localidades favoritas	
Tiene geolocalización	

Tabla 3-3 Ventajas y desventajas de la aplicación AEMET

3.3. Aplicaciones de ámbito internacional

En este apartado, estudiaremos las aplicaciones a nivel internacional para analizar cómo se encuentra la situación de la pluviometría, en Europa y en el resto del mundo.

3.3.1. Servicio de meteorología Irlandés

Met Éireann [5], el Servicio Meteorológico Nacional de Irlanda, es una división del Departamento de Medio Ambiente, Comunidad y Gobierno Local, y es el líder de información sobre el clima y los servicios relacionados con Irlanda.

Posee una aplicación móvil en las principales plataformas (como son iOS y Android). Además, posee una versión web adaptada visualmente para móviles que posean acceso a internet y no pertenezcan a los sistemas operativos móviles que hemos comentado anteriormente.

En la vista principal podemos ver de entrada la temperatura, nubosidad y viento de la ciudad que tengamos elegida, junto con la fecha y la hora de la medición, como se puede ver en la Figura 3.16.

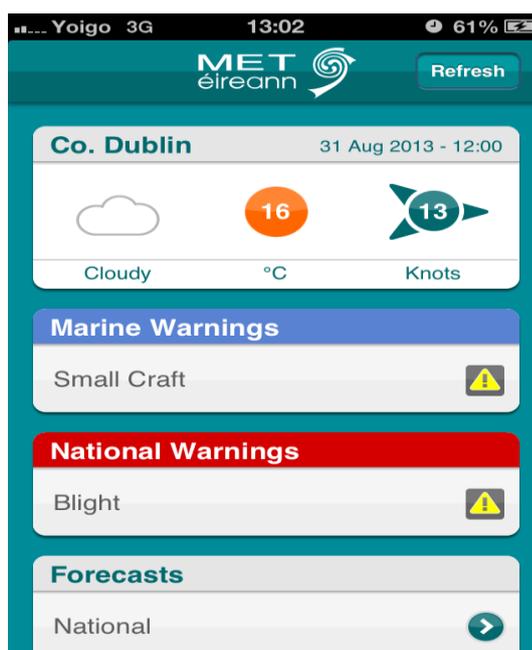


Figura 3.16 Portada de Met.ie

Entre otros muchos servicios también podemos ver los últimos informes de las estaciones repartidas por Irlanda viendo diferentes parámetros, como por ejemplo, el día y la hora de la última actualización (Ver Figura 3.17).



Figura 3.17 Vista de la estación de la aplicación MET Éireann

Si el usuario desea datos más antiguos, solo se pueden obtener mediante la versión web desde el PC, donde puedes elegir entre resúmenes de un día concreto o mensual (Ver Figura 3.18 y Figura 3.19).

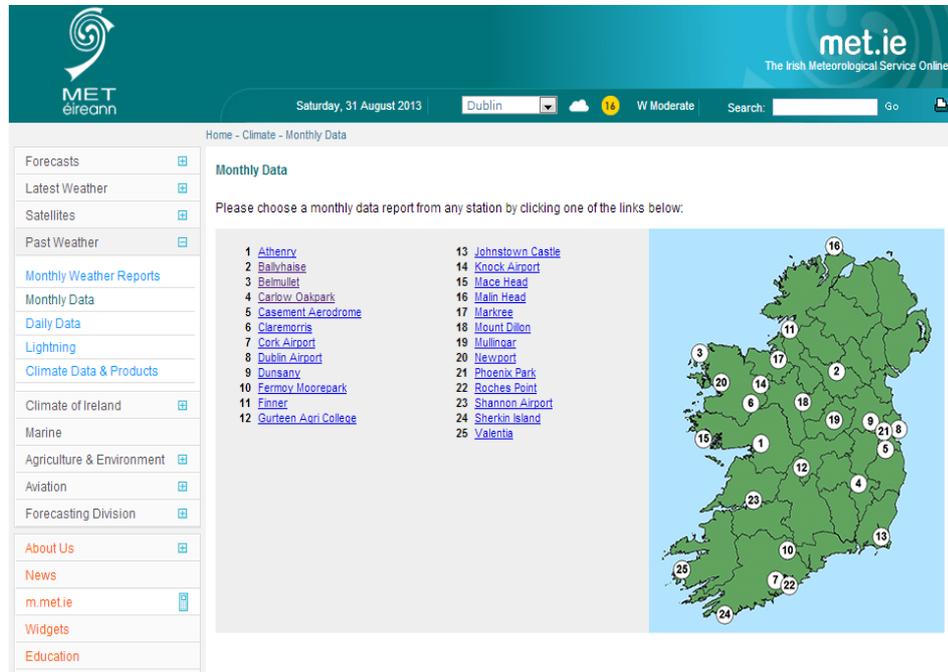


Figura 3.18 Portada web del Met.ie

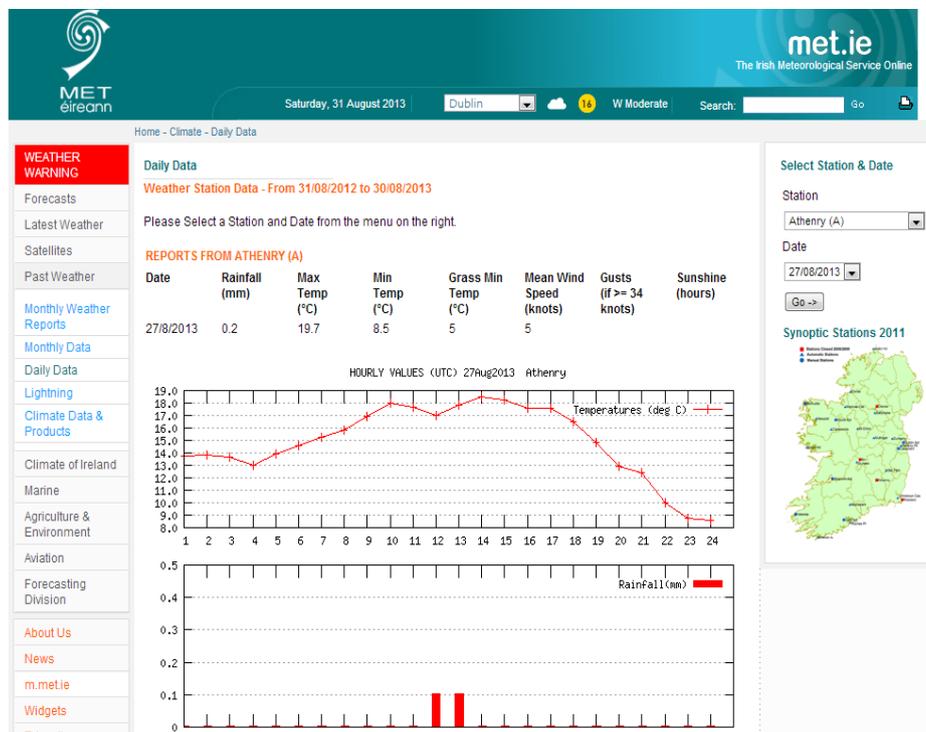


Figura 3.19 Datos diarios de la web del Met.ie

Ventajas y desventajas

En la Tabla 3-4, se muestran las ventajas y desventajas de la aplicación del servicio de meteorología irlandés “Met.ie”. Ésta te permite acceder desde cualquier dispositivo a la información, además se puede seleccionar una localidad concreta que se mostrará en la vista principal de la aplicación con los principales datos de tiempo.

Desde la aplicación móvil solo se puede acceder al último dato publicado, en cambio, desde la web te permite acceder a registros mensuales del estado del tiempo.

Por otra parte, no posee la característica de geolocalización para ver los datos de donde te encuentras aunque no es primordial ya que no hay una gran cantidad de localidades pero si sería recomendable para facilitar el uso al usuario.

Tampoco tiene una vista general donde poder ver una situación general de las distintas localidades de Irlanda y no tener que ir mirándolas una a una. Además no da la posibilidad de compartir los datos mediante las redes sociales y así promocionar el uso de la misma.

En resumen, la aplicación se puede mejorar notablemente añadiendo unas características básicas como pueden ser un mapa con la situación general del tiempo u otras características que hemos comentado anteriormente.

Ventajas	Desventajas
Multiplataforma.	No posee geolocalización.
Gratuita.	No se pueden compartir datos en redes sociales.
Permite seleccionar una estación concreta.	No posee un mapa general del estado de las lluvias o tablas con datos numéricos.
Se pueden acceder a datos antiguos desde la aplicación web.	

Tabla 3-4 Ventajas y desventajas de la aplicación Met.ie

3.3.2. Rain gauge

Rain gauge [6] es una aplicación móvil para las plataformas iOS que muestra las precipitaciones observadas en su ubicación GPS actual o en cualquier dirección que desee buscar.

La información sólo se proporciona en este momento para los 48 Estados Unidos continentales y Puerto Rico.

La aplicación cuenta con un pluviómetro graduado que se llena de agua con el sonido de la lluvia y los niveles de precipitación son en tiempo real, según lo registrado por la NOAA (National Oceanic and Atmospheric Administration).

Los usuarios tienen acceso rápido a su lista de lugares favoritos, permitiendo ver rápidamente los lugares de su interés. Además presenta su información con una interfaz sencilla para una amplitud de períodos históricos, que van desde un día hasta un año (Ver Figura 3.20).

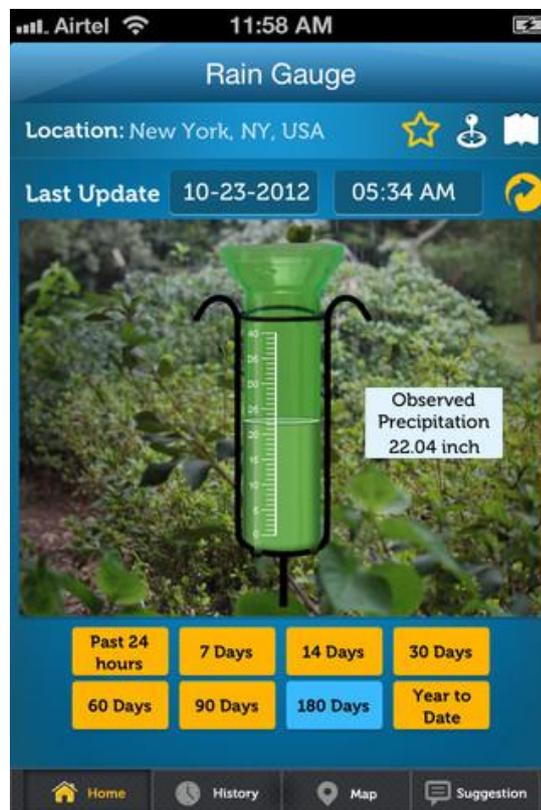


Figura 3.20 Vista principal de la aplicación Rain Gauge

Otra de las funcionalidades disponibles en Rain Gauge es una comparativa detallada con precisión de cuándo y cuántas precipitaciones se produjeron en un período de tiempo determinado y mapas visuales que indican la precipitación regional (Ver Figura 3.21).



Figura 3.21 Vista de las lluvias en un periodo y sus gráficas asociadas

Una de las características más distintivas de Rain Gauge con respecto a las otras es que la información recogida se puede compartir fácilmente como texto a través de una variedad de canales, incluyendo correo electrónico, Facebook y Twitter (Figura 3.22)

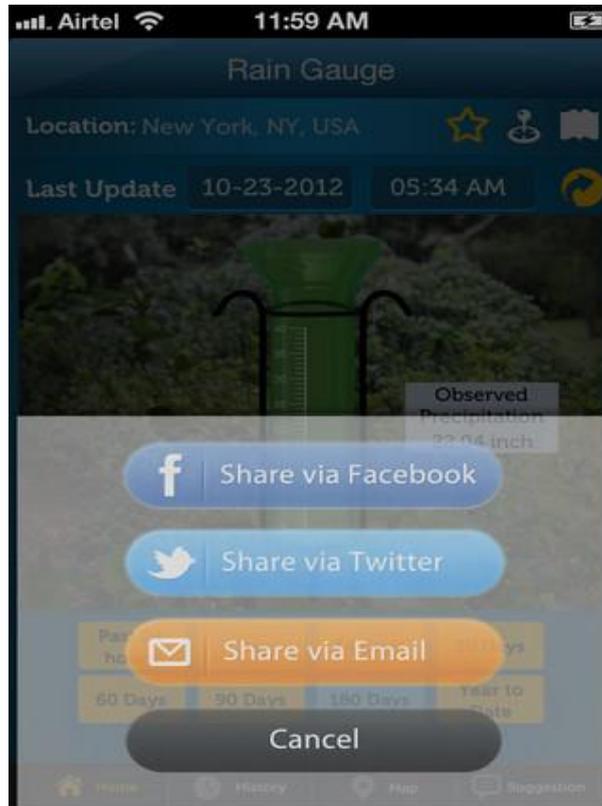


Figura 3.22 de compartir datos en las redes sociales

También posee un mapa pluviométrico donde se puede ver como se han distribuido las lluvias a lo largo del territorio estadounidense (Ver Figura 3.23).

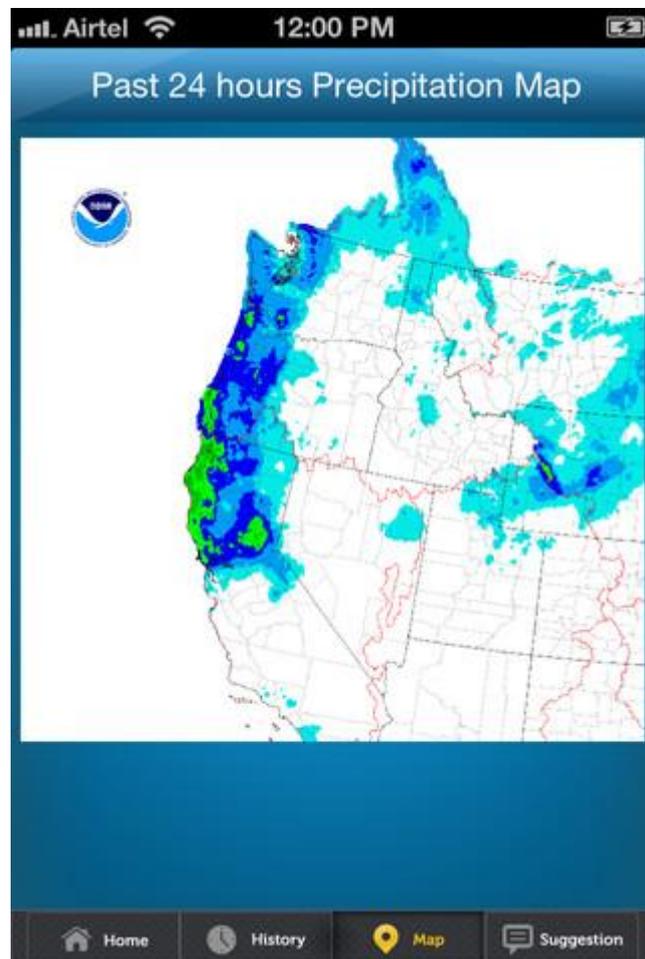


Figura 3.23 Mapa de precipitaciones

Ventajas y desventajas

En la Tabla 3-5 se muestran las ventajas y desventajas de la aplicación “Rain Gauge”, que es de las más completas que hemos analizado ya que posee bastantes características que facilitan la interacción del usuario, como son, geolocalización para encontrar la información de la estación más cerca, marcar estaciones favoritas para acceder directamente a éstas y también tiene un mapa pluviométrico que te muestra en qué zonas ha llovido más o menos.

Como dos puntos fuertes diferenciales de todas las aplicaciones que hemos estudiado, es la única que te permite acceder a datos antiguos de hasta más 90 días de antigüedad y además, te permite compartirlos mediante redes sociales o enviarlos por email.

Por otro lado, solo está disponible para el sistema operativo iOS de Apple y teniendo que pagar para poder descargarla ya que no tiene versión gratuita. Por ahora solo esta operativa en Estados Unidos y Puerto Rico.

En resumen, la aplicación posee características únicas que no tienen ninguna de las que hemos estudiado como hemos comentado anteriormente pero se ve muy penalizado por las desventajas de que no sea multiplataforma y sea de pago.

Ventajas	Desventajas
Posee geolocalización de la estación más cerca.	Es de pago.
Puedes tener estaciones favoritas.	No posee buscador de estaciones.
Puedes compartir datos mediante redes sociales.	No es multiplataforma.
Puedes acceder a datos antiguos.	
Mapas de lluvias.	

Tabla 3-5 Ventajas y desventajas de la aplicación Rain Gauge

3.4. Resumen

En la Tabla 3-6 se muestra un resumen de las distintas aplicaciones estudiadas, de las actualmente existentes en el mercado. En ella, aparecen de forma resumida las características que están presentes en cada una de las aplicaciones.

Las aplicaciones más completas son SAIH Ebro y Rain Gauge. De estas dos, Rain Gauge es la mejor debido a que las características que posee proporcionan al usuario funcionalidades más útiles en comparación con la otra.

Los inconvenientes de que Rain Gauge no sea multiplataforma, sea de pago son puntos negativos que pesan a la hora de elegir esta aplicación.

Por tanto, una vez visto y analizado lo que hay en el mercado se puede decir que no hay una solución óptima que favorezca y cumpla todas las características que se ven resumidas en la siguiente tabla. Aunque este estudio nos ha permitido analizar las características que poseen las aplicaciones, y que queremos ver incluidas en la nuestra.

Entre todas las características que hemos visto vamos a incluir la visualización de los datos de manera gráfica, mediante un mapa interactivo y en tablas porque así se mejora la experiencia del usuario permitiéndole diferentes formas de ver los datos.

Se permitirá compartir datos en las principales redes sociales ya que así facilitamos la difusión entre usuarios la información meteorológica.

Se ha decidido no poner un buscador de estaciones ni lugares favoritos ya que como la isla de Gran Canaria tiene 21 municipios consideramos que las formas de visualización que proporcionamos son más que suficiente para acceder a estos.

Y por último daremos acceso a los datos antiguos mediante la versión de escritorio y limitaremos a 30 días la consulta de estos en la versión del móvil y tableta.

	Pluviómetro de Andalucía	SAIH Ebro	AEMET	Rain Raige	Met.ie
Plataforma		  	  		  
Gratuita	SI	SI	SI	NO	SI
Geolocalización	SI	NO	SI	SI	NO
Acceso a datos antiguos	NO	Hasta 15 días	Hasta 7 días	Hasta 1 año	SI
Exportación de datos	NO	Desde escritorio.	Desde escritorio	NO	Desde escritorio
Compartir datos en redes sociales	NO	NO	NO	SI	NO
Visualización	Una gráfica por estación	Tablas/ Gráficas	Tabla/ Gráfica	Gráficas	Tabla
Lugares Favoritos	NO	NO	SI	SI	SI
Buscador de estaciones	NO	SI	SI	NO	NO
Mapas pluviométricos.	NO	SI	NO	NO	NO

Tabla 3-6 Comparativa de las aplicaciones pluviométricas

4. Metodología

En este apartado, se describirá la metodología y lenguaje de modelado que usaremos para el análisis y diseño de la aplicación web.

Modelo de proceso de software

La metodología empleada en el desarrollo software del proyecto ha sido el modelo en espiral, propuesto originalmente por Boehm en 1988, es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial [7].

Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.

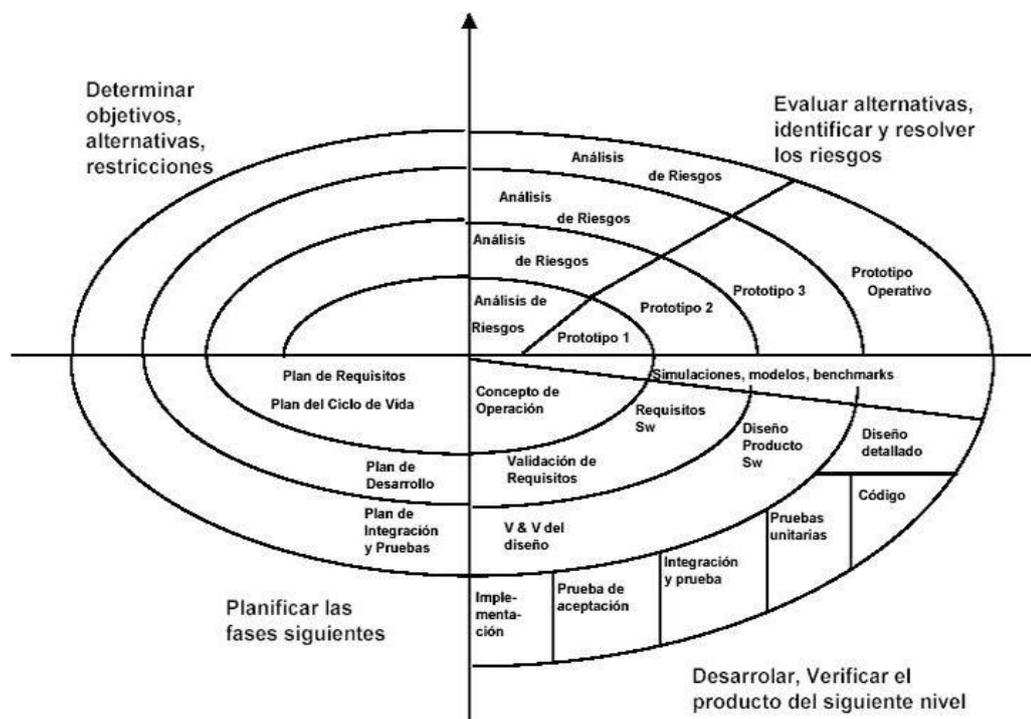


Figura 4.1 Esquema del modelo en espiral

El modelo en espiral se divide en un número de actividades de marco de trabajo, también llamadas regiones de tareas. Generalmente, existen entre tres y seis regiones de tareas. El modelo en espiral contiene seis regiones de tareas:

- **Comunicación con el cliente:** se trata de las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- **Planificación:** las tareas requeridas para definir recursos, el tiempo y otra información relacionadas con el proyecto.
- **Análisis de riesgos:** las tareas requeridas para evaluar riesgos técnicos y de gestión.
- **Ingeniería:** las tareas requeridas para construir una o más representaciones de la aplicación.
- **Construcción y acción:** las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario, como documentación y práctica.
- **Evaluación del cliente:** las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación.

Cuando empieza este proceso evolutivo, se gira alrededor de la espiral en la dirección de las agujas del reloj, comenzando por el centro. El primer circuito de la espiral puede producir el desarrollo de una especificación de productos; los pasos siguientes en la espiral se podrían utilizar para desarrollar un prototipo y progresivamente versiones más sofisticadas del software. Cada paso por la región de planificación produce ajustes en el plan del proyecto. El coste y la planificación se ajustan con la realimentación ante la evaluación del cliente.

El modelo en espiral es un enfoque realista del desarrollo de sistemas y de software a gran escala. Como el software evoluciona, a medida que progresa el proceso, el desarrollador y el cliente comprenden

y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos. El modelo en espiral utiliza la construcción de prototipos como mecanismo de reducción de riesgos, pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto. Mantiene el enfoque sistemático de los pasos sugeridos por el ciclo de vida clásico, pero lo incorpora al marco de trabajo iterativo que refleja de forma más realista el mundo real.

Lenguaje de modelado

El lenguaje de modelado utilizado a lo largo del proyecto será UML o Lenguaje Unificado de Modelado. Permite expresar un modelo de análisis utilizando una notación de modelado con unas reglas sintácticas, semánticas y prácticas. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema. Además de aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Se emplea para definir un sistema, para detallar los artefactos en el mismo, para documentar y construir.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, sólo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos es un complemento de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

El proyecto descrito en este documento se desarrollará en base al modelo de proceso de software y el lenguaje de modelado indicados. A la hora de analizar y diseñar el sistema a desarrollar, se emplearán los

diagramas elaborados siguiendo las pautas establecidas por el Lenguaje Unificado de Modelado UML. Entre estos diagramas destacan la identificación de actores, casos de uso, de clases y de secuencia.

El desarrollo del software se llevará a cabo utilizando el Modelo de Proceso en Espiral. Se irán desarrollando prototipos de validación tras pasar por cada una de las fases descritas en el modelo, sobre todo en las tareas fundamentales de análisis, diseño e implementación.

5. Recursos Utilizados

En este apartado, se mostrarán todos los recursos utilizados para desarrollo del proyecto, divididos en recursos software y hardware.

5.1. Recursos Hardware

En este apartado, mostraremos todos los recursos hardware de los cuales hicimos uso a lo largo del proyecto.

Equipos

Para la elaboración y desarrollo del proyecto se han usado dos equipos informáticos, un ordenador de sobremesa y un portátil. A continuación describimos sus características técnicas.

Ordenador – Sobremesa

- Intel Pentium Dual 2.00Ghz
- 4Gb de memoria RAM
- 500Gb de disco duro
- Sistema operativo Windows 7 Profesional

Ordenador – Portátil

- Intel Pentium 2.00Ghz
- 4Gb memoria RAM
- 250Gb de disco duro
- Sistema operativo Windows 7 Profesional

Servidor Web

Será necesario un servidor web que aloje nuestra aplicación para que nuestros usuarios puedan hacer uso de la misma. Este equipo deberá poseer unas características hardware que le permitan ofrecer la información a múltiples clientes en el mismo momento, con un tiempo de respuesta aceptable, ya que como el ámbito de nuestra aplicación es basado en Web, estos parámetros son críticos para dar un buen servicio al usuario.

Los requisitos mínimos que debe tener el servidor son los siguientes:

- Sistema Operativo Linux, CentOS o Ubuntu Server
- 4Gb Memoria RAM
- 80GB HDD

Dispositivos Móviles

Para poder probar los diferentes diseños de la aplicación web hicimos uso de los siguientes dispositivos:

Móvil

- iPhone 4
- Pantalla de 3.5 pulgadas
- Sistema Operativo iOS 7

Tableta

- Acer Iconia
- Pantalla de 10 pulgadas
- Sistema Operativo superior a Android 4.0

Escritorio

- Intel Pentium Dual E2180 2.00 GHz
- 4Gb Memoria RAM
- 500GB HDD
- Pantalla Acer 17 pulgadas
- Sistema Operativo Windows 7

5.2. Recursos Software

En este apartado, mostraremos todos los recursos software de los cuales hicimos uso a lo largo del proyecto.

Editor web

Para la creación de nuestra aplicación web es necesario un editor compatible con las diversas tecnologías que vamos a usar para así facilitar y optimizar el desarrollo.

Nos hemos decantado por el editor Sublime-Text, que soporta multitud de lenguajes y además posee una serie de características que facilitan y optimizan el trabajo del desarrollador.

Como pueden ser las siguientes

- **Minimapa:** consiste en una previsualización de la estructura del código, es muy útil para desplazarse por el archivo cuando se conoce bien la estructura de este.
- **Multi Selección:** Hace una selección múltiple de un término por diferentes partes del archivo.
- **Multi Cursor:** Crea cursores con los que podemos escribir texto de forma arbitraria en diferentes posiciones del archivo.
- **Multi Layout:** Trae siete configuraciones de plantilla podemos elegir editar en una sola ventana o hacer una división de hasta cuatro ventanas verticales o cuatro ventanas en cuadrícula.
- **Resaltado de paréntesis e indentación:** Cuando el usuario coloca el cursor en un paréntesis, corchete o llave, resalta ésta y el paréntesis, corchete o llave de cierre o apertura correspondiente.

Editor de prototipos

Una de las fases más importantes de la aplicación web, es la creación de la interfaz del usuario, por tanto, necesitamos un editor que nos proporcione facilidad y rapidez a la hora de crearla.

Para la creación de la interfaz hemos usado el Balsamiq Mockups, que es un constructor de mockups, que nos da multitud de objetos gráficos para crear tu diseño de una manera rápida y sencilla.

Editor UML

En diferentes fases de la creación del proyecto, necesitamos elaborar una serie de diagramas que nos ayudarán a separar y a entender todos los aspectos de la aplicación.

Para ello, usamos el Lenguaje Unificado de Modelado UML, por tanto, hemos usado el StarUML, que es un editor compatible con este lenguaje, que soporta y facilita la creación de los diagramas.

Editor de bases de datos

En la fase de la creación del modelo de datos, hemos hecho uso del programa MySQL Workbench, que es una herramienta visual para la creación del diseño de la base de datos.

Editor de textos

Para elaboración de documentos del proyecto se ha usado el procesador de textos Microsoft Word 2007, para llevar a cabo la memoria, manuales necesarios para el mismo.

Control de versiones

Cuando nos encontramos en la fase desarrollo o implementación es necesario contar con un software de control de versiones, donde puede ir guardando los avances realizados y tener un registro de los mismos.

En nuestro proyecto hemos usado GIT, que es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

Para usar GIT, hemos elegido como repositorio del proyecto, Bitbucket, que es una aplicación Web que permite alojar repositorios Git privados y compartirlos entre varios usuarios.

5.2.1. Herramientas

En este apartado, mostraremos las herramientas usadas en el desarrollo de la aplicación.

Foundation – Framework

Es una colección de herramientas para la creación de sitios y aplicaciones web que contiene plantillas de HTML y CSS de diferentes diseños de tipografía, formas, botones, navegación y otros componentes de la interfaz, así como plugins de JavaScript que te facilitan y aumentan la velocidad del desarrollo

Ruby on Rails (RoR)

Es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otras herramientas y con un mínimo de configuración.

PostgreSQL

Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicada bajo licencia BSD. Es un programa de código abierto, por lo que está dirigido por una comunidad de desarrolladores llamada PGDG (*PostgreSQL Global Development Group*).

Comienza su desarrollo en el año 1982 con el proyecto Ingres en la Universidad de Berkeley. Entre sus principales características tenemos la alta concurrencia, la amplia variedad de tipos nativos, y diversas funciones más específicas.

JQuery

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

Leaflet.js

Es una librería JavaScript de open-source ampliamente utilizada para construir mapas interactivos en aplicaciones web, está diseñada para que su uso sea simple, y de alto rendimiento. Funciona de manera eficiente a través de todas las principales plataformas de escritorio y móviles, aprovechando HTML5 y CSS3 en los navegadores modernos sin dejar de ser accesibles en otros más antiguos. Es utilizada por FourSquare, Pinterest y Flickr

5.2.2. Tecnologías

En este apartado mostraremos las tecnologías usadas para el desarrollo de la aplicación web.

HTML

Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, entre otros.

JAVASCRIPT

Es un lenguaje de programación interpretado, se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

CoffeeScript

Es un lenguaje de programación que se compila a JavaScript. El lenguaje añade “azúcar sintáctico” inspirado en Ruby, Python y Haskell² para mejorar la brevedad y la legibilidad de JavaScript, y añade características más sofisticadas, como la comprensión de listas y la coincidencia de patrones. CoffeeScript compila previsiblemente a JavaScript y los programas se pueden escribir por lo general un 33% menos de líneas de código sin ningún efecto en el rendimiento ni en el tiempo de ejecución.

CSS

Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML, XML o XHTML. La idea principal que se

encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

De esta manera, se ha utilizado CSS para dar formato a cada uno de los documentos HTML desarrollados a través de sus etiquetas. Mediante esta tecnología, se facilita la presentación de contenidos y se ofrece una interfaz de usuario más accesible y clara.

AJAX

Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

En la aplicación web que se ha desarrollado, se presenta en numerosas ocasiones la necesidad de mostrar contenidos al usuario sin la necesidad de recargar la página al completo.

En ejemplos como el buscador o compartir los datos por email se observa que la ejecución asíncrona permite mostrar los resultados al usuario sin ningún tipo de interrupción.

Ruby

Es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995.

Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk.

Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y CLU. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre.

SQL

El lenguaje de consulta estructurado SQL (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ellas.

JSON

Es un formato ligero para el intercambio de datos, es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

GeoJSON

Es un formato estándar para la codificación de las colecciones de las características geográficas simples junto con sus atributos no espaciales que utilizan JavaScript Object Notation.

TopoJSON

TopoJSON es una extensión de GeoJSON que codifica topología, a partir de segmentos de líneas compartidas llamados arcos.

Cada arco se define sólo una vez, pero puede hacer referencia a varias veces por diferentes formas, lo que reduce la redundancia y disminuye el tamaño del archivo enormemente.

6. Análisis

En esta etapa comprenderemos de forma detallada cual es la problemática a resolver, verificando el entorno en el cual se encuentra dicho problema, de tal manera que se obtenga la información necesaria y suficiente para afrontar su respectiva solución.

6.1. Requisitos de la aplicación Web

A esta aplicación se va acceder desde diferentes tipos de dispositivos, por tanto, necesitamos crear una interfaz adaptativa a móviles, tabletas y escritorios, siguiendo la filosofía del Mobile First, que explicaremos más adelante en el modelo de interacción.

La estructura de las vistas irá cambiando según cambie el tamaño de la pantalla añadiendo o quitando elementos para facilitar la usabilidad del software. En cuanto, a las funcionalidades, serán idénticas en todas, salvo en la versión de escritorio, que poseerá una adicional, que será el buscador.

Todo ello, estará reflejado en los requisitos software que veremos más adelante.

6.1.1. Requisitos del software

Para poder elaborar el modelo de casos de uso, se determinarán diferentes aspectos relevantes. Por un lado, se deberán identificar los actores del sistema así como su papel en el mismo. Por otro lado, se indicarán cada una de sus acciones correspondientes.

Identificación de actores

En nuestra aplicación web solo habrá un actor principal (Ver figura 6.1), que será el usuario no registrado, ya que solo seremos un portal para mostrar la información y por tanto, para esta versión del proyecto no será necesario ningún usuario más.

Actores



6.1 Diagrama de actores del sistema

Actor	Tipo	Definición
Usuario no registrado	Principal	Este usuario podrá acceder a todas las funcionalidades de la aplicación web sin necesidad de registrarse.

Tabla 6-1 Lista de actores del sistema

Actores – Acciones

En el siguiente apartado, se muestran las acciones asociadas a los actores, además de cuales se pueden usar según el dispositivo (VerTabla 6-2).

Acciones

- **Seleccionar municipio**

En la aplicación web se podrá seleccionar el municipio del que desee obtener la información.

- **Compartir datos**

Se podrá compartir mediante las redes sociales o email una imagen de la vista de los datos en los que se encuentre.
- **Ver mapa**

Se puede ver los datos de la vista principal mediante una imagen de una isla de Gran Canaria.
- **Ver tabla**

Cuando se encuentre en la vista municipios o estaciones podrá ver estos datos en modo de tabla.
- **Ver gráfica**

Cuando se encuentre en la vista municipios o estaciones podrá ver estos datos representados en una gráfica.
- **Seleccionar estación**

Cuando los datos están visualizados como una tabla se podrá seleccionar la estación específica que se desee.
- **Filtrar datos**

Cuando los datos están en formato de tabla en la vista principal se podrá filtrar estos, por municipios o estaciones más lluviosas.
- **Navegación**

En la parte superior aparecerá un registro desde la vista principal hasta la actual para que así pueda navegar por la aplicación web.
- **Buscador**

También se puede usar el buscador avanzado para ver datos más específicos

Actor	Acción	Móvil	Tableta	Ordenador
Usuario no registrado	Seleccionar municipio	X	X	X
	Compartir datos	X	X	X
	Ver mapa	X	X	X
	Ver tabla	X	X	X
	Ver gráfica	X	X	X
	Seleccionar estación	X	X	X
	Filtrar datos	X	X	X
	Navegación	X	X	X
	Buscador			X

Tabla 6-2 Lista de acciones de actores

Diccionario de conceptos

Además de identificar los actores del sistema, así como cada una de las acciones que pueden realizar en el mismo, se presentan a continuación una serie de conceptos que deben ser manejados por la aplicación.

Estaciones

Las estaciones son unos aparatos llamados pluviómetros que se encargan de medir y enviar la cantidad de lluvia que ha caído en el punto donde se encuentran cada 24h.

Precipitación

Es la cantidad de agua recogida por una estación.

Datos

Los datos en nuestra aplicación web se refieren a las cantidades de precipitación en mililitros por hora que recojan las diferentes estaciones.

Dispositivos

Los dispositivos se entienden por los diferentes aparatos donde se puede ver y usar nuestra aplicación web, concretamente son los móviles, tabletas y ordenadores.

Modelo de casos de uso

Una vez identificados los actores del sistema, se podrá elaborar el diagrama general de casos de uso, así como la explicación detallada de cada uno. En el siguiente diagrama se presentan cada uno de los casos de uso del sistema a desarrollar, que serán desglosados y explicados detalladamente a continuación en forma de tablas.

Usuario No Registrado

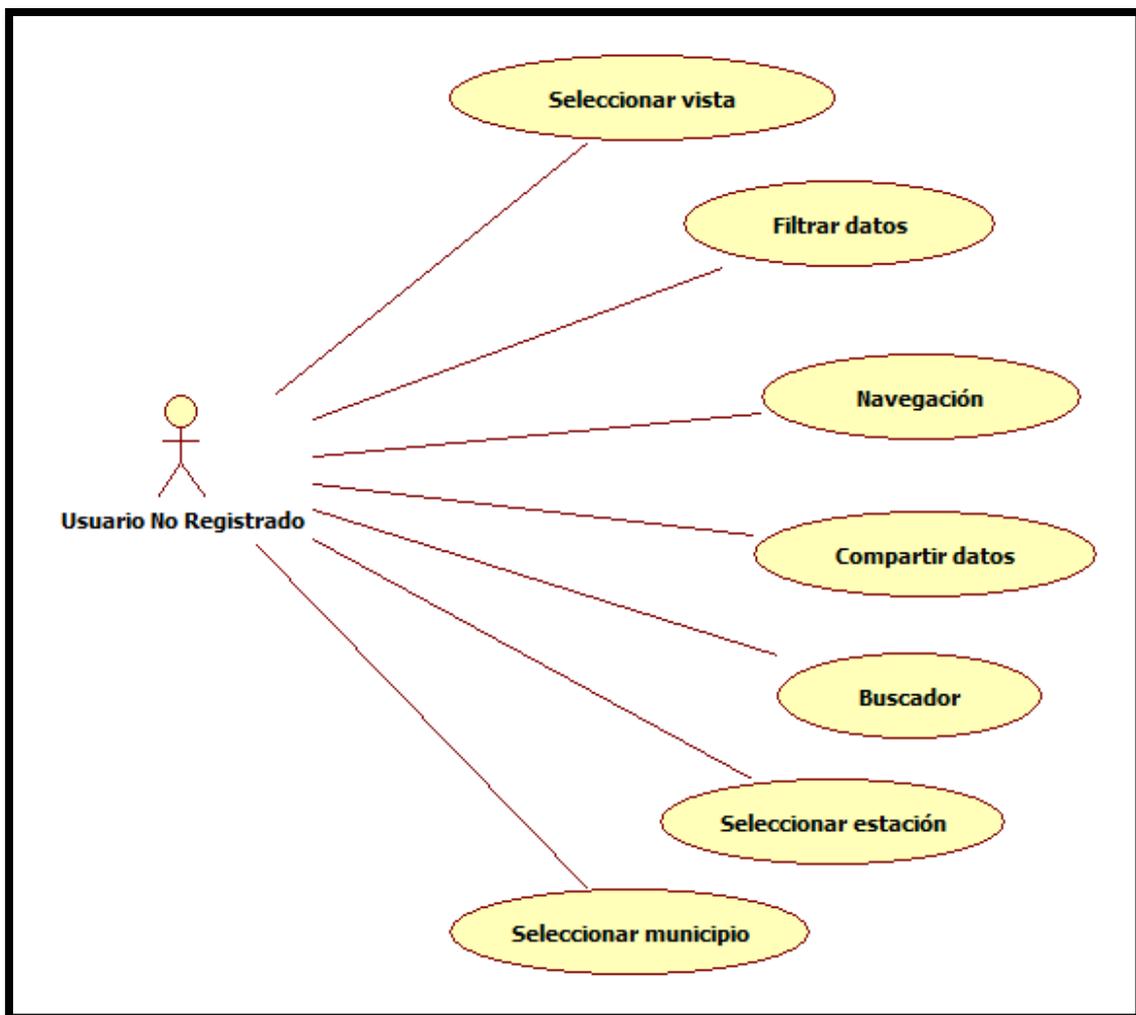


Figura 6.2 Usuario no registrado - Casos de uso generales

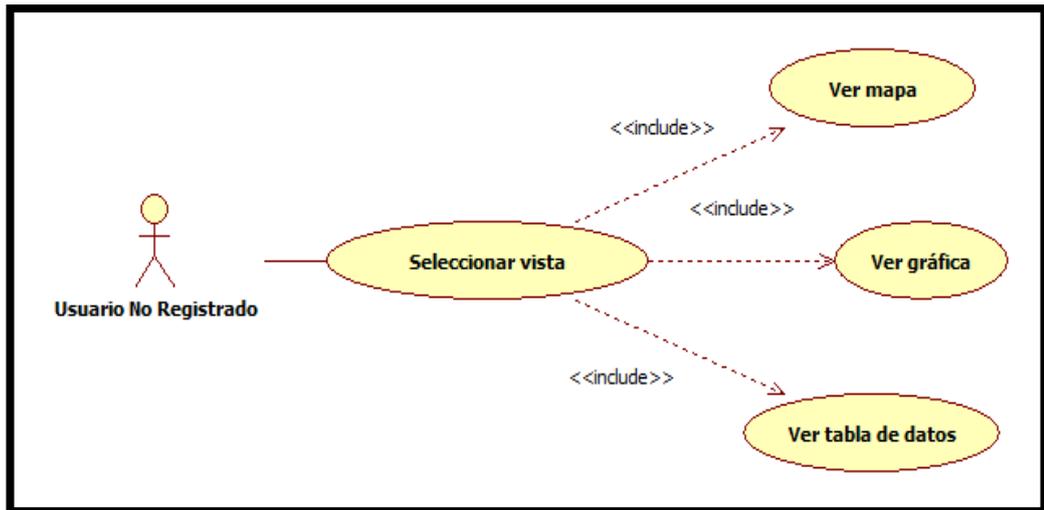


Figura 6.3 Usuario sin registrar - Seleccionar Vista

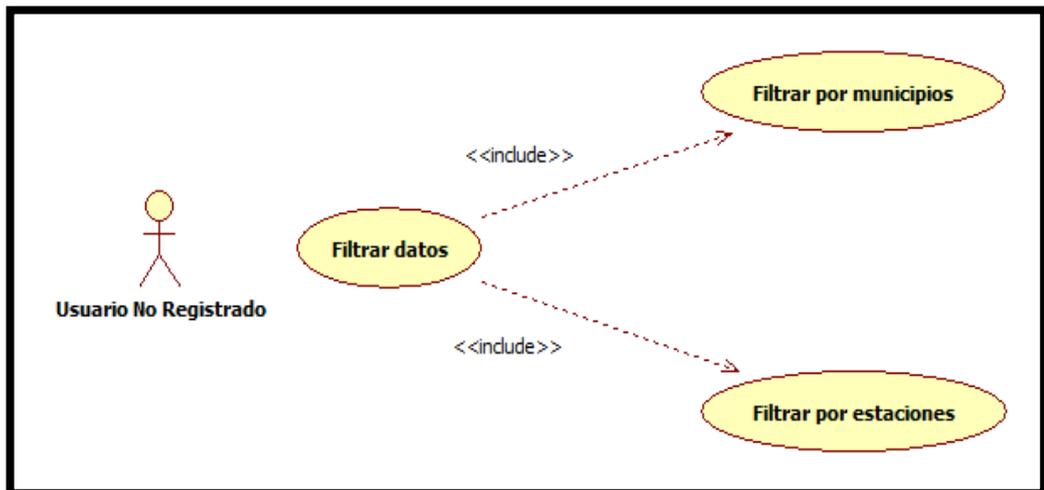


Figura 6.4 Usuario sin registrar - Filtrar por municipios / estaciones

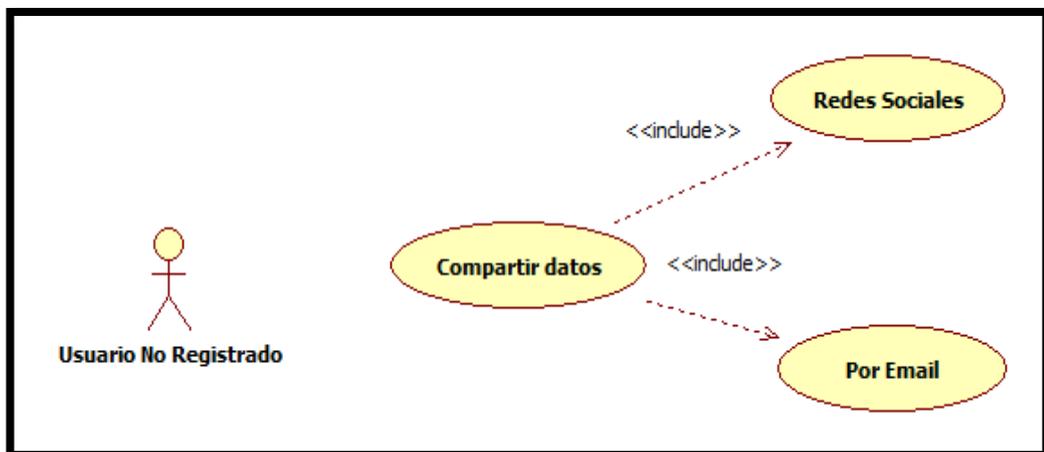


Figura 6.5 Usuario sin Registrar - Compartir datos

Casos de Uso Detallados

Se presentan cada uno de los casos de uso, de forma detallada a continuación.

Nombre	Seleccionar Vista	Identificador	CU – 1
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
<ul style="list-style-type: none"> - <u>Usuario No Registrado:</u> Seleccionará como desea que la información sea mostrada. 			
Descripción			
<ul style="list-style-type: none"> - Al acceder a la aplicación web el “Usuario No registrado” puede seleccionar la forma de visualizar la información. Puede ser en forma de mapa o de tabla de datos, donde aparecerán los municipios y estaciones. Por defecto, se mostrarán los datos en forma de mapa. - Cuando se esté en la vista de municipios o estaciones se podrá ver los datos representados en una gráfica 			
Trigger			
Precondición			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” accede a la aplicación web 2. Por defecto, se mostrará el mapa. 3. El “Usuario No Registrado” podrá seleccionar como ver la información. <ol style="list-style-type: none"> 3.1. Seleccionará el botón Ver Tabla CU 3 para ver la información como una tabla con los datos escritos. 3.2. Seleccionará el botón Ver Mapa CU – 2 para ver la información como un mapa con los datos representados. 4. Si el “Usuario No Registrado” está en la vista de un municipio o estación podrá ver la información representada por una gráfica, para ello, seleccionará el botón Ver Gráfica CU – 4. 			
Flujo Alternativo			

Excepción
Includes
<ul style="list-style-type: none">- Se utilizan los casos de uso<ul style="list-style-type: none">▪ Seleccionar Vista – Mapa · CU – 2▪ Seleccionar Vista – Tabla · CU – 3▪ Seleccionar Vista – Gráfica · CU – 4
Requisitos Especiales
Notas

Nombre	Seleccionar Vista – Mapa	Identificador	CU – 2
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
- <u>Usuario No Registrado</u> : Seleccionará que la información sea mostrada en forma de mapa.			
Descripción			
- El “Usuario No Registrado” elegirá que la información le sea mostrada en forma de mapa, donde aparecerán los municipios, estaciones y las precipitaciones que han caído.			
Trigger			
Precondición			
- Para poder hacer el cambio de vista, se tendrá que estar en la vista de la tabla.			
Postcondición			
Flujo Normal			
1. El “Usuario No Registrado” seleccionará el botón Ver Mapa. 2. Se mostrará la información mediante un mapa.			
Flujo Alternativo			
Excepción			
Includes			
Requisitos Especiales			
Notas			

Nombre	Seleccionar Vista – Tabla	Identificador	CU – 3
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
- <u>Usuario No Registrado</u> : Seleccionará que la información sea mostrada en forma de tabla.			
Descripción			
- El “Usuario No Registrado” elegirá que la información le sea mostrada en forma de tabla, donde aparecerán los municipios, estaciones y las precipitaciones que han caído.			
Trigger			
Precondición			
- Para poder hacer el cambio de vista, se tendrá que estar en la vista del mapa.			
Postcondición			
Flujo Normal			
1. El “Usuario No Registrado” seleccionará el botón Ver Tabla. 2. Se mostrará la información mediante una tabla.			
Flujo Alternativo			
Excepción			
Includes			
Requisitos Especiales			
Notas			

Nombre	Seleccionar Vista – Gráfica	Identificador	CU – 4
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
<ul style="list-style-type: none"> - <u>Usuario No Registrado</u>: Seleccionará que la información sea mostrada en forma de tabla. 			
Descripción			
<ul style="list-style-type: none"> - El “Usuario No Registrado” elegirá que la información le sea mostrada en forma de tabla, donde aparecerán los municipios, estaciones y las precipitaciones que han caído. 			
Trigger			
Precondición			
<ul style="list-style-type: none"> - Para poder hacer el cambio de vista, se tendrá que estar en la vista de un municipio o estación. 			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” seleccionará el botón Ver Gráfica. 2. Se mostrará la información mediante una tabla. 			
Flujo Alternativo			
Excepción			
Includes			
Requisitos Especiales			
Notas			

Nombre	Seleccionar Municipio	Identificador	CU – 5
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
<ul style="list-style-type: none"> - <u>Usuario No Registrado</u>: Seleccionará el municipio ya sea en la vista del mapa o de la tabla el municipio para que se le muestren las estaciones asociadas al mismo. 			
Descripción			
<ul style="list-style-type: none"> - El “Usuario No Registrado” seleccionará el municipio que desee y aparecerá una vista donde se verán las estaciones asociadas al mismo en forma de tabla, donde aparecerán los datos de las precipitaciones que han caído y la fecha. 			
Trigger			
Precondición			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” accede a la aplicación web 2. Selecciona el municipio deseado, se mostrará la vista con las estaciones asociadas al mismo. 			
Flujo Alternativo			
Excepción			
Includes			
Requisitos Especiales			
Notas			

Nombre	Seleccionar Estación	Identificador	CU – 6
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
<ul style="list-style-type: none"> - <u>Usuario No Registrado</u>: Seleccionará la estación ya sea en la vista del mapa o de la tabla el municipio para que se le muestren los datos asociadas a la misma. 			
Descripción			
<ul style="list-style-type: none"> - El “Usuario No Registrado” seleccionará la estación que desee y aparecerá una vista donde se verán los datos asociados a la misma en forma de tabla o gráfica, donde aparecerán las precipitaciones que han caído y la fecha. 			
Trigger			
Precondición			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” accede a la aplicación web 2. Selecciona la estación deseada. 3. Se muestra la vista con los datos asociados a la misma. 			
Flujo Alternativo			
Excepción			
Includes			
Requisitos Especiales			
Notas			

Nombre	Filtrar Datos	Identificador	CU – 7
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
- <u>Usuario No Registrado:</u> Filtrará la información mostrada por municipios o estaciones.			
Descripción			
- El “Usuario No Registrado” filtrará los datos mostrados y se ordenarán por orden decreciente de las precipitación media de los municipios o de las estaciones. Por defecto, estará seleccionada “Por municipios”.			
Trigger			
Precondición			
- Estar en la vista de la tabla.			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” accede a la vista de la tabla. 2. El “Usuario No Registrado” selecciona el criterio de filtrado. <ol style="list-style-type: none"> 2.1. Seleccionará el botón Por municipios CU – 8 en orden decreciente de precipitación. 2.2. Seleccionará el botón Por estaciones CU – 9 en orden decreciente de precipitación. 			
Flujo Alternativo			
Excepción			
Includes			
<ul style="list-style-type: none"> - Se utilizan los casos de uso <ul style="list-style-type: none"> ▪ Filtrar datos – Por municipios · CU – 8 ▪ Filtrar datos – Por estaciones · CU – 9 			
Requisitos Especiales			
Notas			

Nombre	Filtrar Datos – Por municipios	Identificador	CU – 8
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
<ul style="list-style-type: none"> - <u>Usuario No Registrado</u>: Filtrará la información sea ordenada por la precipitación en orden decreciente de cada municipio. 			
Descripción			
<ul style="list-style-type: none"> - El “Usuario No Registrado” elegirá que la información le sea ordenada por municipios, donde aparecerán en orden decreciente según las precipitaciones que han caído. 			
Trigger			
Precondición			
<ul style="list-style-type: none"> - Para poder ordenar los datos por municipios, tendrán que estar filtrados por estaciones. 			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” seleccionará el botón Por Municipios. 2. Se le ordenarán los datos en orden decreciente por precipitación de cada municipio. 			
Flujo Alternativo			
Excepción			
Includes			
Requisitos Especiales			
Notas			

Nombre	Filtrar Datos – Estaciones	Identificador	CU – 9
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
<ul style="list-style-type: none"> - <u>Usuario No Registrado</u>: Filtrará la información sea ordenada por la precipitación en orden decreciente de cada municipio. 			
Descripción			
<ul style="list-style-type: none"> - El “Usuario No Registrado” elegirá que la información le sea ordenada por municipios, donde aparecerán en orden decreciente según las precipitaciones que han caído. 			
Trigger			
Precondición			
<ul style="list-style-type: none"> - Para poder ordenar los datos por estaciones, tendrán que estar filtrados por municipios. 			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” seleccionará el botón Por Estación. 2. Se le ordenarán los datos en orden decreciente por precipitación de cada estación. 			
Flujo Alternativo			
Excepción			
Includes			
Requisitos Especiales			
Notas			

Nombre	Navegación	Identificador	CU – 10
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
<ul style="list-style-type: none"> - <u>Usuario No Registrado</u>: Se podrá navegar entre las diferentes vistas de la aplicación mediante un elemento llamado “BreadCrumbs” que estará situado en la parte superior de la aplicación 			
Descripción			
<ul style="list-style-type: none"> - El “Usuario No Registrado” podrá navegar entre las diferentes vistas de la aplicación mediante el elemento “BreadCrumbs”. 			
Trigger			
Precondición			
<ul style="list-style-type: none"> - Estar en un vista diferente de la principal 			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” entra en alguna vista secundaria de la aplicación. 2. Se mostrará el elemento llamado “BreadCrumbs” 3. Y el “Usuario No Registrado” elegirá a que vista desee ir. 			
Flujo Alternativo			
Excepción			
Includes			
Requisitos Especiales			
Notas			

Nombre	Compartir	Identificador	CU – 11
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
- <u>Usuario No Registrado</u> : Podrá compartir los datos en forma de imagen en las redes sociales o enviarlo por email.			
Descripción			
- El “Usuario No Registrado” podrá compartir los datos en las redes sociales o enviarlo por email.			
Trigger			
Precondición			
- Loguearse en las redes sociales para poder compartir la foto o poner un email correcto para enviar la información.			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” seleccionará el botón de compartir. 2. Le aparecerá un menú donde elegirá entre las redes sociales o enviar por email. <ol style="list-style-type: none"> 2.1. Seleccionará el botón de la red social elegida Compartir – Redes Sociales CU – 12. 2.2. Seleccionará el botón enviar por email Compartir – Por Email CU – 13. 3. Mensaje de confirmación. 			
Flujo Alternativo			
Excepción			
2.* Si los datos de autenticación o la dirección de email es incorrecta, se le mostrará un mensaje de error al usuario para que vuelva a introducir los datos correctamente.			
Includes			
<ul style="list-style-type: none"> - Se utilizan los casos de uso <ul style="list-style-type: none"> ▪ Compartir – Redes Sociales · CU – 12 ▪ Compartir – Por Email · CU – 13 			
Requisitos Especiales			
Notas			

Nombre	Compartir Redes Sociales	Identificador	CU – 12
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
<ul style="list-style-type: none"> - <u>Usuario No Registrado</u>: Podrá compartir mediante las redes sociales de Facebook o Twitter los datos de la aplicación mediante una imagen. 			
Descripción			
<ul style="list-style-type: none"> - El “Usuario No Registrado” podrá publicar en Facebook o Twitter imágenes de datos concretos de la aplicación. 			
Trigger			
Precondición			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” selecciona la red social en la cual desea compartir. 2. Introduce los datos de acceso a la misma. 3. Realiza la publicación de la imagen. 			
Flujo Alternativo			
Excepción			
Includes			
Requisitos Especiales			
Notas			

Nombre	Compartir Por email	Identificador	CU – 13
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
- <u>Usuario No Registrado</u> : Podrá enviar los datos por email.			
Descripción			
- El “Usuario No Registrado” podrá enviar los datos en forma de imagen por email.			
Trigger			
Precondición			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. El “Usuario No Registrado” selecciona enviar por email. 2. Se le abre su cliente de correo con los datos. 			
Flujo Alternativo			
Excepción			
Includes			
Requisitos Especiales			
Notas			

Nombre	Buscador	Identificador	CU – 14
Actor principal			
Usuario No Registrado			
Personal involucrado o intereses			
- <u>Usuario No Registrado</u> : podrá buscar datos antiguos de las precipitaciones.			
Descripción			
- El “Usuario No Registrado” cuando acceda a un ordenador tendrá la posibilidad de usar un buscador para encontrar datos de estaciones o municipios antiguos.			
Trigger			
Precondición			
- Acceder a la aplicación desde un ordenador y seleccionar al menos un municipio o una estación.			
Postcondición			
Flujo Normal			
<ol style="list-style-type: none"> 1. Selecciona el botón del Buscador. 2. Establece los parámetros de fecha, municipio o estaciones. 3. Le aparecerán los resultados de la búsqueda. 			
Flujo Alternativo			
2.1. Si no se han establecido al menos un municipio o estación, se le mostrará al “Usuario No Registrado” un mensaje avisándole del error.			
Excepción			
Includes			
Requisitos Especiales			
Notas			

6.2. Modelo de Interacción

En este apartado se explicará la filosofía seguida para realizar los diferentes diseños de cada dispositivo y el porqué de esta elección.

Según las estadísticas de varias empresas como Cisco, cada vez son más las personas que usan el móvil o una tableta en vez de un ordenador para conectarse a internet. Por tanto, si las previsiones siguen en esta misma línea, los dispositivos móviles están siendo y serán más que fundamentales en el día a día de las personas. Por eso hemos decidido usar la filosofía “Mobile First” para el diseño de nuestra aplicación web.

MobileFirst [8] es una filosofía creada por Luke Wroblewski en 2009, que tiene como idea básica realizar primero el diseño para los móviles y según vaya aumentando las dimensiones de la pantalla se van añadiendo más contenido y funcionalidades según la importancia de éstas para el usuario (Ver Figura 6.6).



Figura 6.6 Diagrama del funcionamiento del Mobile First

Gracias a esta filosofía, tendremos un diseño más preciso y centrado tanto en los contenidos como en las funcionalidades más importantes para el usuario, mejorando así, la experiencia de éste y aprovechando los elementos propios de cada dispositivo en cuestión, como pueden ser el GPS, o bluetooth entre otros. Se consigue así aumentar, la audiencia de la

página llegando cada vez a más gente. Además, se facilita la integración de las diferentes funcionalidades a medida que va cambiando el dispositivo.

Mockups

Los mockups son maquetas que se usan para hacer un boceto preliminar de cómo sería la interfaz de nuestra aplicación y la navegación a través de ésta, entre las diferentes secciones y vistas de nuestra aplicación. Se consigue, así tener de una forma más visual una imagen de todo el proceso. Esto nos permite, poder dividir mejor las tareas, detectar errores en el diseño y poder incorporar cambios de forma sencilla, así como ver el impacto en el resto de la aplicación y su forma de trabajar con ella.

Nuestros mockups se dividen en 3 secciones, móvil, tableta y escritorio. De cada una de ellas, mostraremos las vistas principales e intermedias y la navegación entre ellas.

La estructura en todas ellas está compuesta por las vistas de Portada, Municipios y Estaciones. En el caso, del escritorio añadimos una más, que es el Buscador.

En todos los dispositivos habrá una escala que representará la intensidad de las lluvias. Esta ha sido escogida siguiendo el modelo de la AEMET, aunque ha sido simplificada para los usuarios.

6.2.1. Interfaz para dispositivos móviles

En este apartado presentaremos las vistas para el móvil, divididas en principal e intermedias.

Vista Principal

En la vista principal de la aplicación web se pueden seleccionar dos formas de ver la información, tal como se muestra en la Figura 6.7.

En la primera que será la mostrada por defecto, se verá una imagen de la isla de Gran Canaria, en diferentes colores según la intensidad de la lluvia caída (Ver Figura 6.7 (a)). En ella, se podrá seleccionar cada zona de la misma mostrando un panel con la precipitación media.

La segunda vista mostrada en las Figura 6.7 (b y c), será una tabla donde se mostrarán los diferentes municipios o estaciones según se elija. Éstos, estarán ordenados de forma decreciente según la cantidad de la precipitación media, con su nombre y el color de la intensidad.

Por último, mostramos el menú principal (Ver Figura 6.7(d)), donde podemos seleccionar las dos vistas, el intervalo de tiempo de los datos, que estarán divididos en 1, 7 y 30 días, y botones para compartir los mismos mediante Facebook, Twitter o por correo electrónico.



Figura 6.7 Vista principal – Mapa y Tabla – Móvil

Vistas Intermedias

Si se selecciona un municipio se mostrará una vista intermedia (Ver Figura 6.8) donde se encontrará el nombre y la localización en la isla, además de una tabla con las estaciones pertenecientes a éste, ordenadas de forma decreciente en base a su precipitación media, junto con su nombre e intensidad. También se podrá ver las precipitaciones medias del mismo representados por una gráfica.

Si se selecciona una estación en particular, se verá el nombre de ésta, y sus datos, mostrados en una tabla o en una gráfica según el intervalo que haya elegido el usuario.



Figura 6.8 Vistas intermedias – Municipios y Estación – Móvil

Flujo de trabajo

En la Figura 6.9 mostramos la navegación básica entre las diferentes vistas.

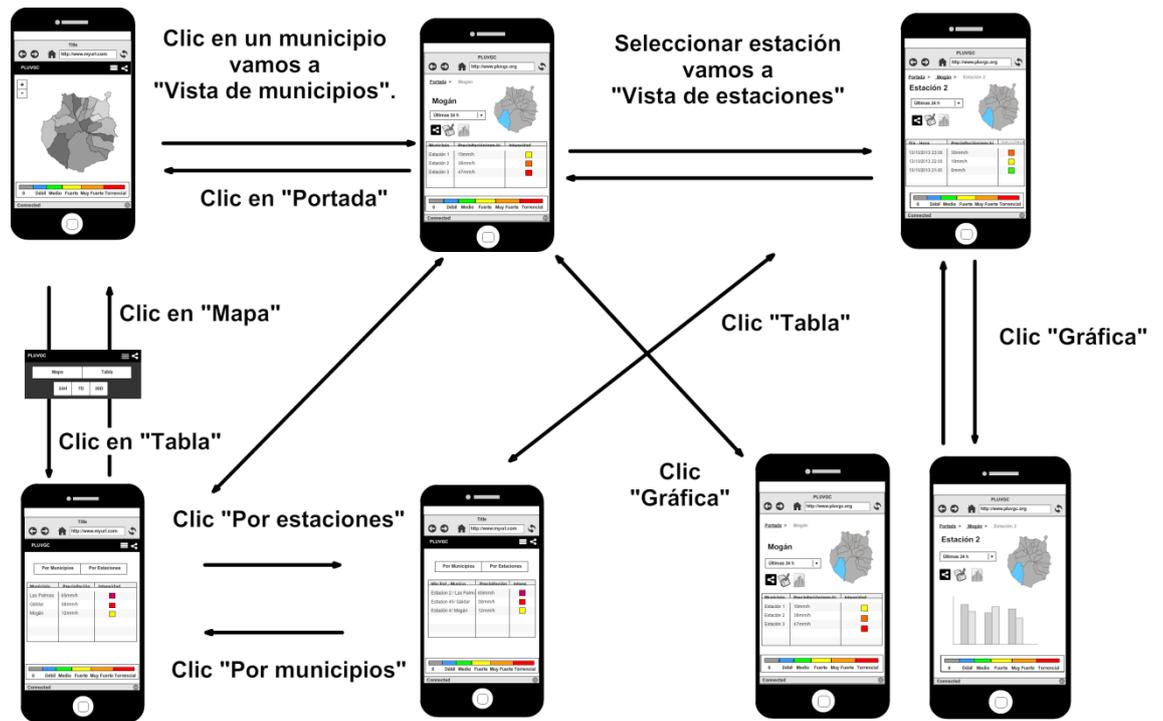


Figura 6.9 Diagrama de flujo – Móvil

6.2.2. Interfaz para tabletas

En este apartado presentaremos las vistas para la tableta, divididas en principal e intermedias.

Vista Principal

La vista principal de la tableta, que se muestra en la Figura 6.10, se estructura de la misma forma que la del móvil.

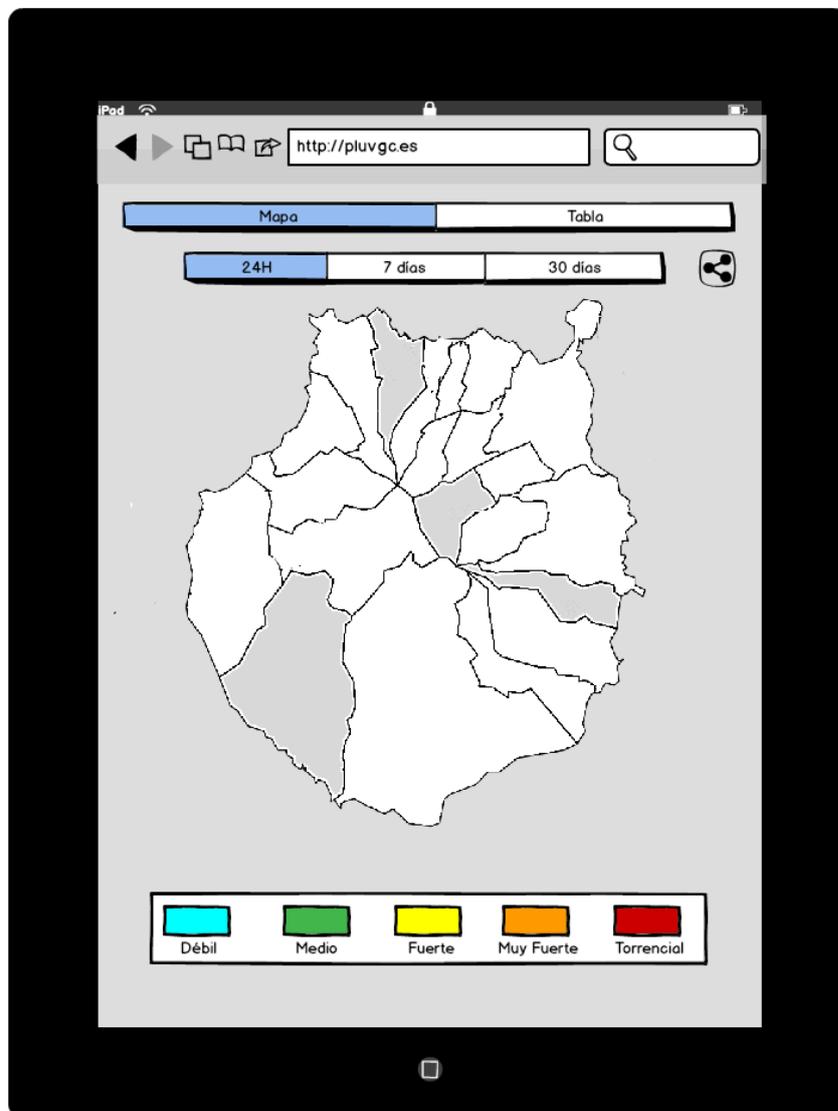


Figura 6.10 Vista principal – Modo Mapa – Tablet

En la vista de la tabla de datos (Ver Figura 6.11), no hay cambios con respecto a la del móvil en cuanto a los elementos a mostrar, ya que son los mismos pero con una distribución diferente.



Figura 6.11 Vista principal – Modo Tabla – Tablet

Vistas Intermedias

Cuando seleccionamos algún municipio se muestra en la Figura 6.12 el nombre y la zona en el mapa, del municipio seleccionado, así como sus estaciones asociadas. Además, podemos ver una gráfica de las últimas precipitaciones medias de ese municipio.

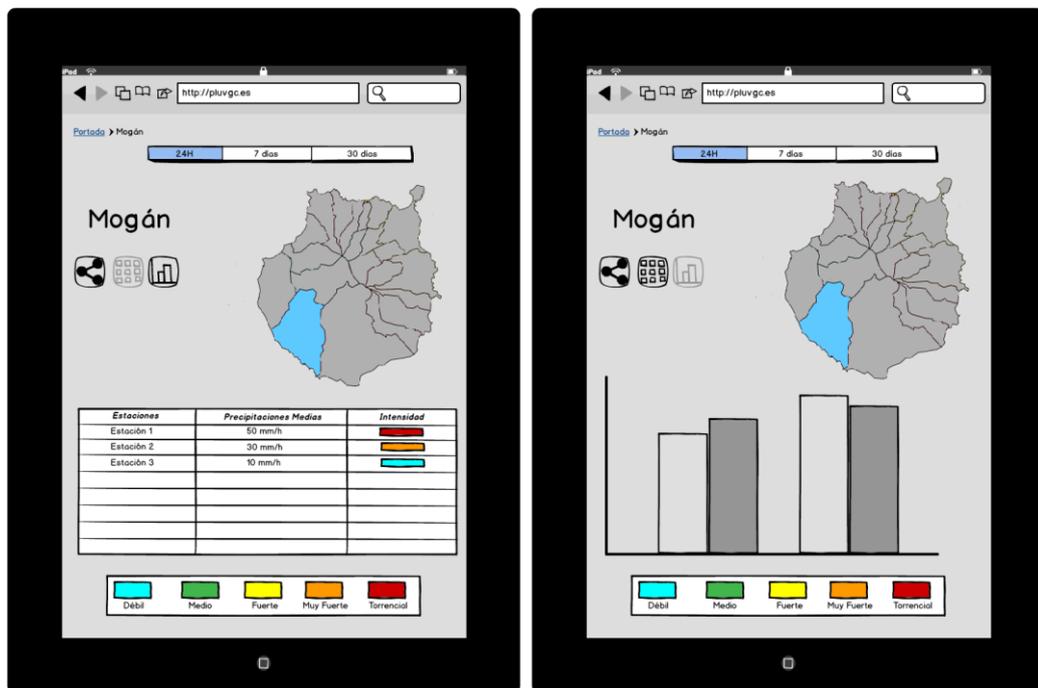


Figura 6.12 Vista intermedia – Municipios – Tablet

Si se selecciona alguna estación se mostrará en la Figura 6.13 sus últimas precipitaciones recogidas ordenadas de forma decreciente según la fecha, estos datos se podrán ver mediante una tabla o una gráfica.

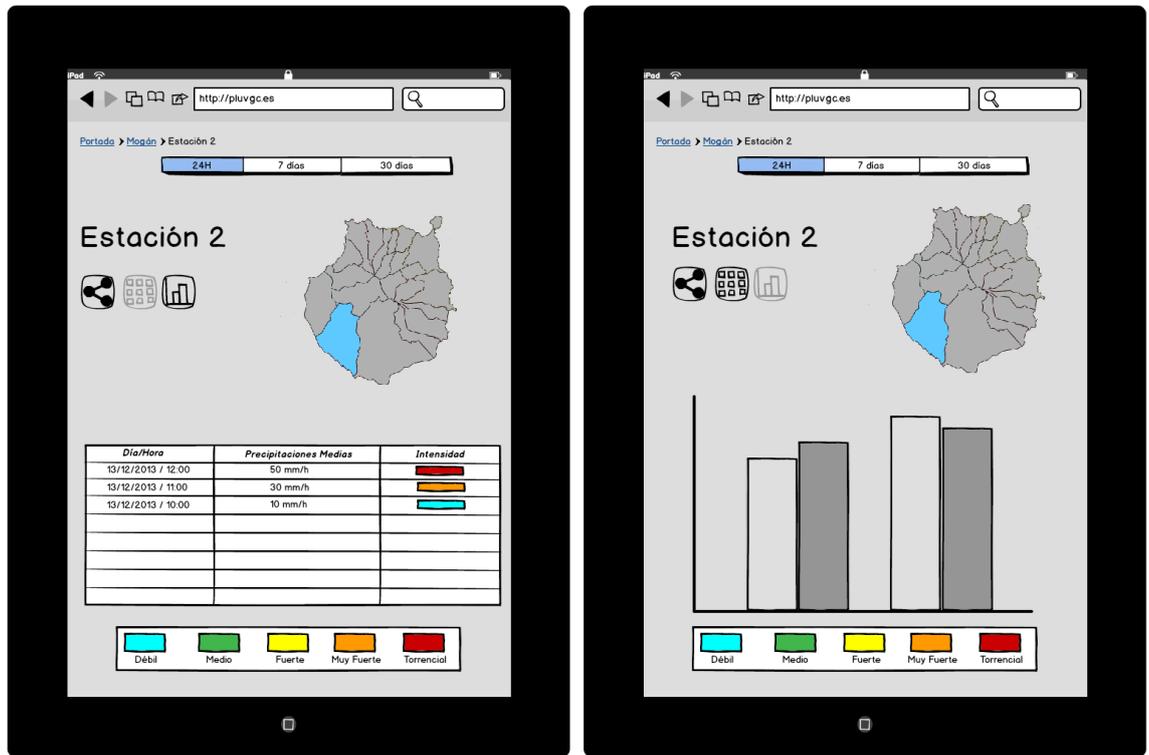


Figura 6.13 Vista de las estaciones – Estaciones – Tablet

Flujo de trabajo

En la Figura 6.14 mostramos la navegación básica entre las diferentes vistas.

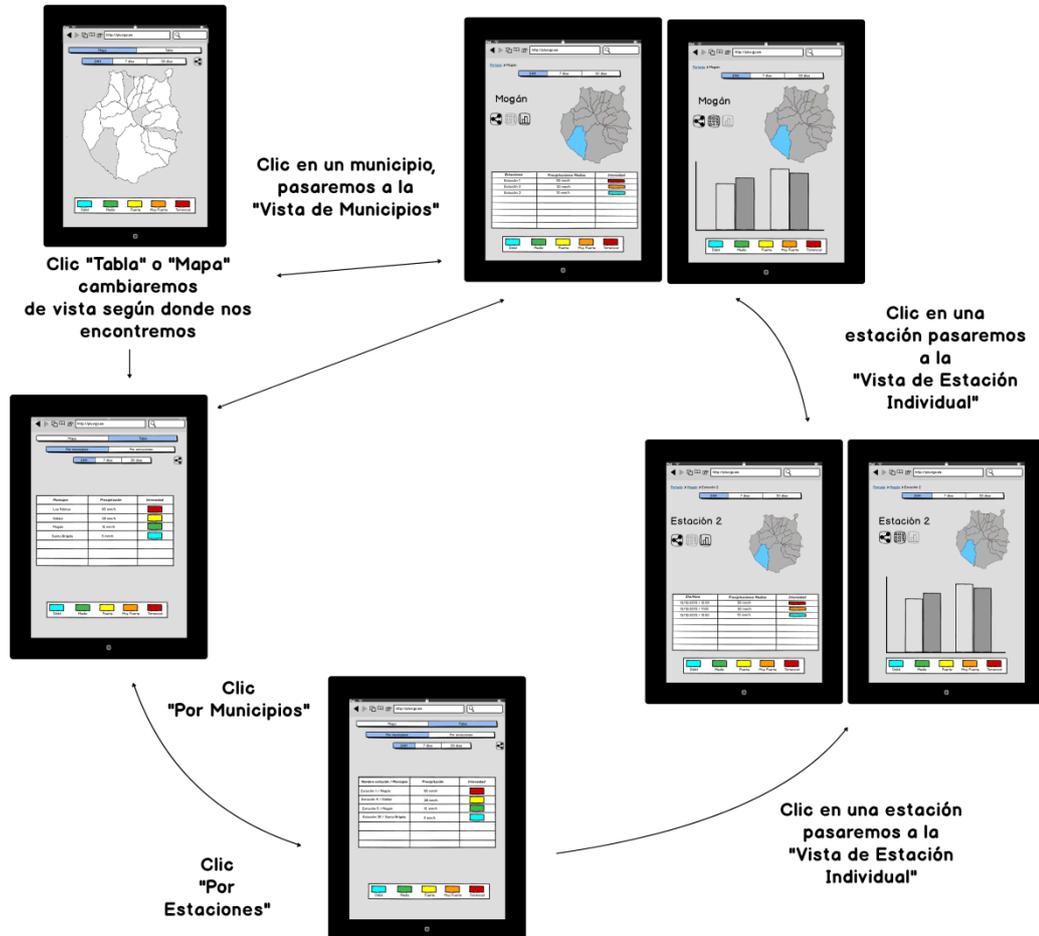


Figura 6.14 Diagrama del flujo de trabajo – Tablet

6.2.3. Interfaz para escritorio

En este apartado presentaremos las vistas para el escritorio, divididas en principal e intermedias.

Vista principal

En la Figura 6.15 para la interfaz del escritorio mostramos tanto la tabla como el mapa para tener una visión global de los datos de las lluvias. También puede accederse al buscador avanzado mediante el botón que aparece en la interfaz para buscar datos más precisos.

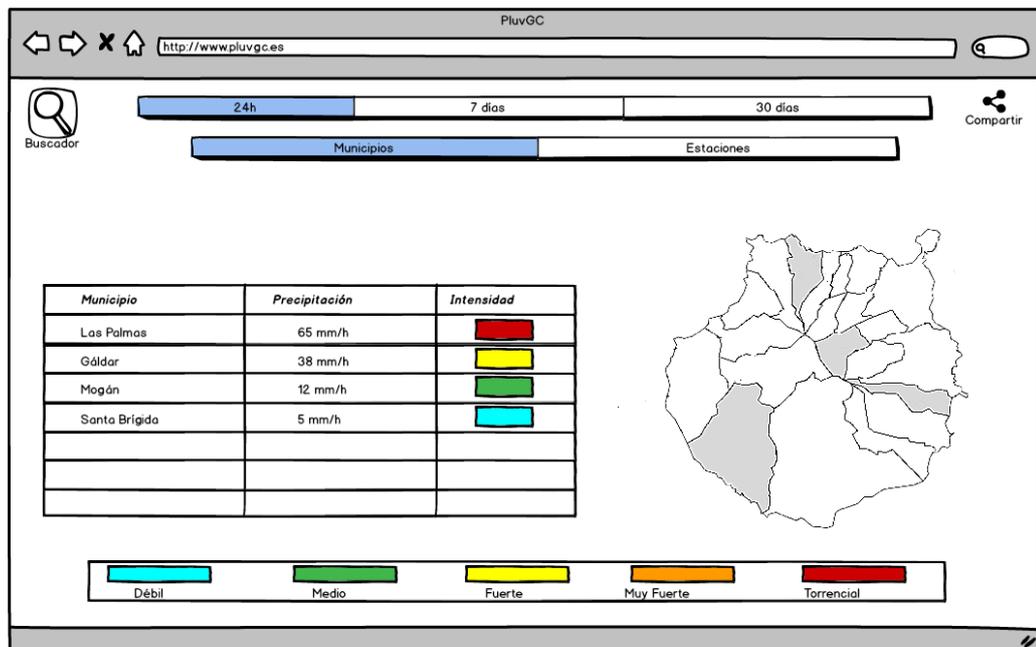


Figura 6.15 Vista principal – Mapa y Tabla – Escritorio

Vistas intermedias

Si le damos al botón del buscador (Ver Figura 6.16) podremos obtener datos de una forma más precisa mediante diferentes filtros como son las fechas, municipio, estaciones e intensidad de las lluvias.

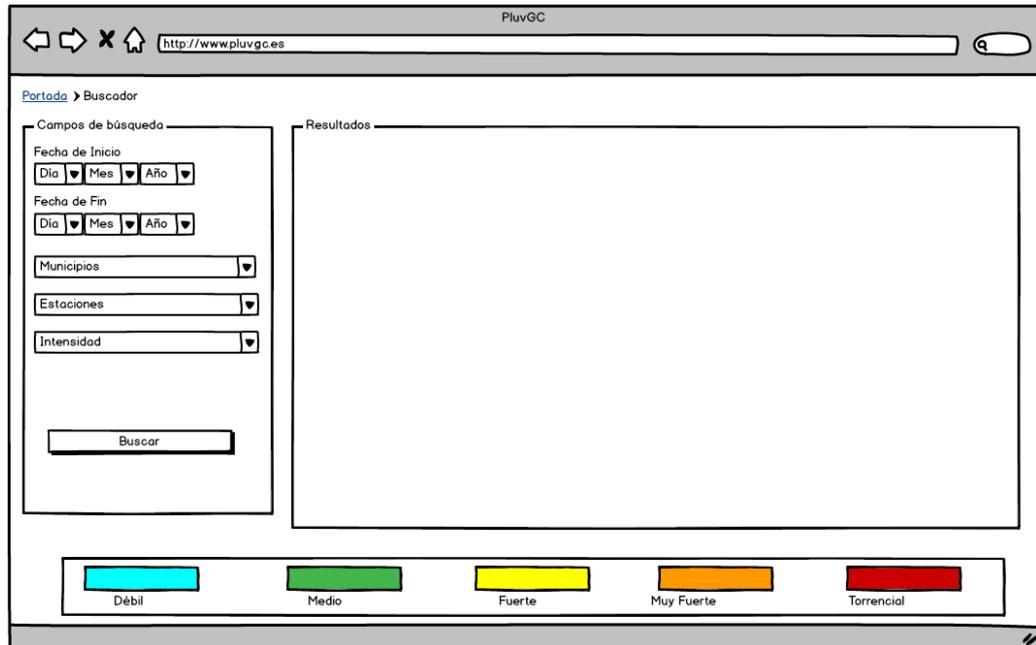


Figura 6.16 Vista intermedia – Buscador – Escritorio

Una vez hayas elegido el municipio se te mostrará una tabla con las estaciones y la zona del mapa a la que corresponde, también se podrá visualizar las últimas lluvias mediante una gráfica. En el caso de la Figura 6.17, se trata en este caso del municipio de Mogán.

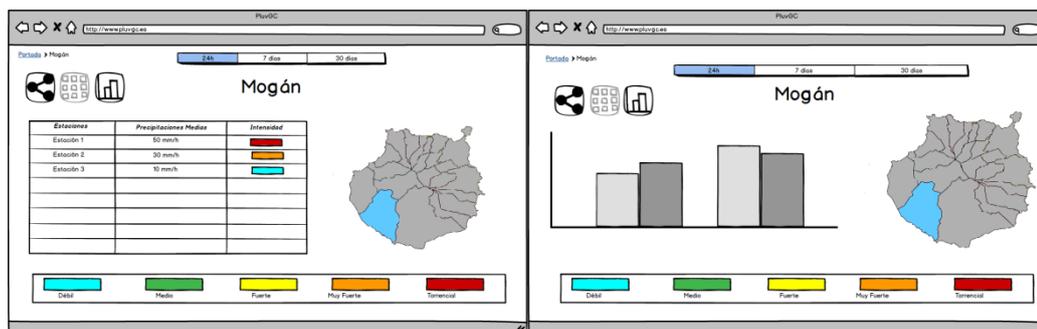


Figura 6.17 Vista Intermedia – Estación – Escritorio

Si se selecciona una estación se mostrará en la Figura 6.18 una tabla con sus últimos datos y su posición en el mapa. Se podrá visualizar también mediante una gráfica de barras.

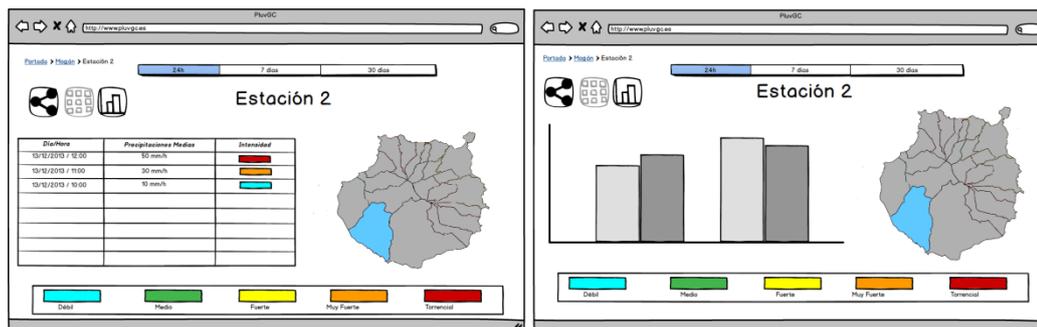


Figura 6.18 Vista intermedia – Municipio – Escritorio

Flujo de trabajo

En la Figura 6.19 mostramos la navegación básica entre las diferentes vistas.

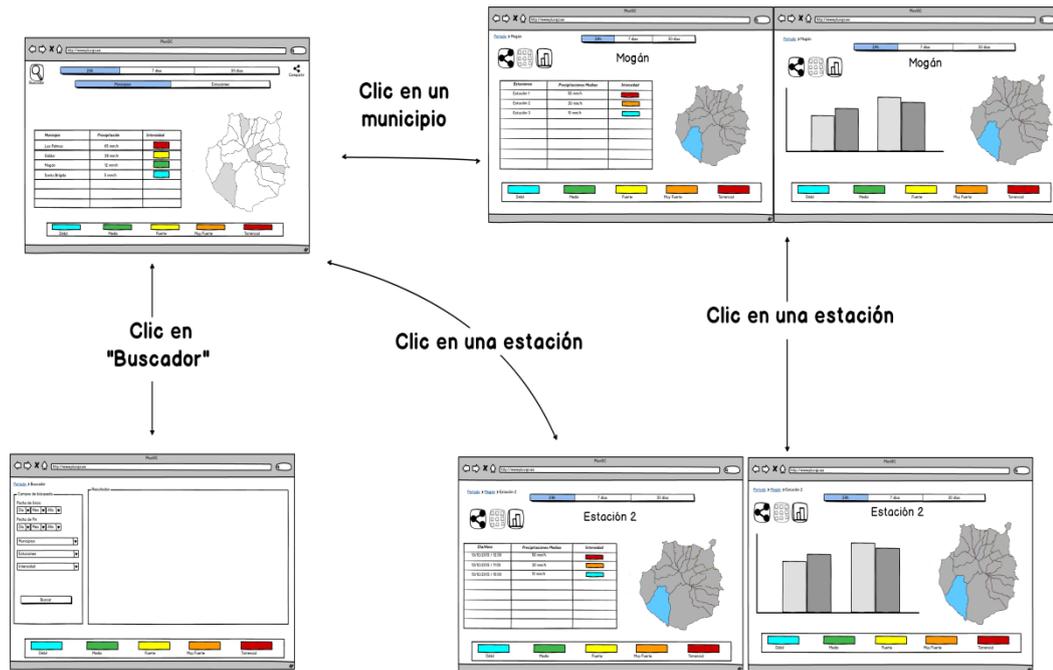


Figura 6.19 Diagrama de flujo – Escritorio

6.3. Requisitos no funcionales

A continuación se detallan los requisitos no funcionales que no están descritos en los casos de uso.

Compatibilidad entre navegadores

La aplicación debe de ser compatible con los navegadores Web más usados del mercado. Para cumplir este requisito, hemos usado el framework Foundation que ha sido diseñado y testeado para ser usado en ellos (Ver Figura 6.20).

Browser/OS	The Grid	Layout/UI	JS
Chrome	✓	✓	✓
Firefox	✓	✓	✓
Safari	✓	✓	✓
IE10	✓	✓	✓
IE11	✓	✓	✓
IE9	✓	✓	✓
IE8	✗	✗	✗
IE7	✗	✗	✗
iOS (iPhone)	✓	✓	✓
iOS (iPad)	✓	✓	✓
Android 2, 4 (Phone)	✓	✓	✓
Android 2, 4 (Tablet)	✓	✓	✓
Windows Phone 7+	✓	✓	✓
Surface	✓	✓	✓

Figura 6.20 Compatibilidad de navegadores

Mantenibilidad

El código fuente debe ser fácilmente legible y mantenible. Con el uso del framework RoR seguimos una serie de principios como son “No te repitas” y “Convención sobre configuración” que te permiten escribir con eficiencia y en la menor cantidad de líneas de código. Además sigue el patrón MVC que explicaremos más adelante en el diseño, que nos permite mantener más ordenado y estructurado el código.

Rendimiento

La aplicación debe atender las peticiones Web con suficiente velocidad para atender a varios usuarios al mismo tiempo.

Usabilidad

La aplicación debe de ser fácil de usar. Con la filosofía “Mobile First”, conseguimos mejorar la usabilidad de la aplicación, centrándonos en los aspectos importantes de la experiencia del usuario en cada dispositivo (móvil, tableta o PC)

Facilidad instalación

El sistema debe ser fácilmente instalable en cualquier plataforma hardware y software.

Integridad información

La aplicación debe validar automáticamente la información que se introduce en el sistema a través de los formularios.

6.4. Modelo de dominio

En la Figura 6.21 se muestran las clases conceptuales identificadas en el proceso de identificación de requisitos, con sus atributos y relaciones. Este modelo sirve para especificar los requerimientos de la aplicación Web.

En nuestro caso, poseemos tres elementos importantes, que son los municipios, las estaciones y las medidas. Y sus relaciones se expresarían de la siguiente manera, cada municipio tiene una cantidad de estaciones que miden diariamente la lluvia caída

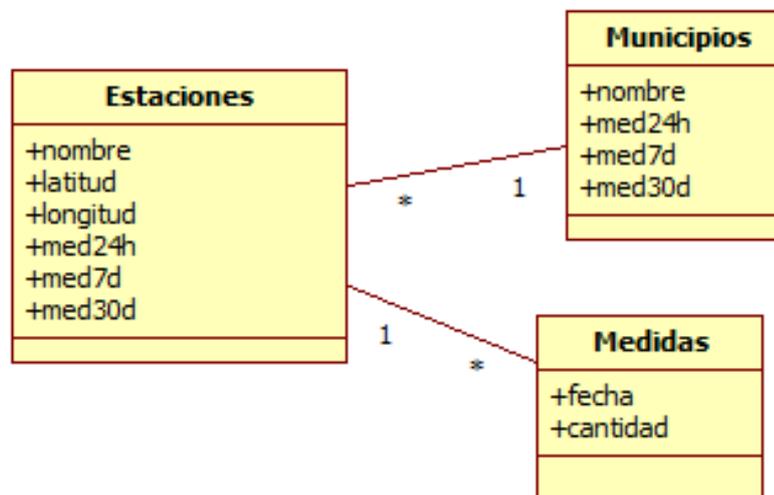


Figura 6.21 Modelo del dominio – PLUVGC

7. Diseño

Una vez obtenida toda la información en la fase de análisis, procederemos en el diseño a modelar la estrategia a seguir para la elaboración de la aplicación web para que soporte todos los requisitos, funcionales y no funcionales.

7.1. Diagrama de despliegue

Nuestra arquitectura estará formada por un servidor web que contendrá nuestra aplicación y base de datos, además de un servidor de correo, que se usará para enviar por email la información pedida por los usuarios.

Los usuarios se conectarán con sus dispositivos (móvil, tableta y/o escritorio) desde el navegador web de éstos, mediante el protocolo HTTP, ya que no se transmiten datos sensibles en la aplicación (Ver Figura 7.1).

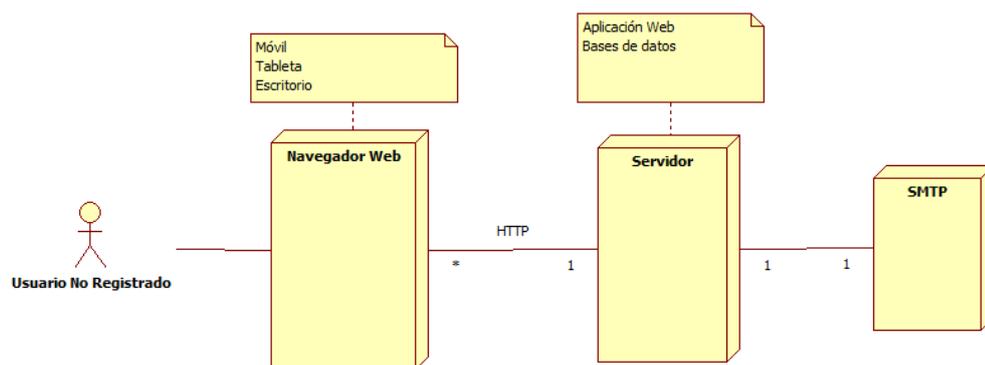


Figura 7.1 Diagrama de despliegue – PLUVGC

7.2. Patrón de diseño – Modelo Vista Controlador (MVC)

Utilizaremos el patrón de diseño Modelo Vista Controlador (MVC) [9], que es un patrón de diseño en la arquitectura del software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Este patrón es utilizado frecuentemente en aplicaciones web, donde la vista es la página HTML, el modelo es la base de datos y el controlador es el encargado de interactuar con ambos para poder generar las vistas con los datos pedidos y le sea devuelta al cliente (Ver Figura 7.2).

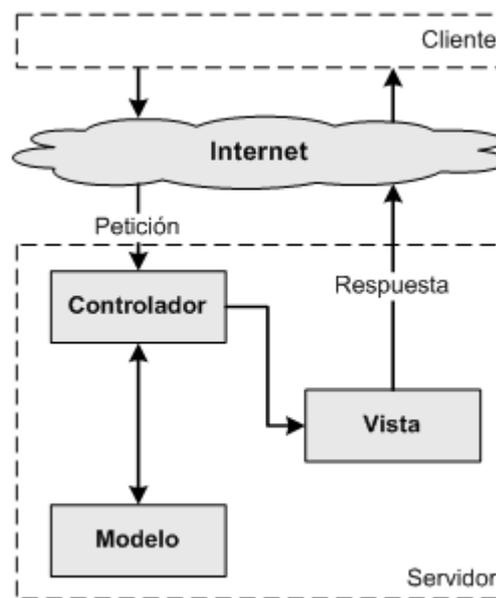


Figura 7.2 Esquema del patrón MVC

7.3. Ruby on Rails

Para el desarrollo de nuestra aplicación web se ha decidido usar el framework Ruby on Rails [\[10\]](#), que permite la creación de aplicaciones de una forma simple y elegante, sigue como paradigma el patrón de diseño que comentamos anteriormente MVC y está escrito en el lenguaje de programación Ruby.

Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración.

Además, Rails proporciona una capa de abstracción ORM (Object-Relational-Mapping) a la base de datos, que permite acceder a los registros de las tablas de la base de datos como si fueran objetos, independientemente del sistema de gestión de base de datos utilizado. Finalmente, Rails incluye una serie de herramientas, como los generadores que nos crean estructuras básicas de desarrollo en segundos.

En Rails se usa por defecto el enfoque REST, que es un estilo de arquitectura de software que consiste en directrices y mejores prácticas para crear servicios web escalables.

El uso por defecto de REST es debido a las siguientes ventajas que nos proporciona para cumplir y seguir la filosofía de Rails:

URLs limpias

En REST las URLs representan recursos y no acciones, por lo tanto siempre tienen el mismo formato: primero aparece el controlador y luego el identificador de recurso. La manipulación requerida es independiente de la URL y se expresa con la ayuda de los verbos HTTP.

Formatos de respuesta variados

Los controladores REST están escritos de manera que las acciones pueden devolver sus resultados fácilmente en diferentes formatos de respuesta. Una misma acción puede entregar HTML, XML, RSS o cualquier otro formato de datos según los requisitos de la aplicación cliente de manera simple y sencilla.

Menos código

El desarrollar acciones únicas capaces de soportar múltiples clientes distintos evita repeticiones en el sentido DRY (Don't Repeat Yourself) y da como resultado que los controladores tengan menos código.

Controladores orientados a CRUD

Los controladores y los recursos se funden en una única cosa - cada controlador tiene como responsabilidad manipular un único tipo de recurso. CRUD viene de que los objetos responden a 4 primitivas: Create, Read, Update, Delete.

Diseño limpio de la aplicación

El desarrollo REST produce un diseño de aplicación conceptualmente claro y más fácil de mantener.

A continuación, mostramos cómo funciona el patrón MVC usado Ruby on Rails (Ver Figura 7.3):

1. El usuario hace una petición mediante la vista, que en el caso del ejemplo es "jobs/12".
2. El controlador recoge la petición del usuario y ejecuta la acción asociada que es "show()"
3. Se busca elemento 12 en el modelo mediante la función "find" y se devuelve al controlador.
4. Se rellena la vista asociada con estos datos y se sirve al usuario.

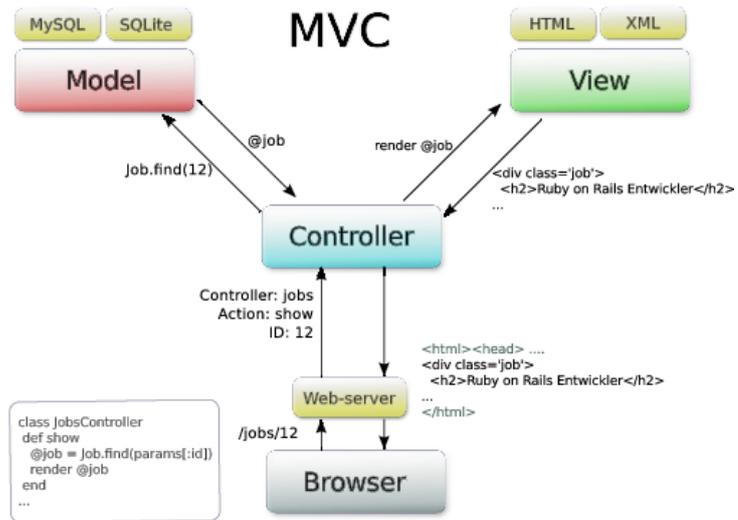


Figura 7.3 Diagrama de funcionamiento MVC – Rails

7.4. Diagramas de clases de diseño

A continuación se mostrarán los diagramas de secuencia de la realización de casos de uso de la aplicación web.

Ver mapa

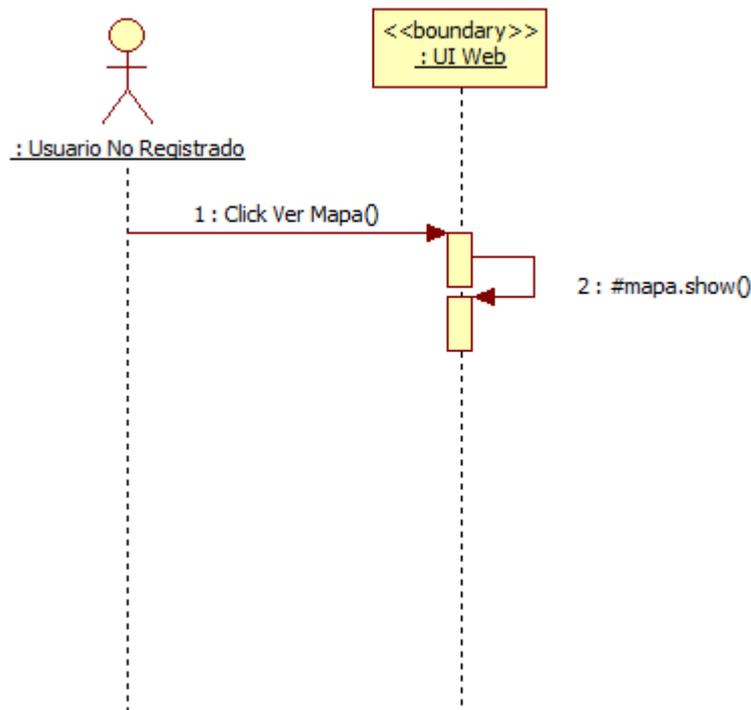


Figura 7.4 Diagrama de interacción – Ver mapa

Ver tabla

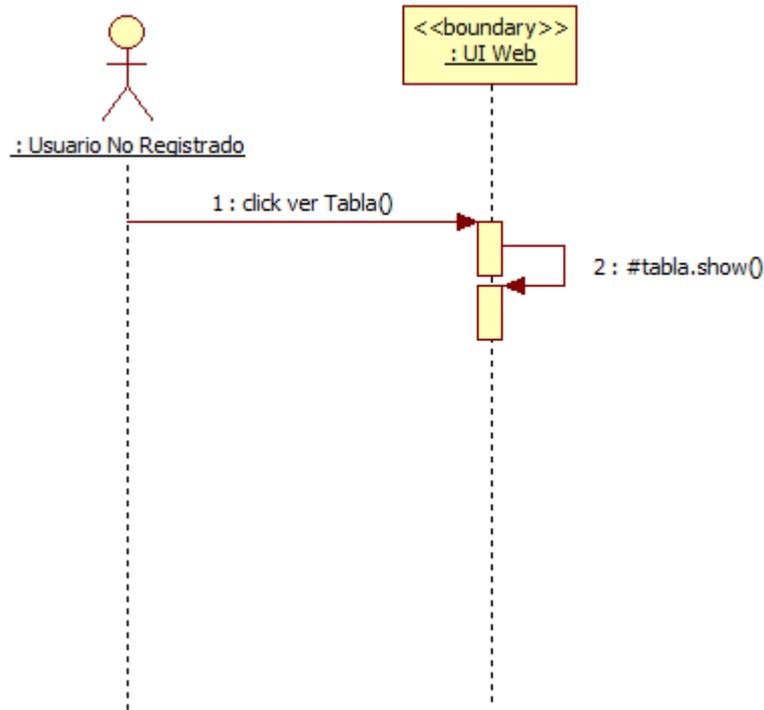


Figura 7.5 Diagrama de interacción – Ver tabla

Ver gráfica

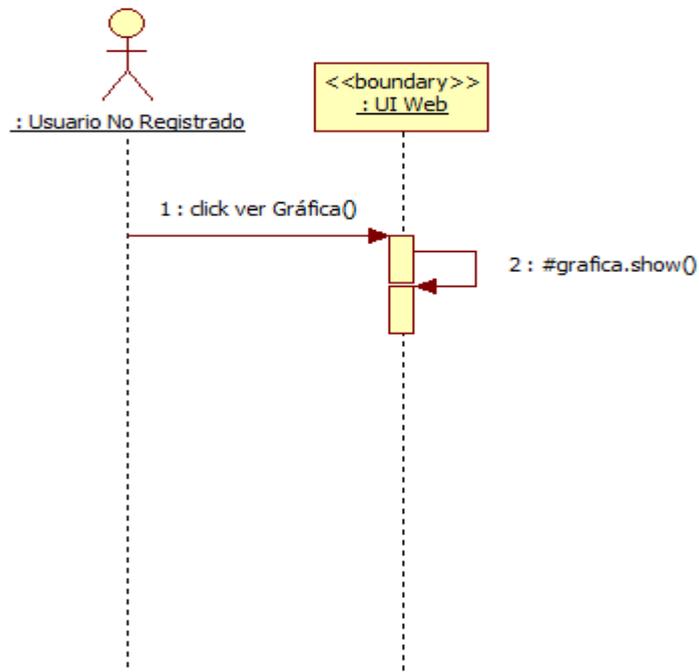


Figura 7.6 Diagrama de interacción – Ver gráfica

Filtrar por municipios

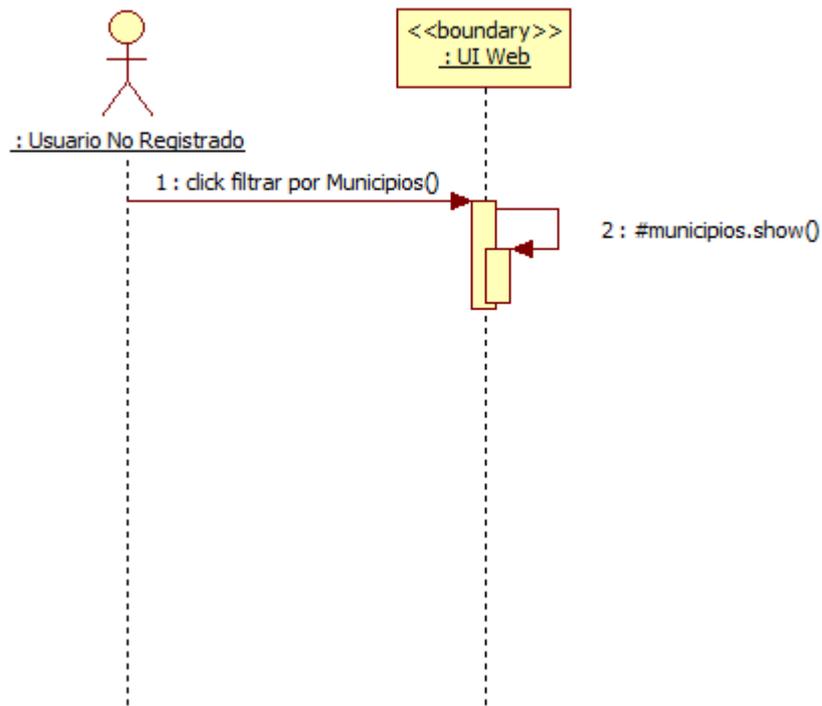


Figura 7.7 Diagrama de interacción – Filtrar por municipios

Filtrar por estaciones

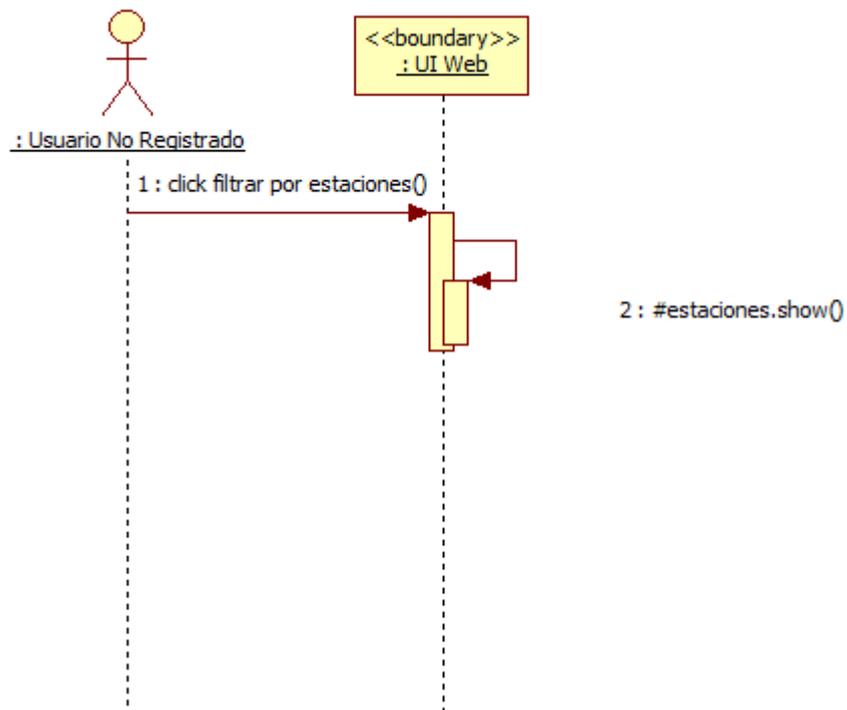


Figura 7.8 Diagrama de interacción – Filtrar por estaciones

Seleccionar municipio

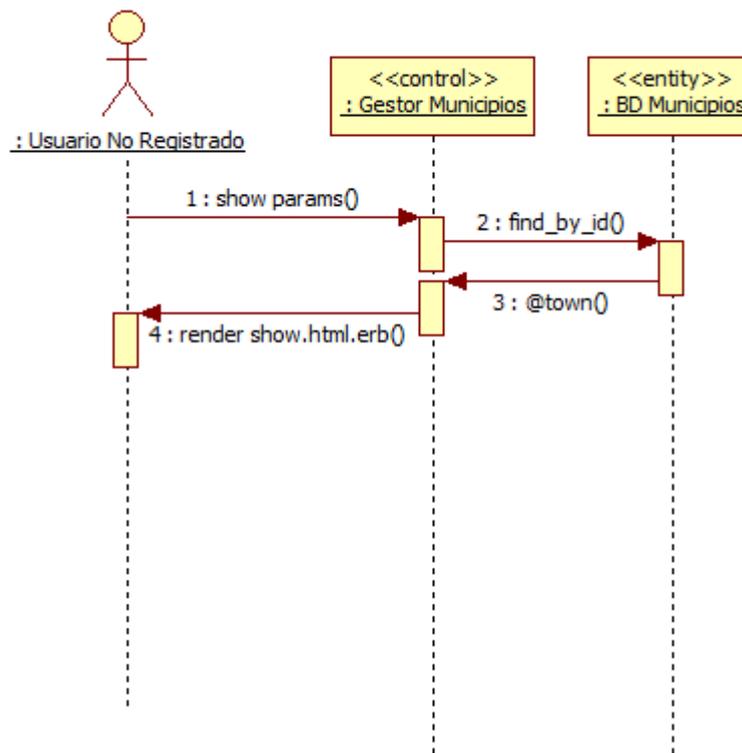


Figura 7.9 Diagrama de interacción – Seleccionar municipio

Seleccionar estación

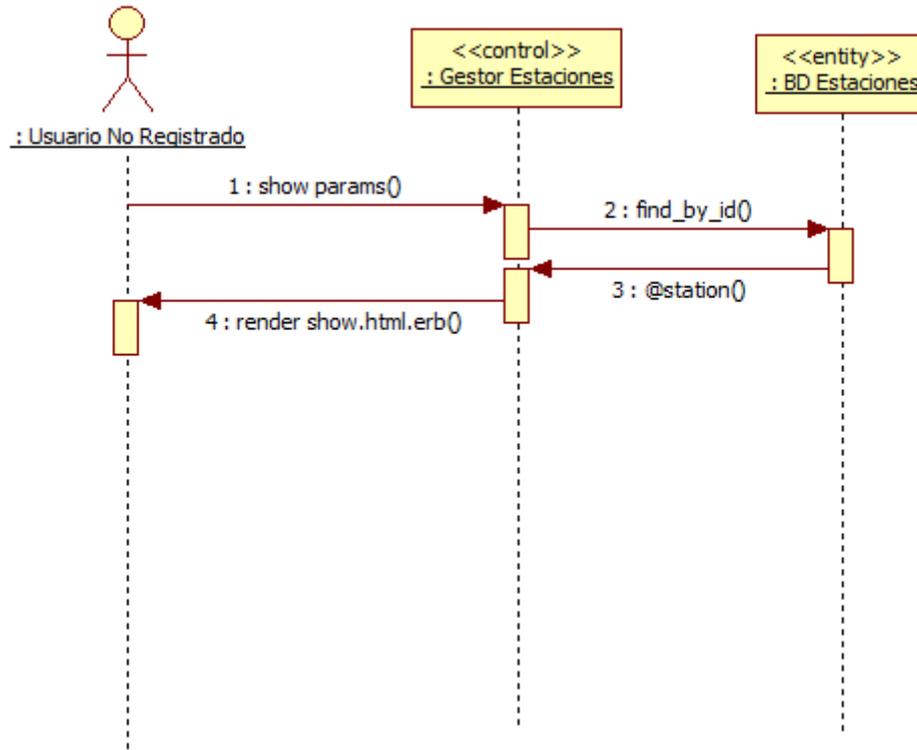


Figura 7.10 Diagrama de interacción – Seleccionar estación

Buscador

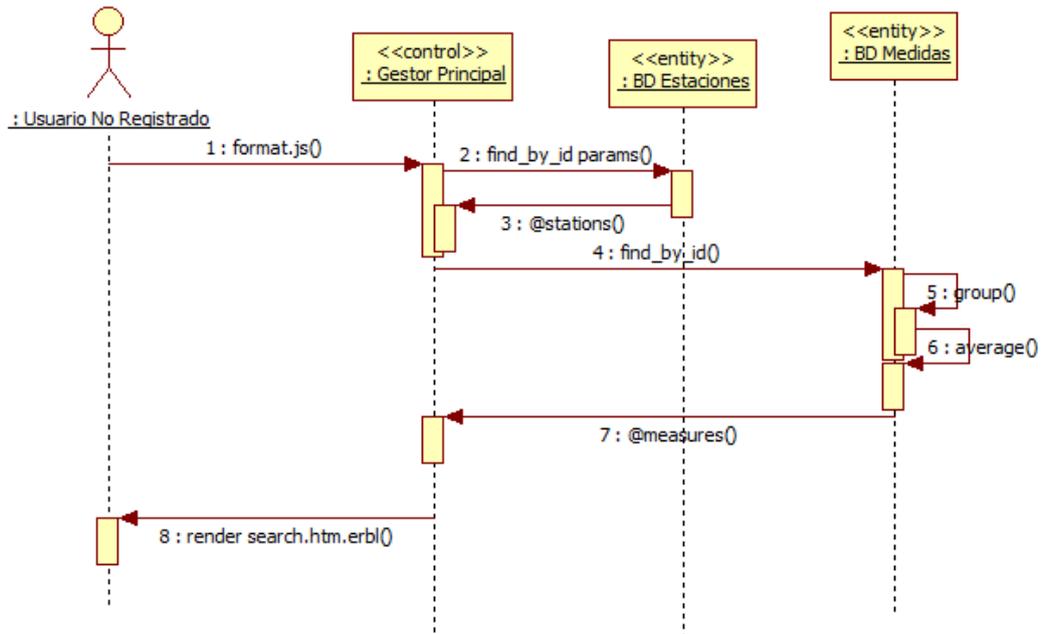


Figura 7.11 Diagrama de interacción – Buscador

Compartir – Redes Sociales

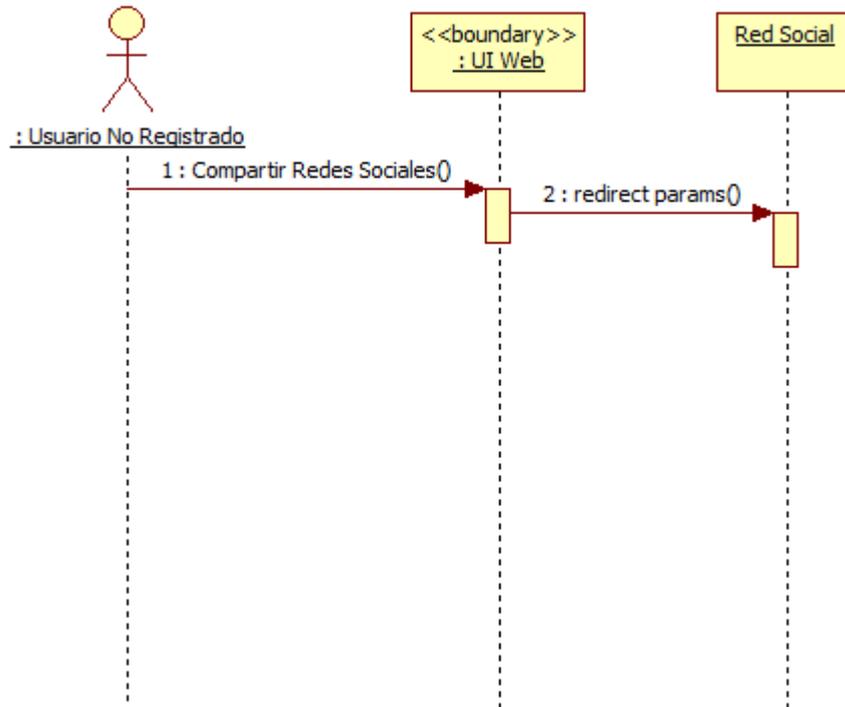


Figura 7.12 Diagrama de interacción – Compartir – Redes sociales

Compartir - Email

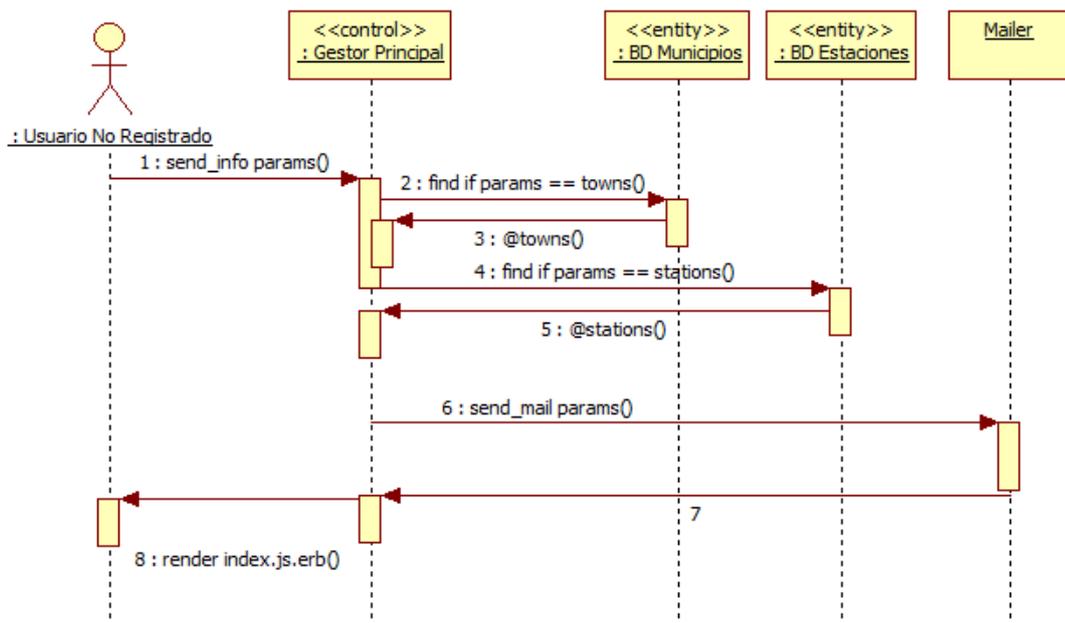


Figura 7.13 Diagrama de interacción – Compartir – Email

7.5. Modelo de datos

Una vez identificado los actores y los casos de uso de la aplicación web hemos creado la base de datos donde guardaremos y manejaremos los datos de las precipitaciones de cada una de las estaciones de la isla de Gran Canaria.

A continuación, se detallan cada una de las tablas de la base de datos y sus campos.

Municipios

La tabla llamada “*Towns*” guardará las medias de cada municipio en los intervalos de 1, 7 y 30 últimos días.

Estaciones pluviométricas

La tabla llamada “*Stations*” almacenará los nombres de cada una de las estaciones repartidas por toda la isla de Gran Canaria junto a su posición geográfica y el municipio al que pertenece junto con la media de sus precipitaciones en los intervalos de 1, 7 y 30 últimos días.

Medidas

La tabla llamada “*Measures*” tendrá los datos de las precipitaciones de todas las estaciones de la isla de Gran Canaria, con sus precipitaciones diarias.

En la Figura 7.14 mostramos el diagrama de la base de datos compuesto por las tablas anteriormente descritas.

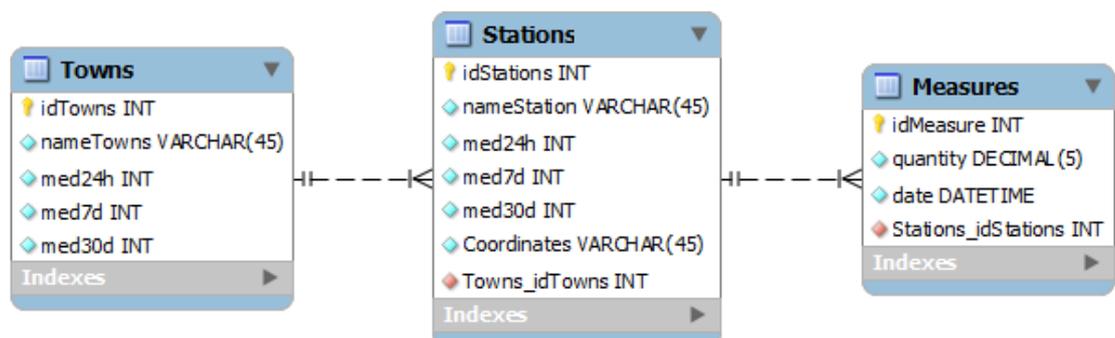


Figura 7.14 Diagrama de la base de datos

8. Desarrollo

En este apartado se explicará los detalles importantes del desarrollo de la aplicación web, hablando de la estructura de un proyecto en Rails, además de las librerías y lenguajes que se han usado en el front-end y back-end de la aplicación.

8.1. Estructura de ficheros del proyecto Rails

Nuestro proyecto Rails se estructura en estos directorios:

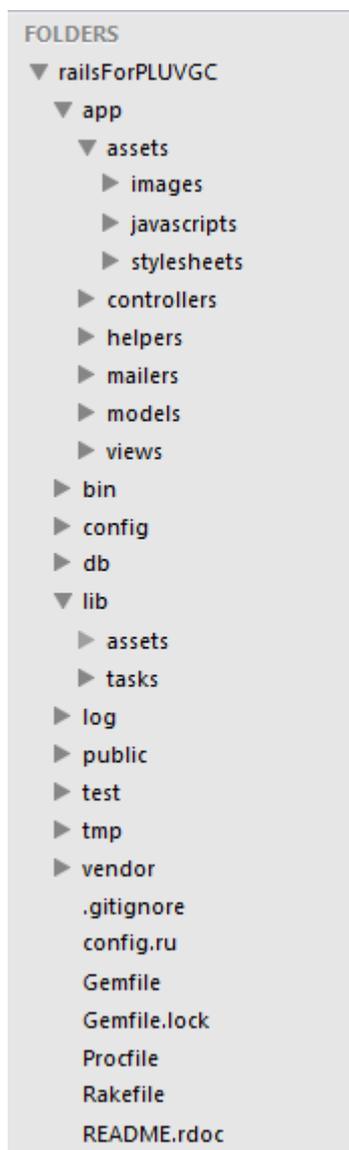


Figura 8.1 Estructura de ficheros del proyecto.

app

En este directorio se organizan los componentes de la aplicación. Tiene una serie de subdirectorios que contienen los controladores, modelos y las vistas que se muestran al usuario. Además de los archivos del lado del cliente.

app/assets

Contiene todos los elementos del lado del cliente como son los archivos JavaScript y CSS e imágenes.

app/controllers

En este directorio es donde se guardan los controladores que se encargan de la lógica de la aplicación.

app/mailers

En este directorio es donde se están los elementos de la aplicación encargados de gestionar los correos.

app/helpers

En este directorio están las clases que asisten a los controladores, modelos y vistas con diversas funciones auxiliares permitiendo dejar más limpio el código de estos.

app/models

En este directorio se encuentran los modelos de la aplicación, que gestionan la información.

app/views

En este directorio están las plantillas que se rellenan con los datos del modelo traídos por el controlador para ser servidos al usuario en HTML.

app/views/common

En este directorio se encuentran todos los elementos HTML que se repiten en las diferentes vistas de la aplicación.

app/views/layouts

En este archivo se encuentra la plantilla principal donde se cargarán los demás archivos .erb.

config

En esta carpeta se encuentra la configuración de su aplicación referente a base de datos, ambiente y el direccionamiento web, entre otras cosas.

db

En este directorio se mantienen los scripts para el manejo de la base de datos relacional.

lib

En esta carpeta se almacenan las bibliotecas o rutinas específicas que se han creado para el funcionamiento de la aplicación.

logs

Aquí van los logs de peticiones y errores de todos los ambientes de Rails (development, production y test).

public

En este directorio tiene todos los archivos que permanecen estáticos.

test

En este directorio se encontrarán todas las pruebas que se han creado para probar el desarrollo creado en Rails.

tmp

En este directorio se encuentran para almacenar archivos temporales para procesamiento inmediato.

Vendor

En este directorio se almacenan bibliotecas provistas por terceras partes.

8.2. Recursos externos

En este apartado nombraremos las librerías externas que se han usado para el desarrollo del proyecto tanto por parte del cliente (Front-End) como del servidor (Back-End).

8.2.1. Cliente – Front-End

En este apartado, mostraremos las herramientas y librerías que hemos usado en la parte del cliente.

Foundation

Es una colección de herramientas para la creación de sitios y aplicaciones web [\[14\]](#) que contiene plantillas de HTML y CSS de diferentes diseños de tipografía, formas, botones, navegación y otros componentes de la interfaz, así como plugins de JavaScript que te facilitan y aumentan la velocidad del desarrollo.

Sigue la filosofía de Mobile First, que explicamos anteriormente en el modelo de interacción.

Leaflet.js

Es una librería JavaScript [\[15\]](#) de open-source ampliamente utilizada para construir mapas interactivos en aplicaciones web, está diseñada para que su uso sea simple, y de alto rendimiento. Funciona de manera eficiente a través de todas las principales plataformas de escritorio y móviles, aprovechando HTML5 y CSS3 en los navegadores modernos sin dejar de ser accesibles en otros más antiguos. Es utilizada por FourSquare, Pinterest y Flickr.

Omnivore.js

Es un plugin [\[16\]](#) para poder convertir los datos de TopoJSON a GeoJSON y sean leídos por la librería que explicamos anteriormente.

TopoJSON.js

Es una extensión de GeoJSON [\[17\]](#) que codifica los datos geométricos en archivos TopoJSON eliminando la redundancia de segmentos en las representaciones.

TableSorter

Es un plugin [\[18\]](#) en JavaScript que te permite ordenar las tablas según un criterio preestablecido, permitiéndote bajar la carga del servidor y delegarla en el cliente.

HighCharts

Es una librería en JavaScript [\[19\]](#) utilizada para poder crear las gráficas con los datos de las precipitaciones.

8.2.2. Servidor – Back-End

En este apartado, mostraremos las herramientas y librerías que hemos usado en la parte del servidor.

PostgreSQL – PG

Es una gema que te permite conectarte a una base de datos de PostgreSQL.

Lazy Highcharts

Es una gema que se usa para poder crear gráficas con la librería Highchart [\[19\]](#) pero definiéndola desde el servidor.

Ransack

Es una gema que te facilita la búsqueda de información en una base de datos mediante una serie de filtros establecidos ahorrándote tiempo.

8.3. Despliegue en Heroku

Heroku[20] es una de las primeras plataformas de computación en la nube, que fue desarrollada en junio de 2007, con el objetivo de soportar solamente el despliegue de aplicaciones web desarrolladas con RoR, pero posteriormente se ha extendido el soporte a Java, Node.js, Scala, Clojure , Python y PHP.

Por eso hemos decidido usar este servicio para el despliegue en producción de nuestra aplicación web, ya que ofrece un plan gratuito, que nos viene muy bien para nuestros propósitos.

Primero una vez registrados en Heroku, accedemos al panel de control, como el que aparece en la Figura 8.2 y seleccionamos Ruby.

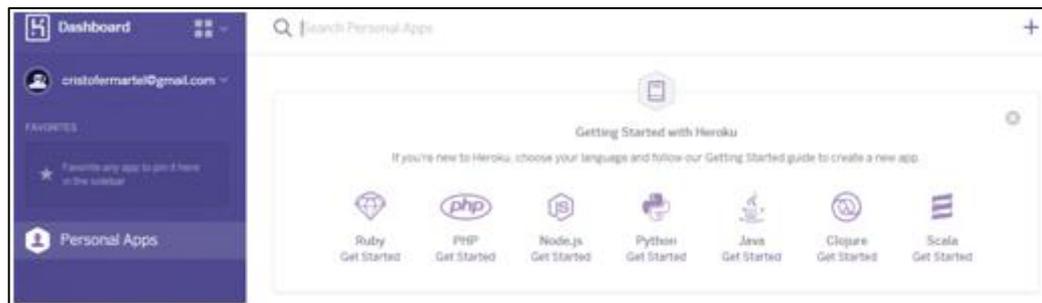
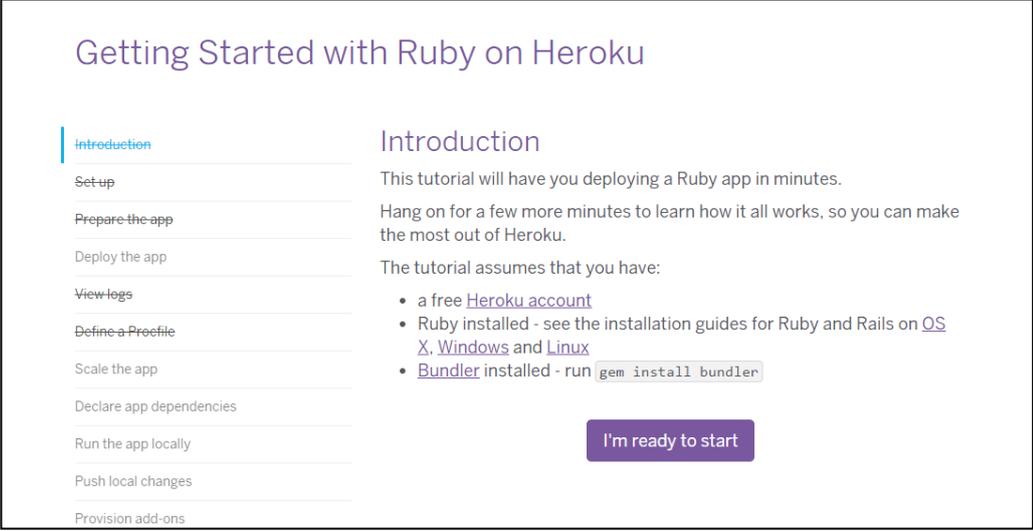


Figura 8.2 Dashboard de Heroku

Nos enviará a una sección donde tendremos que realizar una serie de pasos para subir nuestra aplicación a Heroku (Ver Figura 8.3).



Getting Started with Ruby on Heroku

- Introduction
- Set up
- Prepare the app
- Deploy the app
- View logs
- Define a Prefile
- Scale the app
- Declare app dependencies
- Run the app locally
- Push local changes
- Provision add-ons

Introduction

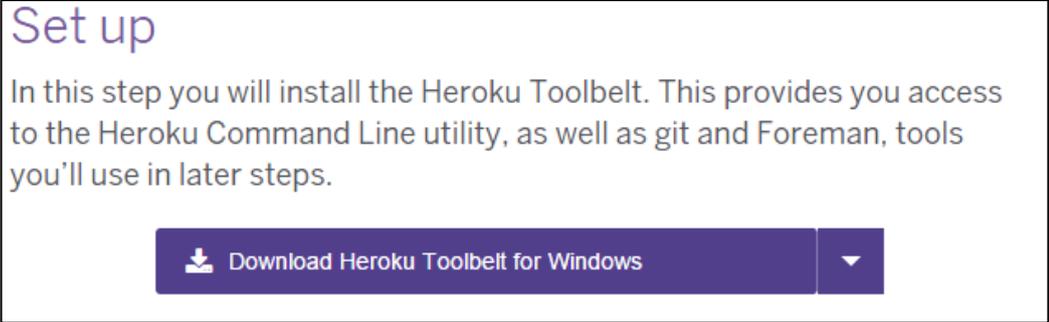
This tutorial will have you deploying a Ruby app in minutes. Hang on for a few more minutes to learn how it all works, so you can make the most out of Heroku. The tutorial assumes that you have:

- a free [Heroku account](#)
- Ruby installed - see the installation guides for Ruby and Rails on [OS X](#), [Windows](#) and [Linux](#)
- [Bundler](#) installed - run `gem install bundler`

[I'm ready to start](#)

Figura 8.3 Primeros pasos para empezar con Ruby en Heroku

Lo primero será instalar Heroku ToolBelt (Ver Figura 8.4), que nos proporcionará acceso a los comandos de Heroku que usaremos más adelante en la consola.



Set up

In this step you will install the Heroku Toolbelt. This provides you access to the Heroku Command Line utility, as well as git and Foreman, tools you'll use in later steps.

[Download Heroku Toolbelt for Windows](#)

Figura 8.4 Descarga del ToolBelt de Heroku

Una vez instalado el ejecutable de Heroku, nos ponemos en nuestro directorio de la aplicación y ponemos el siguiente comando (Ver Figura 8.5), para crear la aplicación de Rails en nuestra cuenta y nos asocien los recursos necesarios para la misma.



Figura 8.5 Creación de los recursos en Heroku

Antes de subir nuestro código a Heroku, tendremos que añadir una serie de archivos y gemas para su correcto funcionamiento. Lo primero de todo será cambiar el servidor web que trae Rails por defecto, ya que para nuestra aplicación en producción, hay que usar un servidor web más potente.

Por recomendación de Heroku, vamos a instalar Puma que es un servidor web para Rails en entornos de producción. Para ello, agregamos la siguiente línea al archivo de gemas '*gem puma*' y añadiremos su configuración básica (Ver Figura 8.6), en el fichero de /config/puma.rb. Esta configuración se encuentra en la [página de configuración de Heroku](#).

```
workers Integer(ENV['WEB_CONCURRENCY'] || 2)
threads_count = Integer(ENV['MAX_THREADS'] || 5)
threads threads_count, threads_count

preload_app!

rackup      DefaultRackup
port        ENV['PORT']      || 3000
environment ENV['RACK_ENV']  || 'development'

on_worker_boot do
  # Worker specific setup for Rails 4.1+
  # See: https://devcenter.heroku.com/articles/deploying-rails-applications-with-heroku-cap
  ActiveRecord::Base.establish_connection
end
```

Figura 8.6 Configuración básica de Puma

El siguiente paso será añadir dos gemas más para que Rails pueda servir archivos estáticos como por ejemplo, imágenes, archivos JS y CSS. Esto es debido, a que en producción se suele usar un servidor proxy en medio de la aplicación de Rails para balancear la carga y servir archivos estáticos. Cuando no lo tiene, Rails envía por defecto un error 404 y la web se vería sin estilo aplicado.

Por tanto, añadimos estas dos líneas (Ver Figura 8.7) al archivo Gemfile para solucionar el problema.

```
#Problema en Heroku con el assets en rails 4
gem 'rails_serve_static_assets'
gem 'rails_stdout_logging'
```

Figura 8.7 Gemas para los archivos estáticos.

Lo siguiente será un fichero llamado “Procfile”, que sirve para especificar el comando que se tiene que ejecutar una vez se inicie la aplicación. Este archivo va colocado en el directorio raíz. El comando que vamos a colocar es el que vemos en la Figura 8.8, que es para levantar el servidor.

```
web: bundle exec puma -C config/puma.rb
```

Figura 8.8 Comando a especificar en el archivo Procfile

Y por último, solo tenemos que subir nuestra aplicación a Heroku mediante git, como se puede ver en la Figura 8.9, hacemos un push en el repositorio de Heroku, que se creó anteriormente, con el comando “*heroku create*”.

```
Now deploy your code:

$ git push heroku master
Fetching repository, done.
Counting objects: 10, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 876 bytes | 0 bytes/s, done.
Total 6 (delta 4), reused 0 (delta 0)
-----> Ruby app detected
```

Figura 8.9 Deploy en Heroku

Después de esto, ya se había completado la subida del código en la plataforma. Como se ve en la Figura 8.10 se puede apreciar que te va guardando las versiones y te permite volver a un estado anterior de una manera muy sencilla.

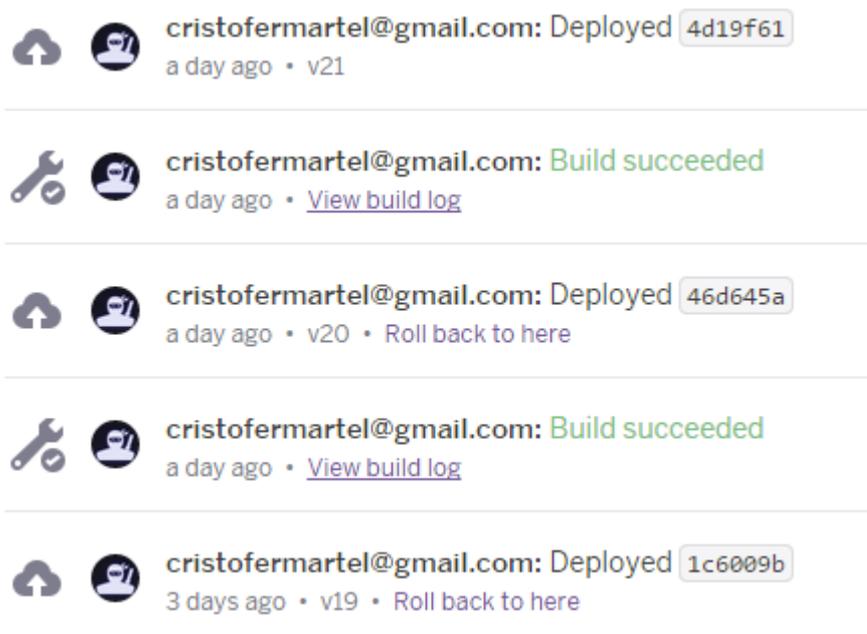


Figura 8.10 Estados de Heroku

Antes de comprobar si la aplicación funciona correctamente tendremos que inicializarla y cargarla de datos. Hemos creado una serie de tareas “rake” que realizarán esta inicialización.

Lo primero que haremos será reiniciar la base de datos y dejarla a 0, antes de empezar a meter datos. Por tanto, aplicamos el siguiente comando (Ver

Figura 8.11) para realizar está acción en el servidor de Heroku.

```
$ heroku run rake db:reset
```

Figura 8.11 Comando para poner a 0 la base de datos

Estos dos primeros comandos (Ver Figura 8.12) solo se ejecutarían la primera vez antes de poner en producción la aplicación. Estos se encargan de inicializar e importar la tabla de municipios y estaciones de la isla de Gran Canaria desde un archivo .csv

```
$ heroku run rake initialize_towns:initialize  
$ heroku run rake import_stations_csv:import
```

Figura 8.12 comandos para inicializar e importar las estaciones y municipios

Con el siguiente (Ver Figura 8.13), importamos las medidas de las precipitaciones desde otro archivo .csv

```
$ heroku run rake import_measures_csv:import
```

Figura 8.13 Comando para importar las medidas

Las dos siguientes tareas “rake” (Ver Figura 8.14), son las encargadas de realizar las medias de las precipitaciones de los municipios y de las estaciones, que serán los datos a mostrar a los usuarios.

```
$ heroku run rake calculate_mean_stations:calculate  
$ heroku run rake calculate_mean_towns:calculate
```

Figura 8.14 Comandos para calcular las medias

Por último, nos faltaría volcar esos datos al mapa, que es lo que hacemos con el comando que nos queda (Ver Figura 8.15). Accedemos al fichero del mapa y cambiamos sus atributos según sus valores.

```
$ heroku run rake update_file_topojson:update
```

Figura 8.15 Comando para actualizar el mapa

Una vez hecho esto, está todo listo para ejecutar nuestra aplicación. Para comprobar que todo funciona correctamente ejecutamos en el terminal, el siguiente comando (Ver Figura 8.16).

```
$ heroku open
Opening pluvgc... done
```

Figura 8.16 Comando para ver la aplicación en producción de Heroku

Y se nos abrirá nuestro navegador con la aplicación en producción, como se puede verse en la Figura 8.17

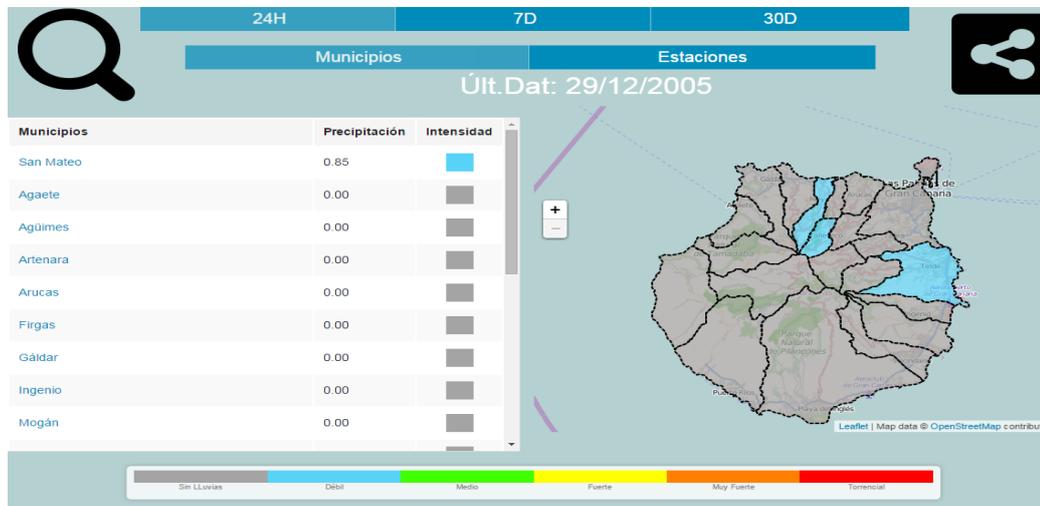


Figura 8.17 Aplicación web en producción

9. Pruebas

En nuestro proyecto hemos realizado una serie de pruebas tanto manuales como automatizadas, probando la integridad de los controladores, modelos y vistas, que hemos creado.

A lo largo del desarrollo se realizaron pruebas manuales de caja de negra comprobando los resultados con diferentes entradas y escenarios.

Para las pruebas automatizadas se hizo uso del framework que viene por defecto en Rails, que se llama *Minitest*, nos provee de una gran cantidad de complementos para testear nuestra aplicación web.

Se realizaron pruebas unitarias y funcionales comprobando cada uno de los componentes de la aplicación. Gracias a estos tests, pudimos descubrir varios errores y casos por contemplar en algunos controladores y modelos, mejorando así el funcionamiento de estos.

A continuación, les mostramos unos ejemplos de las pruebas realizadas:

```
test 'test search ajax without params' do
  get :search
  params = {"utf8"=>"✓", "date"=>{"init(3i)"=>"8", "init(2i)"=>"2",
    "init(1i)"=>"2000", "end(3i)"=>"8", "end(2i)"=>"2", "end(1i)"=>"2015"}, "town"=>
  "tion"=>"" , "intensity"=>"" , "commit"=>"Buscar"}
  xhr :get, :index, params, :format => 'js'
  assert_response :success
  assert_nil assigns(:results), assigns(:results)
end
```

Figura 9.1 Test del controlador

```
test "Station attributes must not be empty" do
  @empty_station = Station.new
  assert @empty_station.invalid?
  assert @empty_station.errors[:nombre].any?
  assert @empty_station.errors[:med24h].any?
  assert @empty_station.errors[:med7d].any?
  assert @empty_station.errors[:med30d].any?
end
```

Figura 9.2 Test del modelo

10. Resultados y conclusiones

Los objetivos que se habían planteado inicialmente para el proyecto se han cumplido completamente, desarrollando la aplicación web que se adapta a los diferentes dispositivos de los que habíamos hablado y desplegándola en producción mediante el servicio de PAAS Heroku.

Se han creado tres vistas para cada tamaño de pantalla de dispositivo, móvil, tableta y escritorio, teniendo prácticamente las mismas funcionalidades. Esto nos permite ver la información de las lluvias mediante diferentes elementos, desde un mapa que ofrece una visión general de la isla, pasando por tablas y gráficas que te dan una visión específica de las situaciones de los municipios y estaciones.

El desarrollo de la misma ha llevado más tiempo de lo que se había planteado ya que la adaptación de las vistas a los diferentes tamaños es un tanto tediosa a pesar de usar un framework que te facilite la tarea.

Llevar a cabo el proyecto final de carrera ha sido una de las cosas más difíciles que he realizado ya que parece que te enfrentas a un reto interminable con multitud de dificultades en cada fase del mismo, pero a medida que vas superando cada una de ellas, vas viendo que para nada es algo imposible.

He afianzado mis conocimientos poniéndolos en práctica y aclarándolos, gracias a la ayuda de los tutores, que me aportaban sus conocimientos en cada paso del proyecto, marcando el camino adecuado a seguir.

Gracias a este proyecto, he podido aprender multitud de lenguajes nuevos y mejorar las cualidades personales de constancia y esfuerzo, que me ayudarán de cara al mundo laboral a integrarme más rápido y a crecer como profesional de la informática.

11. Trabajo Futuro

Se propone a continuación una serie de aspectos que podrían ser desarrollados de cara a mejorar y completar la aplicación web que ha sido implementada:

- **Mejorar el mapa:** Añadir las estaciones al mapa, de forma que se pueda ver la información de las estaciones desde el propio mapa.
- **Mejorar la función de compartir:** Poder compartir una imagen de la situación actual de la aplicación en redes sociales y por email.
- **Buscador en el móvil y tableta:** Añadir a las vistas de móvil y tableta un buscador adaptado.
- **Panel de administración:** Añadir una panel de administración para poder manipular los elementos de la aplicación.
- **Permitir suscripciones a los datos:** Añadir la funcionalidad de suscripción de los usuarios para que reciban un boletín diario de los datos.

12. Bibliografía

1. Consejo Insular de Aguas. <http://www.aguasgrancanaria.com/>
2. SAIH. <http://www.magrama.gob.es>
3. SAIH Ebro. <http://www.saihebro.com/>
4. AEMET. <http://www.aemet.es>
5. Met.ie. <http://www.met.ie/>
6. Rain Rauge. <http://www.edgewaterconcepts.com/>
7. Pressman, Roger S. “Ingeniería del Software. Un enfoque práctico.”
8. Mobile First. https://en.wikipedia.org/wiki/Responsive_web_design
9. MVC. <https://es.wikipedia.org/wiki/Modelo-vista-controlador>
10. Agile Web Development with Rails. 4ª Edition.
11. API de Ruby on Rails. <http://api.rubyonrails.org/>
12. Rails Guides. <http://guides.rubyonrails.org/>
13. RailsCasts: Ruby on Rails screencasts. <http://railscasts.com/>
14. Foundation. <http://foundation.zurb.com/>
15. LEAFLET. <http://leafletjs.com>
16. Omnivore.js. <https://github.com/mapbox/leaflet-omnivore>
17. TopoJSON. <https://github.com/mbostock/topojson>
18. TableSorter. <http://tablesorter.com/docs/>
19. Highcharts. <http://api.highcharts.com/highcharts>
20. Heroku. <http://www.heroku.com/>
21. W3schools homepage. <http://www.w3schools.com>
22. jQuery API Documentation. <http://api.jquery.com/>
23. BitBucket homepage. <https://bitbucket.org/>
24. Git Documentation. <http://git-scm.com/documentation>
25. Stack Overflow. <http://stackoverflow.com/>

26. Wikipedia. <http://www.wikipedia.org/>

27. Interactive Map. <http://www.tnoda.com/blog/2013-12-07>

Anexo I: Manual de uso de la aplicación

En este apartado, se describirá el uso general de la aplicación web.

Vista principal

Como se puede ver en los tres paneles de navegación de las diferentes vistas ordenados por móvil, tableta y escritorio (Ver Figura 0.1(a, b y c)). El funcionamiento es similar, puedes elegir si ver los datos mediante un mapa o una tabla, excepto en el escritorio, que ya por defecto se muestran los dos juntos, además de, elegir el intervalo de tiempo de tiempo de los datos.



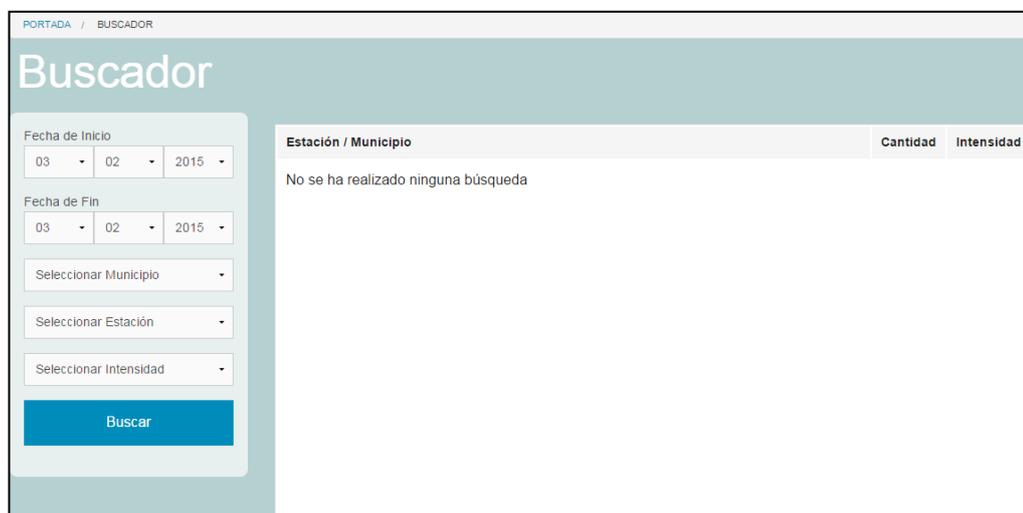
Figura 0.1 Paneles de las vistas

En el lado derecho se puede apreciar un icono en los tres paneles, si lo pulsamos, se nos abrirá una ventana para compartir lo que estamos viendo en diferentes redes sociales (Facebook o Twitter) o por email (Ver Figura 0.2).



Figura 0.2 Panel de compartir

En el panel del escritorio, podemos ver en el lado izquierdo un icono en forma de lupa, si lo pulsamos iremos a la Figura 0.3 que es un buscador de datos, donde poder seleccionar datos específicos de las precipitaciones.

A screenshot of a web application's search interface. The page title is 'Buscador'. On the left side, there are several input fields: 'Fecha de Inicio' with a date picker set to 03/02/2015, 'Fecha de Fin' with a date picker set to 03/02/2015, and three dropdown menus labeled 'Seleccionar Municipio', 'Seleccionar Estación', and 'Seleccionar Intensidad'. Below these is a blue 'Buscar' button. On the right side, there is a table with the following structure:

Estación / Municipio	Cantidad	Intensidad
No se ha realizado ninguna búsqueda		

Figura 0.3 Vista del buscador – Escritorio

Mapa

El uso del mapa en las tres vistas es el mismo, se mostrará el mapa coloreado según las precipitaciones y se podrá hacer clic en cada uno de los municipios del mapa, mostrándose un panel con el nombre y la cantidad de precipitación media caída, en diferentes intervalos de tiempo. Además de poder ir directamente a ver los datos de ese municipio.(Ver Figura 0.4)

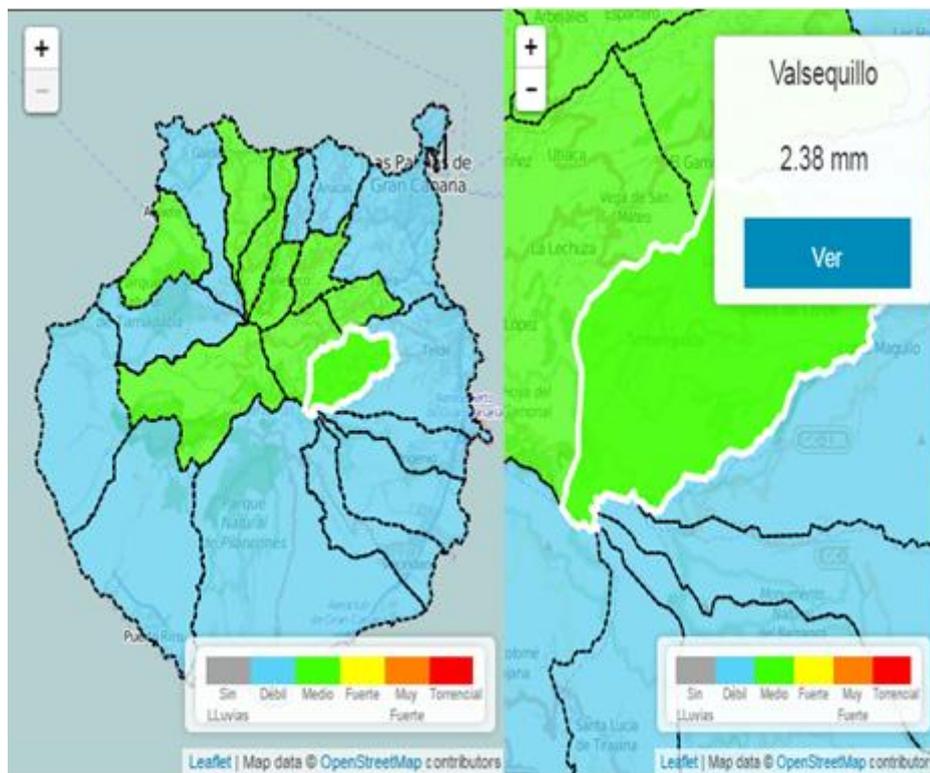


Figura 0.4 Funcionamiento del mapa

Tabla

Como en el caso del mapa, el uso de la tabla es el mismo en todas las versiones, los datos vendrán ordenados en orden decreciente, filtrados por municipios o estaciones (Ver Figura 0.5).

24H			7D			30D					
Por municipios			Por estaciones			Por municipios			Por estaciones		
Municipios	Precipitación	Intensidad	Estación / Municipio	Precipitación	Intensidad						
Agaete	2.96	Medio	Mogán Pueblo / Mogán	19.50	Débil						
San Mateo	1.57	Débil	Cuevas del Pinar / San Bartolomé	18.17	Débil						
Valsequillo	1.54	Débil	Pajonales Pinar / Tejeda	14.75	Medio						
Tejeda	1.42	Débil	Hoya del Gamonal / San Mateo	11.20	Medio						
Valleseco	1.18	Débil									

Figura 0.5 Funcionamiento de la tabla

Vistas Intermedias

En este apartado, se muestran el funcionamiento de las vistas intermedias de la aplicación web. En todas ellas, se podrá compartir los datos mediante el botón compartir, que es el q está más a la izquierda (Ver Figura 0.6 (a)) y ver los datos mediante una tabla o una gráfica (Ver Figura 0.6 (b y c)).



Figura 0.6 Panel de opciones de las vistas intermedias

Mientras estemos en estas vistas podremos navegar mediante el elemento llamado 'Breadcrumbs', que nos mostrará dónde estamos y a donde podemos ir.



Figura 0.7 Breadcrumbs

Municipio

Cuando se hace clic en un municipio se nos muestra las precipitaciones medias en orden decreciente de sus estaciones. Además de poder ver una gráfica con las lluvias medias de los últimos 30 días (Ver Figura 0.8).

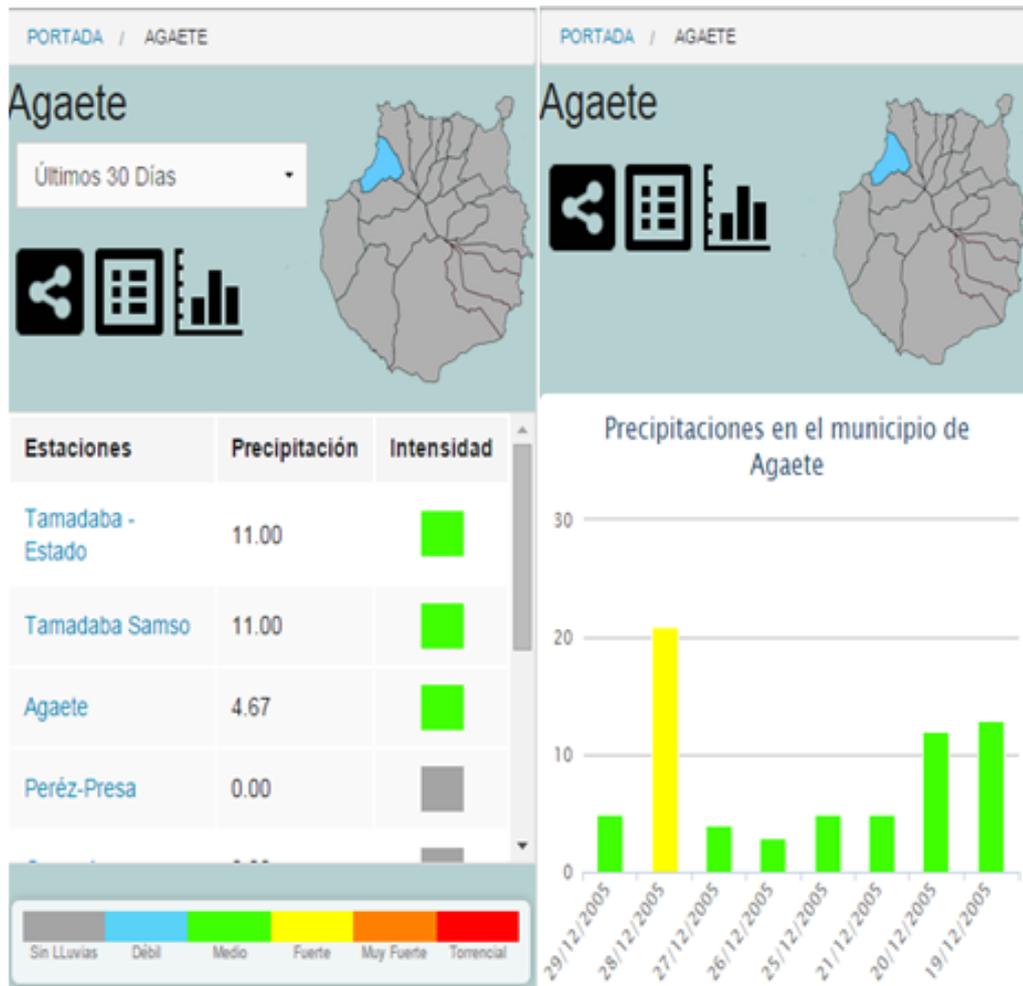


Figura 0.8 Funcionamiento de las vistas de municipios

Estación

Cuando se hace clic en una estación se muestra una vista parecida a la de los municipios, donde se mostrarán las precipitaciones de los últimos 30 días, mediante una tabla o una gráfica (Ver Figura 0.9).

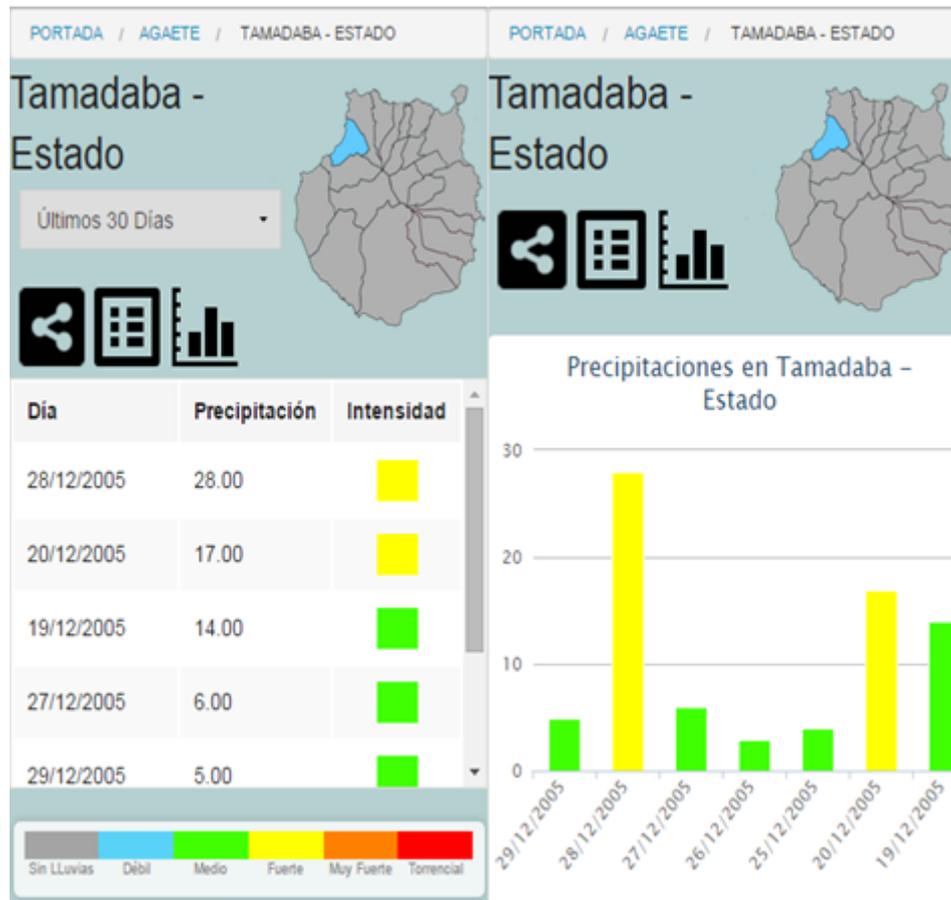


Figura 0.9 Funcionamiento de las vistas de las estaciones

Anexo II: Detalles sobre la creación del mapa del proyecto

Creación del mapa coroplético

El mapa es un elemento fundamental en la aplicación ya que te da una visión general de la situación de las precipitaciones de la isla, además de aportar mayor interactividad a la misma.

Comparativa de librerías

Lo primero de todo fue hacer un estudio de diferentes librerías en JavaScript que poseían una serie de requisitos como pueden ser el dibujo de mapas, eventos de interacción, compatibilidad para móviles, etc.

D3.JS

Es una biblioteca de JavaScript para la manipulación de documentos basados en datos, que ayuda a llevar los datos a la vida usando HTML, SVG y CSS. El uso de D3 en estándares web te da todas las capacidades de los navegadores modernos sin engancharte a un marco de propiedad, que combina componentes de visualización de gran alcance y un enfoque impulsado por los datos a la manipulación DOM.

En la Tabla 0-1 se presentan las ventajas y desventajas de la librería D3.js, después de estudiarlas, concluimos que es muy potente, ya que te permite hacer desde una simple gráfica hasta complejas figuras con multitud de eventos y opciones para afinar hasta el último detalle, obteniendo un resultado excelente, por el contrario, esto conlleva que tienes que dedicarle bastante tiempo porque la curva de aprendizaje es elevada, además de, que no es una librería precisamente ligera y el rendimiento en dispositivos móviles, no es del todo óptimo.

Ventajas	Desventajas
Tiene una gran cantidad de opciones donde puedes manipular hasta el último detalle la visualización de los datos	148 Kb comprimido
Multitud de ejemplos	Curva de aprendizaje elevada
Gran comunidad de usuarios	
Open source	

Tabla 0-1 Ventajas y desventajas - D3.js

JQMAPS

Es un plugin de jQuery que está basado en la librería JVectorMaps, realiza mapas vectoriales utilizando SVG para navegadores modernos como Firefox, Safari, Chrome, Opera e Internet Explorer 9.

En la Tabla 0-2 Tabla 0-1 se presentan las ventajas y desventajas del plugin JQMaps y la librería en la que está sobre la que está construido, una vez estudiados, concluimos que puede ser una opción interesante si tuviera una adecuada documentación o al menos una mayor cantidad de ejemplos, para poder hacer un uso óptimo y saber hasta qué punto nos es adecuada para nuestros propósitos. Además no cuenta con una gran comunidad detrás que la mantenga o la use.

Ventajas	Desventajas
17 Kb comprimido	Poca documentación
Mapas personalizados	Pocos ejemplos
	Falta de comunidad

Tabla 0-2 Ventajas y desventajas - JQMaps

GOOGLE CHART

Es una API de visualización de la empresa Google que te permite crear diferentes tipos de gráficas pasándole datos mediante JSON, devolviéndote la gráfica resultante, es compatible con amplia variedad de navegadores y dispositivos móviles.

En la Tabla 0-3Tabla 0-1 se presentan las ventajas y desventajas de la API de visualización de Google, concretamente el GeoChart, que es tipo de gráfica, se puede concluir que a pesar de su simplicidad y de una buena documentación, vemos que no soporta eventos como hacer zoom, arrastrar o desplazar y tampoco se puede realizar mapas personalizados.

Ventajas	Desventajas
25Kb comprimido	No puedes realizar mapas personalizados
Simplicidad	Es una API
Gran cantidad de ejemplos y documentación	No soporta eventos de interactividad

Tabla 0-3 Ventajas y desventajas - Google Charts

LEAFLET.JS

Es una librería JavaScript de open-source ampliamente utilizada para construir mapas interactivos en aplicaciones web, está diseñada para que su uso sea simple, y de alto rendimiento. Funciona de manera eficiente a través de todas las principales plataformas de escritorio y móviles, aprovechando HTML5 y CSS3 en los navegadores modernos sin dejar de ser accesibles en otros más antiguos. Es utilizada por FourSquare, Pinterest y Flickr.

En la Tabla 0-4Tabla 0-1 se presentan las ventajas y desventajas de la librería Leaflet.js, y una vez analizada, podemos concluir que es bastante completa para el tamaño que posee, dispone de una buena documentación, además de una gran cantidad de ejemplos donde se puede ver las diferentes utilidades que posee. También debido a su popularidad posee una gran comunidad detrás a la cuál recurrir.

Ventajas	Desventajas
33Kb	
Mapas personalizados	
Buena documentación	
Gran Comunidad	

Tabla 0-4 Ventajas y desventajas - Leaflet.js

Resumen

Una vez estudiadas todas las librerías anteriores podemos concluir que la que mejor se adapta a nuestras necesidades es Leaflet.js, ya que es la mejor que cumple los requisitos mostrados en la Tabla 0-5

Las librerías JQMaps y Google Chart no disponen de eventos de interactividad que es una característica imprescindible para nuestra aplicación. Y por otra parte, D3.js, a pesar de cumplir también todos los requisitos como Leaflet.js, es demasiado pesado y con más opciones de las que necesitamos para realizar nuestro trabajo.

Por estos motivos nos decantamos por la librería Leaflet.js ya que es la más adecuada para nuestra aplicación web.

	D3.js	JQMaps	GCharts	Leaflet.js
Aprendizaje	Medio	Fácil	Fácil	Fácil
Peso	148Kb	17Kb	25Kb	33Kb
Compatibilidad	Sí	Sí	Sí	Sí
Interactividad	Sí	No	No	Sí
Comunidad	Sí	No	Sí	Sí
Documentación	Sí	No	Sí	Sí
OpenSource	Sí	Sí	Sí	Sí

Tabla 0-5 Resumen requisitos - Comparativa de librerías JS

Capa interactiva del mapa

Leaflet.js nos permite superponer capas sobre el mapa con las que puedes añadirle colores e interactividad, por tanto, vamos a proceder a realizar una capa con estas características para la isla de Gran Canaria.

Para obtener la capa de la isla de Gran Canaria vamos a seguir un procedimiento donde iremos transformándola en diferentes formatos la para obtener la máxima calidad en el mínimo tamaño compatible con la web.

Lo primero que vamos a realizar es descargarnos el archivo shapefile de la web del Gobierno de Canarias, con las opciones que aparecen en la Figura 0.1. Este archivo posee la geometría de la isla que la exportaremos en otros formatos para que sean compatibles y usables con Leaflet.js.

Figura 0.1 Web del Gobierno de Canarias – Shapefile

Shapefile GeoJSON Topojson

Para poder usar la información del shapefile en una aplicación web, hay que hacer una conversión a JSON. Para ello, pasaremos de shapefile a GeoJSON y de GeoJSON a Topojson. Con estas conversiones estamos reduciendo el tamaño de los datos manteniendo una calidad aceptable de los mismos.

Aquí podemos ver un ejemplo de la diferencia de tamaño que habrá al finalizar la conversión de la información.

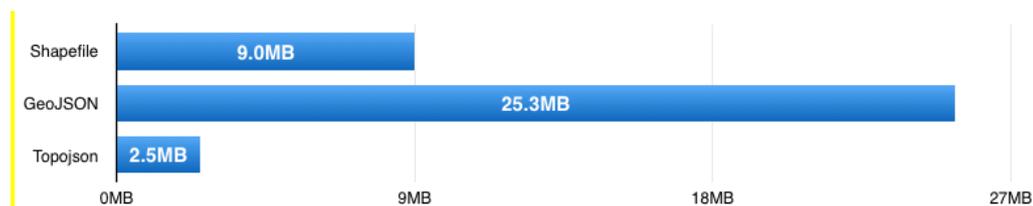


Figura 0.2 Comparativa del tamaño resultando de la conversión

Para realizar la conversión vamos a usar la librería open source **ogr2ogr** que sirve para traducir esta información entre distintos formatos. Un detalle a tener en cuenta, es el tipo de proyección en la que está el archivo, ya que a la hora de hacer la conversión no saldrá la forma que correcta.

En nuestro caso, nuestro archivo estaba en la proyección EPSG: 32628, que es la establecida para el Huso UTM de Gran Canaria (Ver Figura 0.3(a)) y lo pasamos al EPSG: 4326, que es el general para todo el mapa del mundo (Ver Figura 0.3(b)).

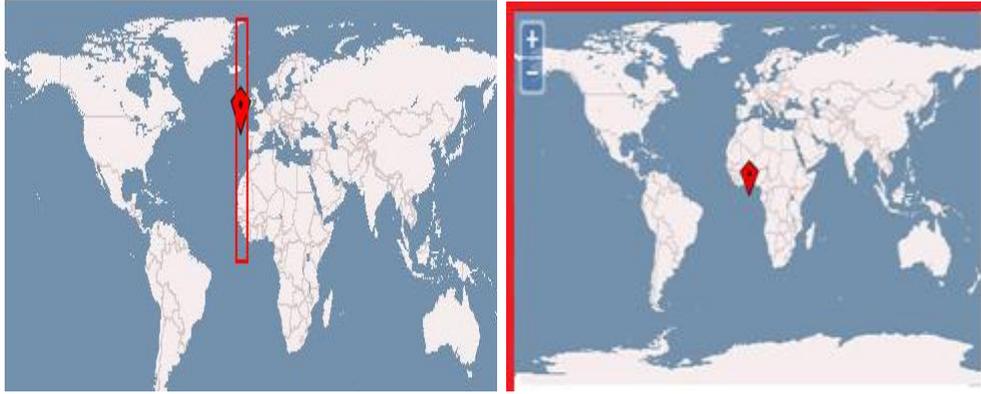


Figura 0.3 Mapas de las proyecciones EPSG 32628 y 4326

```
> ogr2ogr -f GeoJSON MUNICIPIOS.json municipios.shp -s_srs EPSG:32628 -t_srs EPSG:4326
```

Como se ve en el ejemplo anterior con la primera conversión obtenemos un archivo GeoJSON muy grande, esto es debido, a causa de que tiene muchos vértices y atributos en sus datos, de los cuales no necesitamos todos, por tanto, vamos a realizar una serie de procedimientos para reducirlos.

Primero nos centraremos en reducir los puntos geométricos que tiene el mapa, para ello, vamos a usar una herramienta online llamada MapShaper, que es que nos ayudará a reducirlos manteniendo la calidad de la topografía mediante una serie de algoritmos específicos.

Lo único que tenemos que hacer es subir el archivo GeoJSON y bajar la calidad de la misma mediante una barra superior (Ver Figura 0.4). En las siguientes imágenes se ha reducido la calidad de un 100% a un 0.9% si apenas apreciar cambios a primera vista, hemos pasado de 3.8 Mb a 38Kb, el tamaño ahora del archivo es el 1% del tamaño que tenía anteriormente.

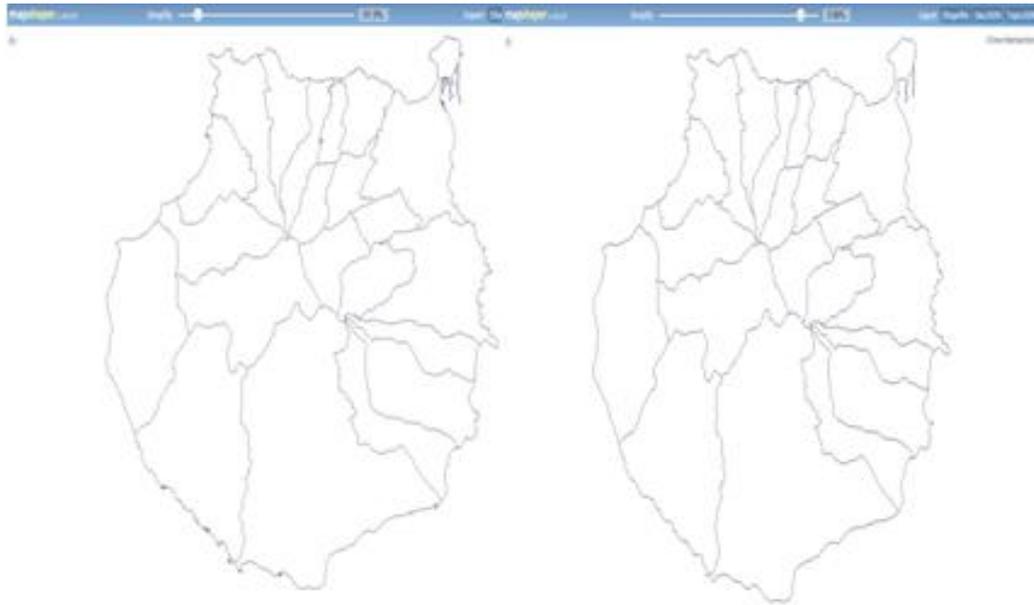


Figura 0.4 Comparativa entre los mapas de Gran Canaria - MapShaper

Ahora por último, nos falta hacer la última transformación, pasar de GeoJSON a TopoJSON, para ello, utilizaremos un paquete CLI llamado Topojson que instalaremos mediante Node.js.

```
npm install -g topojson
```

Una vez instalado aplicamos el siguiente comando, diciéndole mediante `-p` que propiedades del archivo deseamos conservar, en nuestro caso, solo nos quedamos con la propiedad llamada `MUNICIPIOS`, que contiene el nombre propio de cada término municipal.

```
topojson --id-property MUNICIPIOS -o gc.topojson MUNICIPIOS.json
```

Como se puede ver en la Figura 0.5 después de todas las transformaciones hemos pasado de 3Mb a 28Kb, es un gran resultado y tamaño perfecto para poder ser tratado en una aplicación web.

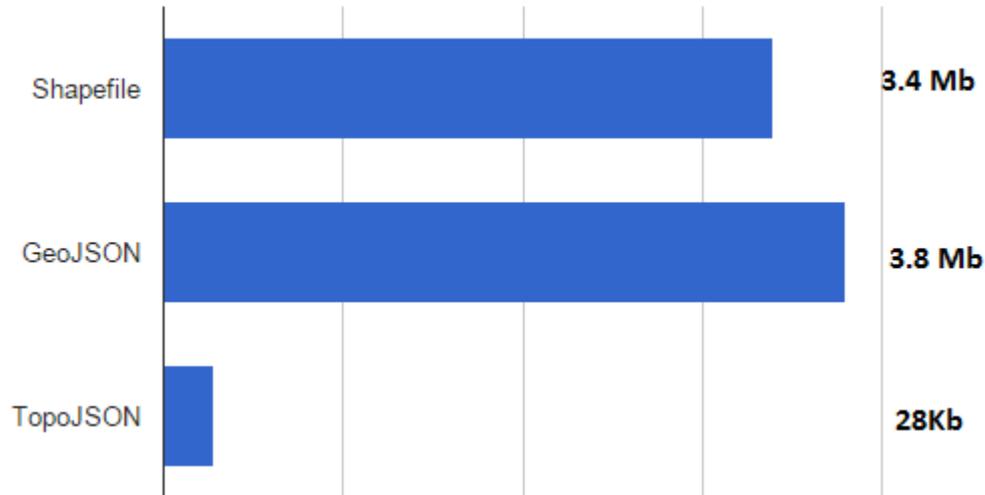


Figura 0.5 Comparativa de tamaños después de la conversión

En las siguientes imágenes vemos el resultado de haber exportado los datos a leaflet.js y haber superpuesto la capa encima del mapa original.

Leaflet.js no soporta nativamente el formato TopoJSON, por tanto, tuvimos que usar un plugin llamado Omnivore, que se encarga de pasar el código a GeoJSON y lo añadimos al mapa (Ver Figura 0.6).

```
geojson = L.geoJson(null, {  
  style: style,  
  onEachFeature: onEachFeature  
});  
var m1 = omnivore.topojson('gc.topojson', null, geojson)
```

Figura 0.6 Conversión de datos a GeoJSON

El resultado se ve en la Figura 0.7.



Figura 0.7 Comparativa del mapa con la capa de interacción

Mediante las siguientes funciones de leaflet.js pudimos añadirle color y e interactividad mediante gestos de zoom en las zonas del mapa donde se hiciera clic.

En la Figura 0.8 se centra la interactividad, llamando a diversas funciones, para hacer transiciones en el mapa.

```
function onEachFeature(feature, layer) {  
  layer.on({  
    mouseover: highlightFeature,  
    mouseout: resetHighlight,  
    click: zoomToFeature  
  });  
}
```

Figura 0.8 Función encargada de la interacción del mapa

En estas dos funciones (Ver Figura 0.9), controlamos los colores que ponemos en el mapa, una lee las cantidades medias de las lluvias en los municipios y la otra establece el color en la capa concreta.

```
function getColor(p) {
    return p > 60 ? '#FF0000' : //rojo
           p > 30 ? '#FF8000' : //naranja
           p > 15 ? '#FFFF00' : //amarillo
           p > 2  ? '#40FF00' : //verde
           p > 0  ? '#58D3F7' : //aguamarina
           '#A4A4A4'; //gris
}

function style(feature) {
    return {
        fillColor: getColor(feature.properties.m1),
        weight: 2,
        opacity: 1,
        color: 'black',
        dashArray: '3',
        fillOpacity: 0.7
    };
};
```

Figura 0.9 Funciones encargadas de la coloración del mapa

Y como resultado vemos en la Figura 0.10 la ampliación de la zona de San Bartolomé de Tirajana y en la otra imagen vemos la coloración de diversas zonas según la cantidad de lluvia.

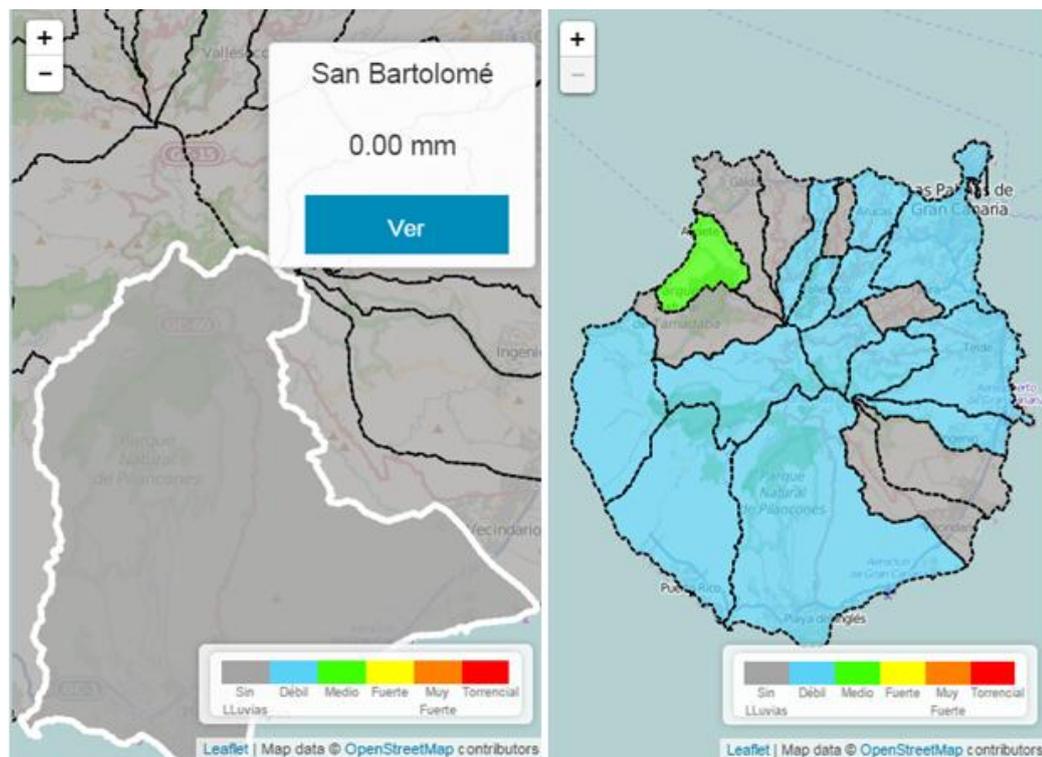


Figura 0.10 Ejemplos de la interacción y coloración del mapa