



**UNIVERSIDAD DE EXTREMADURA**

**CENTRO UNIVERSITARIO DE MÉRIDA**

**Grado en Ingeniería Informática en Tecnologías de la  
Información**

**TRABAJO FIN DE GRADO**

**Aplicación web para el despliegue  
robusto de sensores para agricultura  
intensiva**

Autor: Rafael Montaña Boraita

Fdo.: 

Director: Francisco Luna Valero

Fdo.:

Director: Rafael Marcos Luque Baena

Fdo.:



# Índice general

<b>INTRODUCCIÓN .....</b>	<b>- 10 -</b>
1.1 OBJETIVOS.....	- 11 -
1.2 METODOLOGÍA.....	- 12 -
<b>CONCEPTOS BÁSICOS.....</b>	<b>- 15 -</b>
2.1 HTML 5 .....	- 15 -
2.2 CSS 3.....	- 16 -
2.3 BOOTSTRAP 3 .....	- 18 -
2.4 ANGULARJS .....	- 22 -
2.5 JAVASCRIPT.....	- 22 -
2.6 JQUERY .....	- 23 -
2.7 AJAX .....	- 25 -
2.8 PHP.....	- 26 -
2.9 JAVA.....	- 27 -
2.10 JAX-RS .....	- 28 -
2.11 JMETAL .....	- 28 -
2.12 ENTORNOS DE TRABAJO UTILIZADO .....	- 30 -
2.12.1 Dreamweaver.....	- 30 -
2.12.2 Photoshop .....	- 31 -
2.12.3 Filezilla .....	- 31 -
2.12.4 Xampp.....	- 31 -
2.12.5 Eclipse .....	- 32 -
<b>ANÁLISIS .....</b>	<b>- 34 -</b>
3.1 PRIMERA FASE .....	- 35 -
3.1.1.1 Primer apartado .....	- 35 -
3.1.1.2 Segundo apartado:.....	- 35 -
3.1.1.3 Tercer apartado:.....	- 36 -
3.1.1.4 Cuarto apartado.....	- 36 -
3.1.1.5 Quinto apartado .....	- 37 -
3.1.2 <i>Diseño de web</i> .....	- 37 -
3.1.3 <i>Diseño casos de uso</i> .....	- 38 -
3.1.3.1 Home:.....	- 38 -
3.1.3.2 Despliegue de Mapas:.....	- 38 -
3.1.3.3 Información:.....	- 39 -
3.1.3.4 Contacto: .....	- 39 -

## Índice General

3.1.3.5 Registro.....	- 40 -
3.1.4 Imagen de web.....	- 40 -
3.2 SEGUNDA FASE.....	- 42 -
3.2.1 Organización de apartados.....	- 42 -
3.2.2 Diseño de web.....	- 44 -
3.2.3 Diseño casos de uso .....	- 46 -
3.2.4 Servicio Web.....	- 47 -
3.2.4.1 Conexiones usadas entre servidores .....	- 48 -
3.2.5 Imágenes de del servicio web .....	- 51 -
3.2.5.1 Ordenador: .....	- 51 -
3.2.5.2 Tableta: .....	- 52 -
3.2.5.3 Smartphone:.....	- 53 -
<b>DISEÑO .....</b>	<b>- 57 -</b>
4.1 ESQUELETO.....	- 57 -
4.2 HOME .....	- 59 -
4.3 INFORMACIÓN.....	- 62 -
4.4 CONTACTO .....	- 64 -
4.5 DESPLEGAR SENSORES.....	- 66 -
4.5.1 Mapa.....	- 67 -
4.5.1.1 Búsqueda de situaciones .....	- 68 -
4.5.1.2 Herramienta de polígonos.....	- 68 -
4.5.1.3 Cálculo de área .....	- 69 -
4.5.1.4 Borrar Shape del mapa .....	- 69 -
4.5.1.5 Paleta de colores .....	- 70 -
4.5.2 Herramienta guardar mapa.....	- 70 -
4.5.3 Cargar Mapa .....	- 71 -
4.5.3.1 Popup .....	- 72 -
4.5.3.2 Upload .....	- 72 -
4.5.4 Despliegue de sensores .....	- 73 -
4.6 SERVIDOR DE OPTIMIZACIÓN .....	- 77 -
4.6.1 Comunicación entre servidores. ....	- 77 -
4.6.2 Cálculo de despliegue de sensores .....	- 79 -
4.6.3 Esquemas comunicación entre servidores .....	- 81 -
4.6.4 Esquema de lenguajes utilizados .....	- 82 -
4.7 ESQUEMA DE ARCHIVOS .....	- 83 -
4.8 OPTIMIZACIÓN DE PÁGINA WEB.....	- 84 -
4.8.1 Corrección de fallos HTML.....	- 84 -
4.8.2 Carga en dispositivos .....	- 85 -
4.8.2.1 Optimizaciones de imágenes.....	- 85 -

# Índice General

4.8.2.2 Optimización archivos .....	- 85 -
4.8.2.3 Conclusión .....	- 86 -
<b>RESULTADOS.....</b>	<b>- 88 -</b>
5.1 DISPOSITIVO ORDENADOR SOBREMESA.....	- 88 -
5.1.1 Home.....	- 88 -
5.1.2 Información.....	- 89 -
5.1.3 Contacto.....	- 89 -
5.1.3.1 Prueba de correo.....	- 90 -
5.1.4 Desplegar sensores .....	- 91 -
5.4.1.1 Crear polígonos .....	- 92 -
5.4.1.2 Descarga archivo mapa .....	- 93 -
5.4.1.3 Borrar polígonos.....	- 94 -
5.4.1.4 Carga de áreas.....	- 94 -
5.4.1.4 Sensores desplegados .....	- 97 -
5.2 DISPOSITIVO NEXUS 5.....	- 102 -
5.2.1 HOME .....	- 102 -
5.2.2 INFORMACIÓN.....	- 103 -
5.2.3 CONTACTO .....	- 104 -
5.2.3.1 Prueba contacto en móvil .....	- 105 -
5.2.4 DESPLEGAR SENSORES.....	- 106 -
5.2.4.1 Herramientas de áreas.....	- 106 -
5.2.4.2 Herramienta búsqueda situación.....	- 107 -
5.2.4.3 Guardar y cargar archivo .....	- 108 -
5.3 DISPOSITIVO IPAD MINI .....	- 109 -
5.3.1 Home.....	- 109 -
5.3.2 Información.....	- 110 -
5.3.3 Contacto.....	- 110 -
5.3.3.1 Prueba contacto Ipad .....	- 111 -
5.3.4 Desplegar sensores .....	- 112 -
5.3.4.1 Herramienta buscar situación .....	- 112 -
5.3.4.2 Herramienta crear áreas .....	- 113 -
5.3.4.3 Cargar y guardar mapa de áreas.....	- 114 -
<b>CONCLUSIONES Y TRABAJO FUTURO .....</b>	<b>- 116 -</b>
6.1 CONCLUSIONES.....	- 116 -
6.2 AMPLIACIONES Y TRABAJO FUTURO.....	- 118 -
<b>MANUAL DEL USUARIO.....</b>	<b>- 121 -</b>
7.1 ACCEDER A SMART CULTIVATION .....	- 121 -

## Índice General

7.2 MENÚ Y FOOTER .....	- 121 -
7.2 HOME .....	- 122 -
7.3 INFORMACIÓN.....	- 123 -
7.4 CONTACTO.....	- 123 -
7.5 DESPLEGAR SENSORES.....	- 124 -
7.5.1 Manipulación de mapa .....	- 124 -
7.5.2 Herramientas de áreas.....	- 126 -
7.5.3 Herramienta Línea .....	- 128 -
7.5.4 Herramienta Borrar elemento.....	- 129 -
7.5.5 Herramienta paleta de colores.....	- 129 -
7.5.6 Herramienta búsqueda de lugar .....	- 131 -
<b>AGRADECIMIENTOS .....</b>	<b>- 133 -</b>
<b>BIBLIOGRAFÍA .....</b>	<b>- 134 -</b>

# Índice de figuras

---

1 Diseño de web, primera fase	- 38 -
2 Diseño de caso de uso, primera fase	- 40 -
3 Web home, primera fase	- 41 -
4 Web Información, primera fase	- 41 -
5 Diseño web, segunda fase	- 46 -
6 Casos de uso, segunda fase	- 47 -
7 Web Home, ordenador	- 51 -
8 web información, tableta	- 52 -
9 web mapa, móvil	- 53 -
10 Presentación home	- 62 -
11 Presentación información	- 64 -
12 Presentación contacto	- 66 -
13 Presentación despliegue de sensores	- 76 -
14 Despliegue de sensores en área real	- 76 -
15 Formulario de despliegue, Normal y Avanzado	- 78 -
16 Comunicación servicio Web	- 82 -
17 Lenguajes utilizados	- 82 -
18 Esquema archivos	- 84 -
19 Home Ordenador	- 88 -
20 Información Ordenador	- 89 -
21 Contacto Ordenador	- 89 -
22 Prueba correo campo incorrecto	- 90 -
23 Prueba correo campo incorrecto	- 90 -
24 Correo enviado correctamente	- 91 -
25 Correo recibido correctamente	- 91 -
26 Despliegue sensores ordenador	- 91 -
27 Polígonos creados ordenador	- 92 -
28 Polígonos creados ordenador	- 92 -
29 Descarga mapa ordenador	- 93 -
30 Mapa descargado ordenador	- 93 -
31 Polígonos borrados ordenador	- 94 -
32 Carga archivo áreas ordenador	- 94 -
33 Arrastrando archivo ordenador	- 95 -
34 Archivo cargado ordenador	- 95 -
35 Subiendo archivo ordenador	- 96 -

## Índice de figuras

36 Áreas cargadas ordenador	- 96 -
37 Selección de área sencilla	- 97 -
38 Parámetros de área sencilla	- 97 -
39 Sensores desplegados Sencillo	- 98 -
40 Área con distintas peligrosidades	- 98 -
41 Parámetros de baja robustez	- 99 -
42 Despliegue peligrosidad y baja robustez	- 99 -
43 área sin peligrosidad y robusta	- 100 -
44 Parámetros despliegue robusto	- 100 -
45 Sensores desplegados sin peligrosidad	- 101 -
46 Sensores desplegados con robustez y peligrosidades	- 101 -
47 Menú en home      48 Home móvil 49 Home móvil	- 102 -
50 Información móvil      51 Información móvil	- 103 -
52 Reproduciendo video completo	- 103 -
53 Contacto móvil      54 Contacto móvil	- 104 -
55 Error enviar mensaje móvil	- 105 -
56 Envió correcto móvil	- 105 -
57 Creando polígono móvil      58 Ampliando área móvil	- 106 -
59 Leyenda colores móvil	- 107 -
60 Búsqueda móvil      61 Sitio encontrado móvil	- 107 -
62 Despliegue menú carga mapa móvil	- 108 -
63 Home Ipad      64 Home Ipad	- 109 -
65 Información Ipad      66 Información Ipad	- 110 -
67 Contacto Ipad	- 110 -
68 Fallo contacto Ipad	- 111 -
69 Relleno de contacto Ipad      70 Confirmación contacto Ipad	- 111 -
71 Desplegar sensores Ipad	- 112 -
72 Herramienta situación Ipad      73 Situación encontrada Ipad	- 112 -
74 Creación de área Ipad      75 Modificación de área	- 113 -
76 Muestra de áreas creadas Ipad	- 113 -
77 Carga de mapas Ipad	- 114 -
78 Barra dirección	- 121 -
79 Menú de navegación	- 121 -
80 Footer	- 122 -
81 Botones información Home	- 122 -
82 Controles de vídeo	- 123 -
83 Formulario de contacto	- 124 -
84 herramientas Google maps	- 125 -



## Índice de figuras

85 Botón satélite con etiquetas	- 126 -
86 Botón Mapa con relieve	- 126 -
87 Herramientas áreas	- 126 -
88 Área creada	- 127 -
89 Dimensión de área	- 128 -
90 Línea dibujada	- 128 -
91 Área eliminada	- 129 -
92 Leyenda paleta de colores	- 130 -
93 Línea cambiada de color	- 130 -
94 Herramienta buscar	- 131 -



# Capítulo 1

---

## Introducción

Este proyecto fin de grado se enmarca dentro del uso de las Tecnologías de la Información y Comunicación (TIC) en el mundo agrario, un sector de gran impacto en España en general y más en concreto en la Comunidad de Extremadura. La modernización de las explotaciones agrarias es un factor clave para mejorar la competitividad del sector, peculiarmente las TIC juegan un papel clave. En concreto, el trabajo desarrollado en este TFG ha ido encaminado en proporcionar un servicio web que permite desplegar, de forma robusta, un conjunto de sensores en una finca que se desea monitorizar. La monitorización adecuada de una finca permite, por ejemplo, ahorrar el consumo de agua, realizar fertilizaciones a medida o detectar posibles incidencias en zonas concretas del terreno (contaminación, desajustes del pH, etc.). Se hablará de despliegue de sensores robusto ya que el usuario podrá determinar qué zonas de su finca son más propensas a que uno los sensores dejen de funcionar (condiciones climatológicas extremas, proximidad de un camino muy transitado, etc.). La aplicación únicamente necesita una conexión a Internet para su correcto funcionamiento.

Con este servicio se podrá realizar un despliegue de sensores en un área determinada, la cual los usuarios previamente hayan elegido. Con las herramientas que proporciona el servicio, el usuario podrá seleccionar el área arbitraria del terreno, así poder asignar diferentes grados de peligrosidad a las diferentes zonas. Este grado de peligrosidad modela la probabilidad de caída de un sensor que se posiciona en esa zona. Todo esto es posible realizarlo gracias a la integración de los mapas de Google Maps que estará en un apartado del servicio web.

Esto se aplicará a un problema real, eligiendo un área con la herramienta que se proporcione. Con el área que se genere se podrá hacer un despliegue de sensores. Es necesario dar cobertura a los sensores para el terreno, para ello se calcula el tamaño, se determina cuántos sensores se van a necesitar y cuáles serán las posiciones exactas donde colocaría cada sensor. La posición

de estos sensores será calculada por un algoritmo con un cálculo mateheurístico multiobjetivo, por esto el despliegue tendrá una robustez muy alta.

Este algoritmo realiza el despliegue de forma automática, monitorizando todo el terreno. Para ello se calcula el área total del área asignada, se hace una previsión optimista del número de sensores máximo que será necesario. El algoritmo entonces posiciona los sensores de forma que se minimice su número (ahorro de costes del despliegue), la energía consumida por la red (para que su tiempo de vida sea máximo) y una medida de la robustez del despliegue (medida en base a caídas aleatorias que serán más frecuentes en las zonas más peligrosas definidas por el usuario). El coste computacional de este método es tan alto, que se ha desacoplado su ejecución para no sobrecargar el front-end de la aplicación.

Además se contará con la posibilidad de enviar correos electrónicos, por si el usuario tuviera cualquier duda o sugerencia.

### **1.1 Objetivos**

Los objetivos del proyecto se describen a continuación.

El primero es la investigación de las diferentes tecnologías que se utilizarán en este proyecto, para la creación e integración de éstas en el servicio web. Tendrá que tener una apropiada interacción con el sistema y el despliegue de sensores, hay que tener en cuenta las diferentes formas que el usuario pueda interactuar con el sistema por medio de los diferentes dispositivos. Ya que cada vez hay más formas de conectarse para la utilización de aplicaciones como esta. Se quiere crear este servicio para obtener información desde cualquier dispositivo con una conectividad a internet, y un navegador que soporte la visualización del sistema en el dispositivo. Una vez realizado este estudio se decidirán que tecnologías serían las más adecuadas, además de poder ser utilizado en cualquier dispositivo.

Este estudio tiene un especial interés en el proyecto, porque puede contextualizar perfectamente la tecnología utilizada, y así obtener la mejor decisión para la implementación. Con esto se realizará la base para

implementar todo el sistema, además de proporcionar información al usuario de cuáles serían las posiciones de los sensores con una distribución robusta. Por medio de esto se podrá promover la utilización del sistema, así se podría probar la instalación de los sensores y su comunicación robusta.

Para llegar a una mayor cantidad de usuarios, se creará una implementación sencilla y teniendo en cuenta el diseño, proporcionar un fácil manejo del entorno al usuario. Podrán interactuar con una herramienta proporcionada en el servicio web, además de determinar un área y así decidir enviarla para el despliegue, en el cual, el sistema se encargará del procesamiento de despliegue de sensores. Para este problema, se realizarán diferentes pruebas con los distintos navegadores para un rendimiento óptimo de cada dispositivo. La interfaz está diseñada para que se pueda utilizar en los dispositivos que se vayan incorporando al mercado.

El segundo es el desarrollo del servidor, el cual recibirá las peticiones y además hará el cálculo para el despliegue. Para esto se definirá un lenguaje donde irá un algoritmo evolutivo. Se tomaron algunas indicaciones para un desarrollo más optimizado con algoritmo evolutivo que fue recomendado.

Por último se hará la implementación del algoritmo, se tendrá en cuenta que se necesita un servidor potente para el procesamiento. El algoritmo habrá que modificarlo para que reciba los parámetros necesarios con los cuales podrá optimizar las posiciones de los sensores, y a continuación devuelva lo esperado. Para así poder proporcionarlo al usuario y que él pueda visualizar en el ordenador el área donde se encontrarán los sensores de forma óptima.

## 1.2 Metodología

La realización del proyecto constará de las siguientes fases:

1. Estudio de los diferentes lenguajes de programación y tecnologías para el desarrollo del servicio web.
2. Decidir que lenguajes y tecnologías se utilizarán para su implementación.

## Capítulo 1 - Introducción

3. Crear el diseño del servicio web.
4. Desarrollar e implementar el servicio web para su uso.
5. Hacer pruebas en distintos dispositivos para un correcto funcionamiento.
6. Comunicación entre el mapa del servicio web y el *servlet* para la generación del despliegue de sensores.
7. Optimización del sistema web en distintos dispositivos y navegadores en los que las tecnologías son soportadas.
8. Pruebas necesarias para su perfecto funcionamiento de la declaración del mapa y el algoritmo de despliegue.
9. Evaluación de diseño del servicio web para una posible mejora para un futuro.
10. Conclusiones finales.



## Capítulo 2

---

### Conceptos Básicos

En este capítulo abordaremos los distintos lenguajes y tecnologías escogidas para el desarrollo del servicio web, se argumentan los motivos por el que fueron elegidos cada uno de los lenguajes.

Se podrá utilizar el sistema aquellos que tengan conectividad a internet y un navegador que soporte las tecnologías con las que se desarrollará el sistema.

Se contractará la necesidad que hay entre el *back end* y el *front end*, también se verá que lenguajes son los que se usan para cada uno de ellos y poder identificarlos en este proyecto.

#### 2.1 HTML 5

Es la última versión de HTML (*Hiper-Text Markup Language*), es el lenguaje utilizado para las páginas web de la aplicación, donde se han incorporado muchas mejoras de las anteriores versiones, se adjudicó como uno de los lenguajes principales.

Algunas de las mejoras que incorpora este lenguaje es un uso típico de los sitios web modernos, antes se utilizaban las etiquetas `<div>` para cada contenido que se quería realizar en el sitio web. En esta versión se han implementado algunas etiquetas nuevas y otras se han actualizado. Algunas de las etiquetas que se han utilizado son `<article>`, `<video>`, `<audio>`, `<canvas>`, `<nav>`, `<footer>`, entre otras etiquetas.

Estas definen algunos de los lugares en la web para la inserción de contenidos, esto hace que sea mucho más fácil de leer un servicio web al programador y para la verificación del robot de Google. Este robot se encarga de estar continuamente enlazando nuevos servicios, que se están incorporando en servidores de internet cada pocos segundos. Estas etiquetas



que se han incluido en el servicio web, hacen que al desarrollador le sea más sencillo enlazarlo con diferentes lenguajes de programación. Las etiquetas utilizadas se han situado en la estructura de la web, para una mayor integridad con los componentes.

Gracias al lenguaje HTML 5, los usuarios puedan visualizar lo necesario para interactuar con el sistema, además hace de esqueleto para los lenguajes que se van a utilizar. Con la utilización de diferentes lenguajes que puedan interpretar en la parte del cliente (front-end), con una carga mínima en el cliente, entre otros beneficios.

Esta ha sido la estructura elegida para todas las webs del proyecto:

```
<html><head>  
  
<title>Smart Cultivation</title>  
  
</head>  
  
<body>  
  
<nav></nav>  
  
</body>  
  
<footer></footer>  
  
</html>
```

## 2.2 CSS 3

CSS3 ha sido el lenguaje utilizado para definir los estilos y disposición de los contenidos incluidos en HTML, para una mejor manipulación del diseño. En la utilización de este lenguaje, la programación se divide en diferentes módulos para una mayor facilidad de utilización y de revisión del código, en el caso de

tener que modificar una parte del código o añadir alguna instrucción nueva será más sencillo.

Un ejemplo sería este:

```
p.serif { font-family: "Times New Roman", Times, serif;}
```

Este código lo que interpretaría es: dentro del código HTML que se le tiene asignado este estilo, todas las etiquetas p con la clase *serif*, y el texto que este dentro de esto, se modifica al tipo de letra que tiene este ejemplo. Empezando por la primera fuente, en la cual si el sistema no tiene por defecto el primer tipo de letra, colocaría el tipo de fuente segunda y así sucesivamente.

Algunas novedades que presenta este lenguaje son bordes, fondos, sombras sobre textos, transformaciones de elementos, capacidad de añadir fuentes en vivo entre otros. Esta es la última versión del lenguaje CSS y se han añadido algunas mejoras, como es poder cambiar el tamaño, dependiendo del dispositivo que está reproduciendo el servicio web, esto ha sido muy útil para la manipulación y la distribución en todos los dispositivos. Donde se pretende mostrar la información más relevante, además de las herramientas que se han incorporado en esta página web.

En este proyecto se han creado tres reglas para cada uno de los distintos dispositivos que hay ahora en el mercado, esto se hace por la capacidad de pixeles que soporta el dispositivo.

Uno es para los ordenadores de sobre mesa o portátiles, donde se puede mostrar toda información y herramientas que se han incorporado en el sistema web.

La siguiente regla es para las tabletas, que tienen suficientes pixeles de ancho en la pantalla, para mostrar toda la información que se ha añadido al servicio web.

Una tercera regla que ahora es la más utilizada son para los *Smartphones*, esta regla restringe algo de información, donde mayormente son imágenes que

no son relevantes. Gracias a este lenguaje se podrán manipular los estilos de todo el servicio web de una forma extraordinaria.

Por esto se ha contado de varios archivos, donde se encuentran los comandos para las partes del diseño de las distintas páginas webs. Cada una de las reglas del servicio web, modifica el estilo cuando aparecen en el árbol DOM, es lo principal para realizar el aspecto que se ha querido en el diseño.

### 2.3 Bootstrap 3

El objetivo de utilizar esta herramienta es tener una colección de reglas ya creadas en CSS, donde se le añaden en las clases de las etiquetas. Éstas se activan para modificar la estructura, dependiendo de los pixeles y el dispositivo donde se esté reproduciendo. La ventaja de usar esta herramienta radica en la cantidad de funcionalidad que trae implementada, permitiendo así reutilizar mucho código.

Es una colección de clases para manipular los sitios webs que contienen HTML y CSS. *Bootstrap* se ha elegido para la creación del servicio web, donde no hay que crear cada una de las reglas que se necesitan para la manipulación del estilo, antes se investigará si viene incorporada la función que se necesita. También hay muchas de ellas que se encuentran y hacen algo parecido a lo que se busca, entonces estas se modifican para un uso requerido para la acción que se requiere.

Para la adquisición de información se ha investigado en un libro llamado (Spurlock, 2013). Además de obtener información de distintos lugares alojadas en internet como (team), (), (Surguy), (bootply).

Otra por las cosas que se ha elegido esta herramienta, es que en su última actualización, que es la que se utilizará, se recomienda comenzar el diseño por un dispositivo Smartphone, para así poder ir ampliando la densidad de pixeles hasta llegar a una pantalla de ordenador. Esto hace que pueda reproducirse en los dispositivos, que es lo que se pretende con este proyecto.

La ventaja de empezar por el dispositivo Smartphone, es que se puede diseñar para que no se muestre la información no tan relevante con un diseño más atractivo. Para después ir aumentando el tamaño de la pantalla “*Píxeles*” hasta conseguir el dispositivo más grande, que en este caso es la pantalla de ordenador o una Smart TV. Con esto se puede ir reorganizando estructuralmente el diseño, para que cada dispositivo muestre lo se crea más necesario y sin perder la estructura, ni el diseño estético que se pretende.

Gracias a la cantidad de clases, que son las que hace que se comuniquen con la librería de *Bootstrap* para modificar los componentes, para poder organizar la estructura del dispositivo donde se está mostrando. De la forma que se desea presentar el componente o componentes que se ha decidido mostrar en esta clase. Tiene unas clases determinadas para la modificación dependiendo de la densidad de píxeles del dispositivo, se han usado en casi todos los componentes que engloban la totalidad del servicio web, para poder reproducir los datos de una forma idéntica, además que el usuario pueda leer e interactuar con del servicio web adecuadamente con su dispositivo.

Para adquirir la última versión de ésta librería, se buscó en esta web (), que es la página oficial. En ella se pudo descargar la última versión, que es la versión 3.1.1. En esta versión se han encontrado muchas herramientas útiles, para la modificación de la estructura del diseño, que han servido para el desarrollo de este servicio web. Si no estuvieran ninguna de las herramientas que se necesitan implementadas, habría que invertir mucho tiempo para conseguir lo que ya viene implementado en *bootstrap*.

Estas son las herramientas que se han utilizado de dicha librería:

- *Navbar-wrapper* es para diseñar un menú, donde ya hay creado un estilo y un tamaño de letra, en este se cambiaron algunos estilos para los que se han modificado para personalizar de una forma única este menú. Además se han usado otras clases para que el menú, se reestructurara de forma que se pudiera reproducir adecuadamente en cualquier dispositivo de los mencionados anteriormente. en el momento que un dispositivo no tuviera la densidad de píxeles adecuada, los botones se ocultan para dejar paso a un solo botón, el cual tiene la propiedad de deslizar los botones que se ocultaron, donde estos son los

encargados de dirigirse a los distintos apartados del servicio web. Este botón único se incorpora en la parte derecha, esta acción también se aplica gracias a otra clase que tiene implementada *Bootstrap*, amoldándose al dispositivo que muestra este despliegue de menú. esto hizo no necesitar implementar muchas acciones que se realizan con CSS.

- *Carousel* es otra clase que se ha utilizado para la visualización de imágenes en el diseño de la página Home. Aquí se encuentran la muestra de algunas fotografías y algún ejemplo de cómo se vería una vez desplegado todos los sensores. Además de ver los polígonos creados para que esto sea lo más real posible, esta clase es la encargada de pasar de imagen cada cierto tiempo, incorpora un botón a cada lado que se oscurece para saber que puede presionarse sobre él, así pasar a la imagen siguiente o anterior.

Estos botones están a la derecha e izquierda de la imagen, también incorpora tantos botones como imágenes se incorporen en el carrusel creado, para poder mostrar directamente la imagen que tiene enlazada los botones creados. la propiedad la tiene pulsando para mostrar en el carrusel dicha imagen, con la clase *Carousel* también se puede incorporar información relevante, por delante de la imagen, como texto u otra información que se desee. Para mostrar información ha sido bastante importante, ya que se deseaba mostrar información en la página principal, para que el usuario sepa cuál es la aplicación que tiene este proyecto y como puede abordarlo con su dispositivo.

- *Container Marketing* esta clase se ha utilizado para toda la información que se quería proporcionar del proyecto, además de mostrar los beneficios que tiene el poder utilizar los sensores en este sistema, con su sistema de riego. Aquí se han utilizado distintas imágenes, se han colocado para una visión más directa de lo que trata este proyecto, como se podrá observar en los distintos dispositivos, no se ven todas las imágenes en todos los navegadores.

Con el lenguaje *Javascript* se han incorporado unos anclajes en unos textos, para poder ver una información detallada de la imagen. Esto se

menciona porque también se ha utilizado *Bootstrap* para el deslizamiento vertical, el cual tiene desde que se le da al botón mencionado, hasta que llega a la parte de información requerida.

Por la utilización de esta clase, también depende del dispositivo que se esté visualizando, se amolda a él para poder leer toda la información que se ha querido presentar. Se colocan las imágenes y el texto de forma vertical, ya que los dispositivos como Smartphone tengan la posibilidad de deslizar verticalmente y así poder leer toda la información que se ha incorporado.

- *Img-circle* las imágenes que se le indiquen en esta clase las presenta redondas para una visualización más estética en el servicio web, así el usuario le interese más lo que se está mostrando. Con la utilización de otra clase estas imágenes se colocan a la izquierda o a la derecha, llamando a estas librerías: *pull-left* y *pull-right*, de esta manera quedan mejor colocadas para poder leer la información, sin tener mayor dificultad de visualización.
- Con las clases *col-sm* se decide cuantas filas se quiere que tenga en el navegador, para así amoldarla al dispositivo que se está reproduciendo. se ha utilizado para unas pequeñas imágenes, tienen más espacio o menos, dependiendo del dispositivo donde se reproduzca, son las que están en la parte inferior del servicio web.
- *Navbar-static-bottom* se ha utilizado para el *footer* de todas las webs requeridas, nos proporciona la seguridad, que la clase *footer* estará en la parte inferior del servicio web siempre. En esta clase se ha modificado para no estar completamente pegada, la clase *footer* a la parte inferior del diseño del servicio web y así pueda quedar estáticamente mejor.

## 2.4 AngularJS

Es un *framework* para aumentar la interactividad del front end, para aplicaciones basadas en MVC. Este *framework* está basado en *Javascript* y es código abierto, además esta mantenido por Google.

Fue un *framework* que tenía muchas posibilidades de ser utilizado para este proyecto, ya que cumplía muchas expectativas que se buscaban, además de ayudar con la gestión que se conoce como aplicaciones de una sola página.

Se empezó a estudiar en un libro llamado (Sebadri, 2013) y además sacar información y ejemplos para poder incorporarlos de este otro libro (Knol, 2013). Se ha encontrado mucha información para poder usar este lenguaje para el proyecto desde el principio en un blog (Azure, 2013), también se investigó en otra web (jmcunningham, 2013). Como se pudo ver se le podían implementar muchas aplicaciones para la funcionalidad y beneficio en este proyecto.

Después de haber obtenido información de todos los sitios mencionados anteriormente, se decidió no incorporar este *framework*, ya este proyecto no iba a tener una gran influencia con bases de datos o datos que tratar, para una inyección de dependencia. Por lo que este proyecto se ha basado más en la funcionalidad con el entorno por parte del cliente. Ya que este lenguaje está basado en la manipulación de datos, como por ejemplo de una base de datos, se decidió que no se tenían tanta información como esta para mostrar.

## 2.5 Javascript

Es un lenguaje de programación integrado, se define como orientados a objetos, es muy utilizado sobre todo en la parte del cliente, además es interpretado. Ha sido muy importante en la definición de funciones, para poder crear operaciones necesarias en la creación de este servicio web como se ha diseñado, además de incorporarle movilidad y no ser estática. Es importante ya que se va a utilizar una biblioteca con muchas funciones que están creadas con *Javascript*. Este lenguaje tiene la propiedad de crear clases y funciones en archivos externos. Así estas funciones pueden ser llamadas desde donde se necesiten dentro del proyecto, simplemente hay que tener

incorporado la dirección del archivo donde se encuentran las funciones que se desean utilizar. De esta manera se le define la dirección donde se encuentra el archivo, el sistema se encarga de compilar las funciones que tiene incorporadas, para posteriormente ser utilizadas desde cualquier parte del sistema, esto es lo que hace muy poderoso al lenguaje *JavaScript*.

La funcionalidad que proporciona este lenguaje es muy grande y por eso se ha elegido incorporarlo para la creación de este proyecto. Tiene mucha información en internet ya que este lenguaje tiene mucho tiempo, esto conlleva que hay muchas clases en internet. Hay mucha información que ha sido aportada por otros programadores con conocimientos de este lenguaje, con distintas funciones y distintas respuestas de preguntas que van surgiendo, mientras se ha estado desarrollando, algunas de las web, blogs o foros donde se ha buscado información son, (w3schools), (Bidelman), (MDN), (anieto2k).

Para implementar los mapas de Google ha sido necesario además de indispensable este lenguaje, ya que toda la funcionalidad que tiene está implementada con *Javascript*, para cualquier método que se tenga que acceder o implementar en el sistema de mapas o implementaciones.

Se han implementado varios archivos con *Javascript* para la creación del proyecto, se han obtenido algunas librerías de usuarios que han creado y han subido a internet para poder ser utilizadas. Algunas referencias de las que se han utilizado son, la librería: (BlitzMap), se han recogido las funciones que se van a utilizar, en vez cargada toda la librería, tiene también el problema que es más pesado para el procesamiento del servicio web. La librería (Twitter) son las transiciones que tiene *Bootstrap* el *framework* hablado anteriormente, (Robbin), (Klinggaard) son otras de las librerías que se han utilizado para comportamientos del sistema.

## 2.6 JQuery

Es una biblioteca de *Javascript*, que permite simplificar la manera de implementar documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX. Es un



software libre y código abierto, esto hace que sea tan utilizado en la gran mayoría de las web que se están creando desde hace varios años.

Esta herramienta se está utilizado mucho en el proyecto, ya que para el manejo del árbol DOM se utiliza mucho, esto ocurre en la mayoría de los apartados de este servicio web. Se utiliza mucho para las acciones de botones que están incorporados en el sistema, la acción la realizan, por una función que se encuentra en otro archivo, el cual se incorpora por una instancia hacia él. Las funciones son más sencillas de ejecutarlas desde uno de los botones, con la utilización del árbol DOM. Así no se necesita generar tanto código fuente para el funcionamiento del sistema.

Igual que se utiliza para los botones, se utiliza para otras implementaciones que han sido requeridas, un ejemplo es en la incorporación del mapa de Google. Se ha necesitado esta librería, para todas las acciones y llamadas que se hacen a cada una de las herramientas que se han incorporado en la utilización del mapa.

Se ha incorporado una librería de *JQuery* que se llama localmente para una mayor velocidad de respuesta, esta librería también tiene la propiedad de obtener funciones que se encuentran en otros archivos.

Esta librería es *Javascript*, en este proyecto se está utilizando más *JQuery*, para reproducir la función definida. Para que esta función pueda estar operativa, tiene que inicializarse con una relación hacia esta de la librería, que se cargue en el servicio web y así poder ser llamado desde cualquier función.

Para obtener la máxima información sobre esta biblioteca, se han estudiado dos libros, (Experts, 2010) y (Flanagan, 2011), además también se ha recogido mucha información y ejemplo de numerosos blogs, webs y foros dedicados a la comunidad de creación de webs con *JQuery*. Aquí se dejan algunas de las referencias más destacadas, (Varios), (Stackoverflow), (desarrolloweb.com), entre otros., Se van a utilizar varias librerías de otros autores, que han sido muy útiles para poder desarrollar funcionalidades en el servicio web. Estas librerías son algunas de las que se han obtenido para la implementación de funciones requeridas, (Robbin), (Klinggaard), (Simpsons) (Malopinsky), y se ha

incorporado la librería de *JQuery*, que se obtiene por el enlace de una dirección de las Apis de Google.

## 2.7 Ajax

Esto es una técnica de desarrollo web para crear aplicaciones interactivas, se ejecuta en el cliente, pero mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma se pueden realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad de las aplicaciones. Es una tecnología asíncrona, esto significa que solicita los datos al servidor y se cargan en segundo plano, sin transferir con la visualización ni el recargo de del servicio web que está utilizando el usuario.

Es muy útil para usar funciones de lenguajes que solo pueden ser aplicados en el servidor, además se pueden obtener los datos que devuelve la función o funciones a las que se les hace la llamada, por eso es tan importante en algunas ocasiones utilizarlo. Esto hace que se hagan llamadas al servidor, mientras el usuario está utilizando alguna herramienta o visualizando algo en el servicio, por este motivo también se ha incorporado para la creación del servicio web. Es de gran importancia para hacer la transferencia que hace con al servidor, y así poder hacer llamadas a funciones que se necesitan del para una seguridad me mayor rango.

Esta herramienta se utilizará para hacer llamadas a archivos *PHP*, estos se interpretan en el servidor, y por esto también será necesario para el traspaso de datos entre los dos servicios que se necesitan en este proyecto. Se utilizará para la carga de los mapas, y además se podrán hacer conexiones para transmisión de datos al servidor. Esta herramienta nos proporciona además una seguridad, ya que al enviar los datos al servidor, el usuario no sabe dónde o que parámetros se han requerido, ya que las conexiones se harán entre servidores.

También se utilizará para, proporcionar una comunicación con los archivos en *PHP*, que es el lenguaje que va a ser utilizado, para la verificación de datos y poder enviar los que necesita el servidor de despliegue de sensores.

## **2.8 PHP**

Este es un lenguaje diseñado para desarrolladores web pero también usado para uso general como lenguaje de programación. Este lenguaje tiene bastante tiempo, y por eso es utilizado por los desarrolladores web, ya que este se reproduce en el servidor, esto tiene una gran ventaja, seguridad y velocidad, entre otras, tiene una seguridad, ya que el usuario no puede interactuar directamente con este lenguaje, al ser reproducido en el servidor. Los datos que se calculan o se reproducen, el usuario no puede acceder a ellos, la velocidad la da por el hecho que, normalmente un servidor es más rápido al reproducir un código, que también puede necesitar una mayor carga, por los algoritmos que se ejecutan para la obtención de datos.

Este lenguaje también es muy utilizado para la explotación de la base de datos, además tiene otra propiedad, poder guardar variables en sesiones, para la operatividad con ellas cuando se están haciendo llamadas al servidor, como datos de conexión para saber que el usuario es el que se ha conectado, esto es una ventaja al no necesitar *Javascript*, ya que el usuario puede tenerlo desactivado.

Para este proyecto se utilizará, sobre todo para la validación de formularios y variables necesarias para la creación del archivo KML (Mapa de áreas), que es necesario para obtener el mapa para el usuario, también se podrá utilizar para la creación de los propios algoritmos, el *parseo* y *desparseo* del archivo KML, esto se hace para que el usuario no pudiera manipular esta salida de archivo, y que el sistema cometiera errores. Con este lenguaje se hará la conexión con el servidor para el despliegue de los sensores, como se ha mencionado antes. El usuario no sabe qué datos se le están enviando, esto hace que no pueda manipularlos para que el sistema cometa errores.

Las conexiones que se realizaran, harán que se aprenda como funciona este lenguaje con distintas conexiones, ya que un servicio web bien diseñado tiene que tener velocidad además de seguridad. Al poder utilizar este servicio en cualquier dispositivo se necesitaba tener un equilibrio, que conseguirá gracias al poder hacer comunicaciones entre diferentes lenguajes.

## 2.9 Java

Este lenguaje de programación fue diseñado para las ejecuciones concurrentes, está basado en clases, orientado a objetos y tiene dependencia de implementación. El código generado tiene que ser compilado antes de poder ser procesado, una cosa interesante que tiene java es que puede ser ejecutado en cualquier maquina java (JVM), es uno de los lenguajes de programación más populares en uso, particularmente para cliente-servidor en aplicaciones web.

Gracias a la implementación que tiene este lenguaje con cliente-servidor, se pueden enviar datos para que JVM reciba los datos y pueda procesarlos en el lenguaje java, ya que es mucho más rápido y seguro ya que está en el servidor. Se puede ejecutar una gran cantidad de procesos, que se generan en el servidor sin tener que generar estos procesos en la parte del usuario.

Gracias a la implementación de este lenguaje que se hará en nuestro sistema, se podrán realizar las operaciones tan sumamente pesadas que se requiere. Para la obtención de este despliegue robusto, al recibir los datos del lenguaje de *PHP*, habrá que *desparsear* los datos que vienen del servicio web al servidor de optimización. En el servidor de optimización se harán los cálculos oportunos de cada una de las áreas que el usuario enviara, para pasarlos a una matriz, donde se hace un proceso de selección de las distintas áreas con más peligrosidad, comprobando que están encima del área principal “Verde”.

El terreno seleccionado por el usuario donde se ubicaran los sensores será el área principal, la cual se le proporcionara al sistema para la generación de la matriz y posteriormente realizar el cálculo del despliegue robusto.

Se va a utilizar un *framework*, que está diseñado para hacer cálculos *metheurísticos*, además de tener la programación evolutiva para una mayor optimización y robustez a la hora del despliegue de los sensores. Este *framework* se llama JMetal, para el cálculo del despliegue de los sensores necesarios para cubrir el área, que el usuario ha marcado, además de hacer el cálculo con los distintos niveles de peligrosidad del área.

## 2.10 JAX-RS

Es una API del lenguaje Java, que proporciona soporte a la creación de servicios web de acuerdo al estilo arquitectónico REST, se ha elegido Jersey para esta implementación, ya que tiene las herramientas necesarias para la utilización que se necesitaba.

Para la utilización de esta herramienta, se implementará un servidor *Tomcat* para realizar las llamadas, esto se creará en el entorno de programación Eclipse. Una vez se han insertado las librerías necesarias para el uso de esta herramienta, se generará un XML, el cual se incorporará en la llamada para poder hacer el *parse* de este. En este caso, será una clase que se encargará de la optimización del servicio, esta llamada se hará a través del protocolo Post, al ser un servicio basado en HTTP es fácil hacer la llamada de esta manera, se enviarán los datos a través de este protocolo para mayor seguridad. Ya que el protocolo *Get*, que es otro protocolo que tiene el servicio HTTP, este envía la llamada a través la barra de direcciones, y esto hace que sea más inseguro para la llamada.

Con este sistema también se está controlando, que llegue la llamada a través del método Post y no por *Get*. Si llega alguna llamada por *Get*, se mostrará un mensaje donde pondrá que no se acepta por este método

## 2.11 JMetal

Es un Framework orientado a objetos, para una optimización de objetivos múltiples con metahurística. Este sistema se ha utilizado para el cálculo de una matriz o matrices, cada matriz puede tener unas dimensiones muy amplias, esto depende del área que haya seleccionado el usuario, con la aplicación en el mapa del servicio web.

Un ejemplo, la matriz de un área de unas dimensiones normales, puede llegar a ser de 2000 x 2000, esta matriz es de un área seleccionada de 8000 m<sup>2</sup>, se realizará un cálculo para obtener lo necesario, para que el sistema de optimización lo reciba, además con un sistema de optimización potente, ya que se tienen que calcular las diferentes posibilidades de los sensores y el cálculo de las evoluciones, que consiste en comprobar la conectividad

desactivando sensores, esto conlleva que se necesite un buen sistema de optimización, como un cómputo potente para poder generar la matriz con los sensores colocados de forma robusta.

Este algoritmo evolutivo se ha elegido basándonos en el proyecto de fin de carrera de (Espada Jiménez), se ha podido sacar mucha información sobre como desplegar robustamente sensores en una matriz. Este proyecto también proporciona muchas más opciones de optimización robusta, pero se ha utilizado el que está dentro de la carpeta problemas robustos.

*WSN\_robust* es el que se va a utilizar, ya que en esta clase, tiene creado un ejemplo de despliegue de sensores robustos, se han tenido que hacer varias modificaciones, ya que el apartado de ejemplo que tiene generado, son variables fijas.

Para este proyecto se necesitaba enviar diferentes variables, ya que además de mandar diferentes tamaños de áreas, también se le da la posibilidad al usuario de hacer su despliegue con distintos parámetros, entonces hubo que crear una clase diferente con una solución al problema que se necesitaba para el cálculo del despliegue de sensores robustos.

Este nuevo problema del despliegue de sensores con diferentes áreas de peligrosidad, crea es un array de matrices por cada uno de los diferentes áreas que el usuario desea hacer el despliegue. Con estos se sabrá cuáles son las posiciones del mapa donde tienen que colocarse los diferentes sensores, el resultado se envían al sistema Java, es quien genera el cálculo donde se colocarán los diferentes sensores. Este sistema también se encarga de crear el o los archivos *Kml* necesarios, donde están puestos las diferentes áreas y los diferentes sensores.

También se encarga de generar archivos con los sensores elegidos por el sistema de optimización, para que el usuario pueda llevar la posición de latitud y longitud a un entorno real, con esto poder colocarlos exactamente donde el sistema ha decidido.

Una vez terminado los archivos, se le mandan al usuario al correo electrónico que decidió en el formulario de despliegue, donde podrá cargar el mapa y ver

el sistema el despliegue generado, con las áreas que este usuario decidió incorporar.

Esto se asemeja a la realidad ya que el mapa que se utiliza es de Google y son fotografías tomadas desde un satélite, aunque en el sistema se puede elegir con qué tipo de mapa se desea operar.

## **2.12 Entornos de trabajo utilizado**

### **2.12.1 Dreamweaver**

Para la creación de las páginas web que aloja el servicio web. De todos los entornos que se investigaron, se quería escoger uno que soportara la mayoría de los lenguajes, para ser apoyados por el entorno, se escogió Dreamweaver V.14 CC. Porque tiene un entorno conocido, además bastante amigable para la comprensión con las herramientas, los lenguaje que mejor se utilizan con este entorno, son el HTML 5, CSS 3, *Javascript* y *PHP*. Que son casi todos los lenguajes que se van a utilizar, una propiedad que tiene Dreamweaver, es que ayuda al programador, con los detalles de cada lenguaje soportados.

Se sabía que tiene una herramienta la cual va marcando los errores en la sintaxis de la línea programa, que se van cometiendo con cada lenguaje, también tiene la propiedad de tener un entorno amigable, para la navegación entre archivos del proyecto.

Para mayor facilidad tiene otra herramienta que permite la transferencia de archivos, está bien conseguida ya que solo muestra los últimos archivos que se han actualizado en el servidor o en local. Esto conlleva que cada vez que se le da a actualizar los archivos, no tengan que hacer la subida de todos ellos para probar alguna actualización.

Tiene además otra herramienta muy potente para el manejo de CSS, ya que tiene un panel, donde aparecen todos los archivos alojados, que están en la página donde se está desarrollando. Tiene un panel para poder saber que selectores de CSS tiene la página en la que se está programando, puede elegirse que muestre solo los que tiene alojado ese archivo, o los selectores de todos globalmente.

También incorpora un panel donde están todas las propiedades del selector que se seleccione, con esto muestra de forma gráfica todas las herramientas que tiene o se pueden hacer con este selector. Tiene una manera muy sencilla de añadir selectores, a todas aquellas etiquetas que se deseen, para que tengan la propiedad deseada. Cuando se tiene abierto un archivo HTML, al tener el link a distintos archivos CSS, *Javascript* o *PHP*, tiene un submenú de archivos para que se pueda llegar al archivo que se requiera, de forma instantánea al elegirlo en el submenú.

### **2.12.2 Photoshop**

Este entorno se ha utilizado para la creación y modificación de imágenes del servicio web, además de diseñar algunas para su exposición. Este entorno está hecho para el diseño de imágenes, por esto se eligió este para su desempeño.

### **2.12.3 Filezilla**

Este entorno se utiliza para subir gran cantidad de archivos al servidor, hay veces que se necesita subir todos los archivos del servicio web o un archivo que tiene un tamaño elevado. Este entorno tiene muy bien conseguido la subida de archivos masiva, es capaz de subir varios archivos de forma simultánea, además tiene un entorno gráfico fácil de manipular, es muy rápido ya que es un programa que no tiene mucha carga en el sistema.

### **2.12.4 Xampp**

Este programa se ha utilizado para crear un servidor en el sistema operativo que se está utilizando, ya que es más rápido probar una actualización o corregir errores en un entorno interno de la red. Con esta herramienta, una vez que se haya probado la funcionalidad, hay que subirla a un servidor real. Ya que los servidores, pueden tratar las herramientas o las funcionalidades de forma diferente, pueden tener un archivo configurado de distinta manera al que está en la red interna. Por esto es también es importante subir al servidor



y probarlo, este proyecto se ha probado en dos servidores diferentes, para intentar suplantar los máximos errores posibles.

### **2.12.5 Eclipse**

Por este entorno se ha podido implantar un servidor, en el cual se requería reproducir una maquina Java, se ha podido desarrollar el código y reparar lo necesario con una herramienta que tiene. Esta herramienta se llama *debug*, se han podido manipular los errores que daba cuando se probaba en el sistema.

Para la manipulación de clases este entorno tiene una gran potencia, además de poder insertarle *plugins* para las funcionalidades que se necesiten, esto ha hecho que sea más sencilla la modificación del código del algoritmo que se manipula.



## Capítulo 3

---

### Análisis

Se propusieron varias ideas por las que se quería encaminar el proyecto. Estaba bastante claro que iba a ser orientado al diseño y creación de un servicio web donde pudieran interactuar usuarios con él. También se quería que las tecnologías que se utilizasen, estuvieran entre las más actuales del momento.

El objetivo es utilizar un algoritmo evolutivo el cual calcula algo parecido a lo que se necesitaba, donde a partir de un área determinada y un conjunto de sensores, los despliegue por la zona que generó el usuario para que su cobertura sea lo más robusta posible. Se entenderá por robustez la cualidad que permite que tras la caída fortuita de un sensor, la región definida pueda seguir estando cubierta.

El servicio web requerido, conectará a un conjunto de usuarios potenciales con un servidor encargado de ejecutar el algoritmo de despliegue.

Se empezó a dar forma a la idea, creando una página en la que el usuario pudiera interactuar con el servicio web. Además, sería interesante que la aplicación web, tuviera información sobre el proyecto que se está ideando realizar y los beneficios que conlleva.

Se empezó a hacer bocetos para el diseño del servicio web, se requería simplicidad en la estructura, para no saturar de información o de elementos innecesarios en ella. Tener un apartado donde poder mandar un mensaje de información, en caso de falta de información en la aplicación web o cuando el usuario tuviera alguna duda más específica.

Uno de los elementos más relevantes sería el mapa, donde el usuario pudiera interactuar con él y poder señalar una región, para así desplegar los sensores en esa área elegida. Se había pensado además que el servicio web debía de ser 2.0, versión habitual actualmente en el desarrollo web, con la posibilidad de

redes sociales, personas que quieran saber más, o personas que ya lo hayan utilizado.

Para saber qué es lo demandando en el mercado del diseño de servicios web, se investigó en algunas redes sociales, además de en grandes compañías como Apple, Microsoft, Google, etc. Los apartados que lucían en las webs de estas compañías, eran todas muy interesantes, con lo que al final se decidió por adoptar el diseño de uno de los apartados de la web de Google. Se escogió la página que tenía el nuevo dispositivo el *Google Nexus 5* (Google).

El menú era el que se buscaba, que pudiera desplazarse con el deslizamiento del *scroll*, también tiene un menú que estaba en la parte izquierda que también era lo que se deseaba, así que se creó un boceto arreglo a lo que se vio aquí, además apoyándonos en el servicio web que se está comentando.

### 3.1 Primera Fase

Cuando se tenía una idea, del propósito que se quería conseguir en el servicio web, se comenzó a decidir que incluir en cada uno de los apartados del proyecto, siguiendo una serie de pautas.

**3.1.1.1 Primer apartado:** página de inicio, donde se describe brevemente de que trata este proyecto o servicio web, aquí se definirían las características principales por las cuales este servicio web se ha creado. Junto con una presentación de un despliegue de sensores, creado con *canvas* (herramienta de HTML5). Con un diseño donde fuese fácil de ver las explicaciones que se desean mostrar al usuario, además explicar brevemente que desempeño realiza este servicio web.

**3.1.1.2 Segundo apartado:** desarrollo de la página dónde estaría el mapa de *Google map*, que es donde el usuario tendría la posibilidad de interactuar con las herramientas del mapa, para crear uno o varios polígonos y determinar un área o áreas que desee. Estas regiones seleccionadas serán donde se desplegaran los sensores, a su vez pudiese guardar las áreas generadas, así poder cuando quiera volver a cargarlo. Así se le da la posibilidad al usuario de

poder seguir en otro momento, cargando antes los mapas guardados, con lo que una vez que se tuvieran las áreas determinadas, se pudieran manipular borrando o añadiendo nuevos elementos.

La forma de guardar y cargar los mapas es bastante intuitiva, pero solo se pueden cargar los mapas que se descargan desde este servicio web, ya que se recogen datos diferentes que no se recogen en otros sistemas. Por la parte del despliegue de sensores, habrá un botón que hará una petición al servidor donde esté alojado el algoritmo de despliegue, para el cálculo de las posiciones adecuadas de los sensores en esta área o áreas que se envían en la petición.

En esta llamada se envía la posición de cada marca del polígono y el tamaño del área en m<sup>2</sup>. Así se obtendría toda la información necesaria para el cálculo, además de las posiciones exactas de cada uno de los sensores. Para el cálculo de éstos, hay que tratar de asegurar que el área siempre esté cubierta, por al menos dos sensores. Incluyendo además el hecho de ser una red robusta, lo que indica tolerancia a caídas fortuitas de algunos nodos de la red. Se procede a la respuesta, que va dirigida nuevamente al servicio web, para poder recoger los datos e incorporarlos en el mapa.

**3.1.1.3 Tercer apartado:** donde estaría contenida toda la información, suficiente para que el usuario pudiera visualizar el despliegue de sensores generado en una zona agraria. También se necesitará un elemento explicativo el cual detalle el ahorro y el beneficio del despliegue de dichos sensores, se ha pensado en utilizar un video, pues es más fácil de comprender, además que proporciona un diseño más atractivo y dinamismo a la web. Este apartado debería contener una breve explicación de cómo utilizar las herramientas del mapa, para crear un área determinada, las diferentes opciones que aportan junto con la funcionalidad que añade cada una de ellas.

**3.1.1.4 Cuarto apartado** aquí se encontraría la forma de contactar con el administrador del sistema para comunicar alguna duda o resolver algún problema que el usuario tuviera, bien sobre la manipulación del mapa, bien sobre el despliegue de sensores. Se dispondrá de un formulario sencillo para poder rellenar y enviar. Se pediría al usuario, su nombre, email, finca y el mensaje que quiere proporcionar, controlados además para que no se puedan introducir campos erróneos. Una vez que se le da al botón proporcionado, el

sistema comprueba si hay algún campo que no corresponde a lo indicado, si está todo correcto, este formulario se enviará al correo de la persona responsable.

**3.1.1.5 Quinto apartado** Aquí los usuarios se podrán registrar, para el control de usuarios, donde se les pedirá nombre, apellidos, correo electrónico, clave de acceso, fecha de nacimiento y una imagen que el usuario quiera proporcionar. Con esto se recogerían los datos suficientes para tener constancia de la persona que se ha registrado. Esta información se guardará en una base de datos donde se hará una consulta para insertar al usuario.

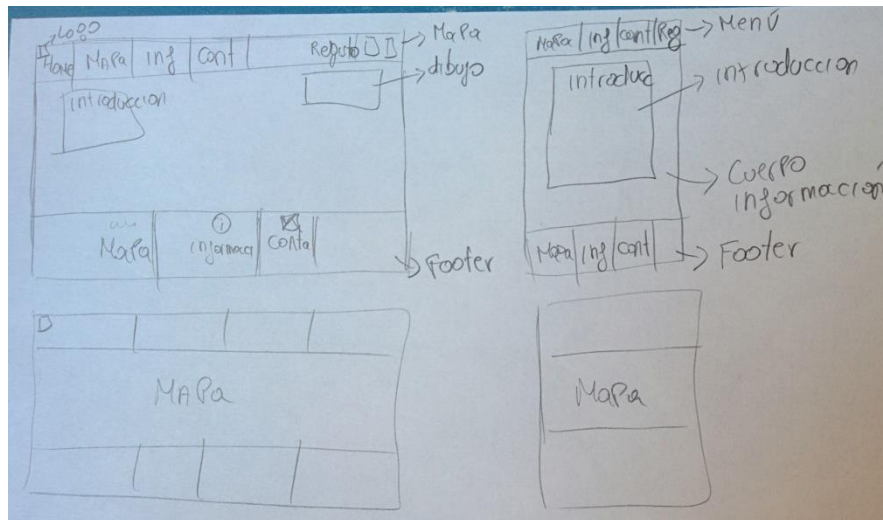
### **3.1.2 Diseño de web**

Para el diseño se comenzó realizando un boceto que pudiera recoger la mayor parte del servicio web, sobre todo indicando los apartados que no iban a modificarse en ninguna página. Entre estos elementos se encuentra el menú, el logotipo del servicio web en la parte superior izquierda, en la parte central irá la botonera y en la parte más a la derecha irá el botón de registro.

Seguidamente en la parte derecha del menú, se encontrarían los botones de las redes sociales. Entre el menú y el *footer*, irá toda la información o aplicaciones que se necesitará para el usuario, dependiendo del apartado en el que se encuentre. En los apartados home, información, contacto o registro, va todo lo necesario para que la persona se informe, y/o pueda manipular este servicio. Para el apartado mapa, se requiere ocupar toda la pantalla en el dispositivo, para una mejor visualización del sistema además de las herramientas, para la manipulación de polígonos.

El *footer*, tiene que tener las credenciales de quien lo ha creado, su correo, un teléfono de contacto y el copyright.

Con esto se hizo un primer boceto del servicio web, donde distinguía como sería el esqueleto de todos los apartados:



1 Diseño de web, primera fase

### 3.1.3 Diseño casos de uso

Se empezó haciendo el diagrama UML del servicio web, definió la manipulación del sistema para un usuario, para hacer de una forma más sencilla, además de diseñar el código de mejor forma. Para la creación de este boceto, tendría que tener incorporado lo descrito en los anteriores apartados.

**3.1.3.1 Home:** se mostrará a primera vista, que servicio está proporcionando este aplicativo web, que hace o implementa, a este apartado se le llamó Home.

**3.1.3.2 Despliegue de Mapas:** aquí se encontrará el mapa, donde se utilizará con las herramientas necesarias. Creación, manipulación y borrado de áreas, además de herramientas proporcionadas para la manipulación del mapa. Estas áreas creadas podrán ser guardadas en un archivo y cargarlas en este apartado. Estas herramientas proporcionadas son para desplegar los sensores, se podrá observar de la forma más real por la funcionalidad del mapa, se verán los sensores desplegados y el área que se creó.

Herramienta **despliegue de sensores:** para el uso tiene que haber una transmisión de datos, entre el servicio web y el servidor donde se encuentra el algoritmo de despliegue, esto consiste en hacer una llamada al servidor donde se encuentra este algoritmo. Se mandan las coordenadas de cada una de las áreas, el tamaño en m<sup>2</sup> y otros parámetros, que el usuario puede elegir de

forma avanzada. De esta manera el algoritmo recoge los datos necesarios para generar el cálculo, así el algoritmo empezará a generar una posible solución. Una vez tenga calculado la posición de cada sensor, devolverá la petición.

Se pueden recibir dos tipos de peticiones: con los datos que el usuario ha enviado, si el algoritmo no llega a un cálculo satisfactorio, se enviará un correo indicando que no se ha podido realizar el despliegue. La otra petición será la satisfactoria, enviando dos archivos como adjuntos, uno son las posiciones de los sensores, con sus latitudes y longitudes, es el mapa que se podrá delegar en la aplicación.

Además se envía una petición de vuelta al servicio web, donde se muestra un mensaje en el que dice que recibirá un correo electrónico, para que así el usuario, pueda saber que la petición del despliegue se ha realizado correctamente. Esto ocurre si el servicio web sigue abierto, de lo contrario si no es así, no recibirá el mensaje de que le llegará en breve un correo, pero si recibirá en el correo correspondiente. El despliegue del mapa en el servicio web haría que el usuario pudiese ver de forma más real como quedarían colocados.

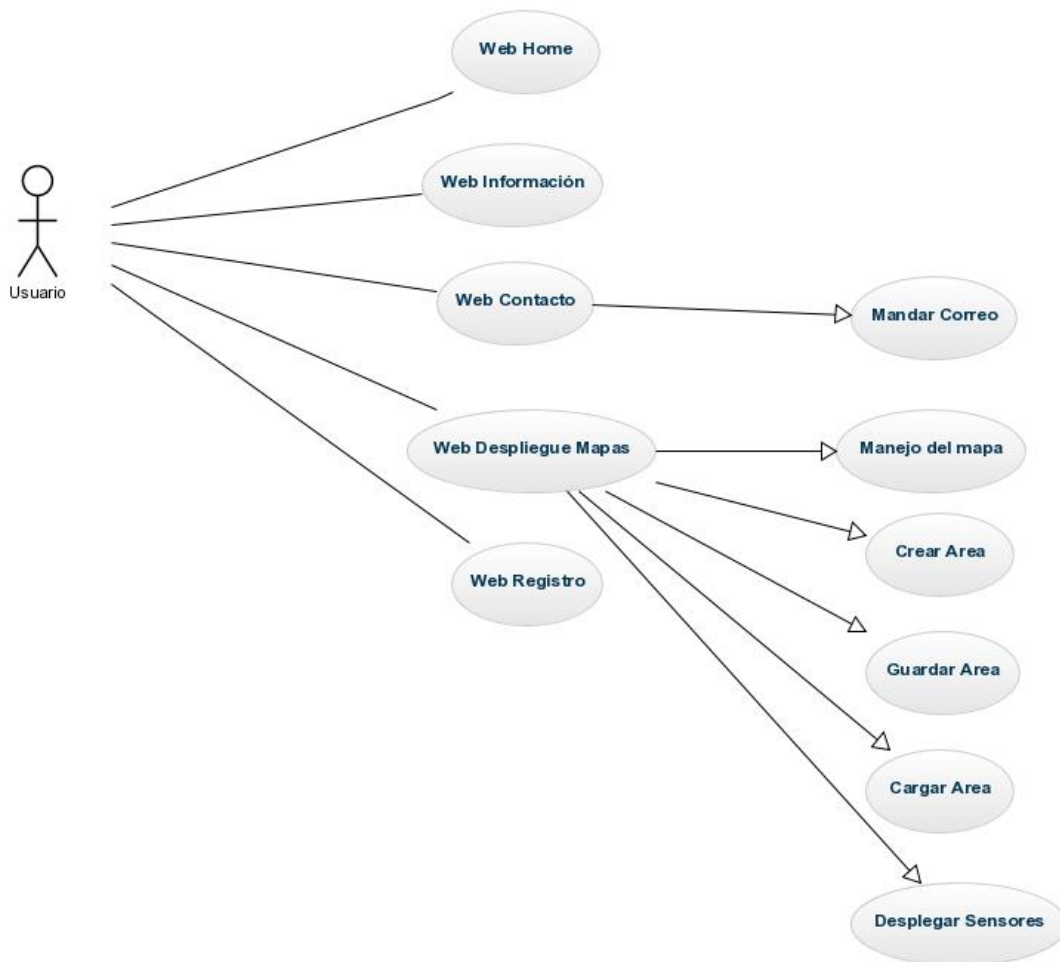
**3.1.3.3 Información:** apartado el cual muestra la información que necesita para saber cuál es el motivo de este servicio, además de tener las instrucciones para poder manejarlo. Se informaría de los beneficios de agregar estos despliegues de sensores en un área agraria y el ahorro considerable que tiene. Se quiere proporcionar un video explicativo, donde se mostrará el beneficio en un área agraria real, se quería recoger la forma de usar la herramienta de mapa en el siguiente apartado, así proporcionar una orientación a los usuarios.

**3.1.3.4 Contacto:** se crearía una forma de contacto, el cual pudiese ayudar en cualquier duda o problema que tuviera sobre este servicio. Sería una forma externa al sistema, el sistema mandaría tal correo y las comunicaciones siguientes se harían por otras vías.



**3.1.3.5 Registro:** es para el registro de usuarios, solamente se le pedirían los datos del usuario, para tener una información de los usuarios que están en el sistema. La aplicación se ha realizado para que se pueda utilizar sin registro, esto hace que abarquemos muchos más usuarios.

Este es la implementación del servicio web, esquema UML:



## 2 Diseño de caso de uso, primera fase

### 3.1.4 Imagen de web

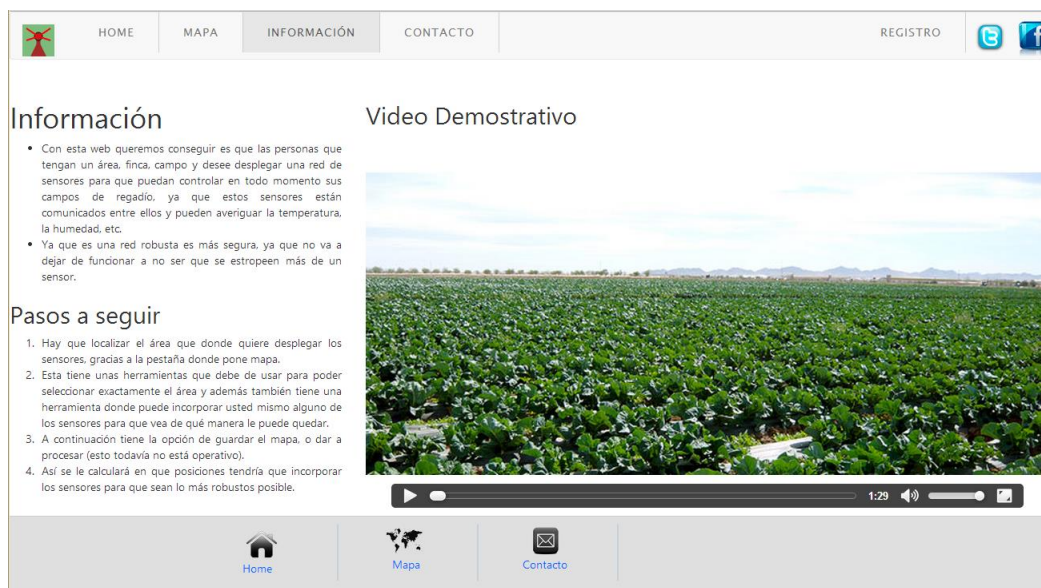
Con el boceto que se presenta, se empezó a diseñar los componentes descritos en el apartado anterior, donde el servicio web se asemeje a este boceto, para esto se hizo el proceso de diseño. Se creó un fondo simple, para que los usuarios se centren en la información que había en el apartado home incorporada. Así se desarrollaron los otros apartados, para que el usuario le parecería más bonito estéticamente, además que le fuese más sencillo el poder

## Capítulo 3 – Análisis

navegar por el sistema web. El mapa tendrá que albergar todo el ancho posible como se describe en el boceto, cuando se vaya haciendo más reducido en los dispositivos, tendrá la propiedad de amoldarse a la pantalla. Esto sucederá en los dispositivos más reducidos como son los móviles, se diseñará para que se reduzca sin perder funcionalidad, en todos los dispositivos móviles tendrán las mismas herramientas. Estas es una de las funcionalidades más importantes que se quería desarrollar desde el principio del proyecto:



### 3 Web home, primera fase



### 4 Web Información, primera fase

Como puede apreciarse en las imágenes, se han creado todo lo incluido en el boceto que se aprobó, tiene el menú y el *footer* para que estén siempre visibles, así los usuarios no tendrían que subir la página para moverse entre los distintos apartados. Como se puede observar en el menú de arriba, está montado como se describía en el diseño y el boceto, El *footer* como se aprecia, tiene los botones para acceder a los demás apartados, menos en el apartado que está en ese momento mostrándose.

Como se aprecia en el apartado de información, está apropiado para una pantalla de ordenador, si esta pantalla se hiciera más pequeña, el video se pondría debajo de la información. La información se amoldaría al espacio que tiene para no perder el estilo.

Se han incorporado redes sociales y el logo en el lugar que están en el boceto, para respetar también esta parte en la creación de la página web.

## 3.2 Segunda Fase

### 3.2.1 Organización de apartados

Una vez que se tenía desarrollado las páginas web para su presentación, aun así no terminaba de gustar el estilo que había quedado, Así que se volvió reconstruir de nuevo. Se empezó a diseñar otra vez el boceto desde el principio, al empezar de nuevo se volvieron a pensar otra vez en la manipulación del servicio web. Para un uso fácil y que pudiese interactuar con cualquier dispositivo. Al empezar todo de nuevo, se añadieron nuevas utilidades y herramientas, además en el desarrollo mejoraron las herramientas, cada apartado y la utilidad que tendrá en cada una ellas.

Se decidió no incluir el apartado de registrar, pues era solamente para información del usuario en el servicio, cuando se empezó el desarrollo fue siempre pensando que se pudiera utilizar sin registro, desde el principio era llegar al mayor número de usuarios y dispositivos que fuesen posibles. Como es cada vez más común no poder utilizar las herramientas de los servicios sin

registro. Los usuarios crean más cuentas de distintos sitios servicios, que al final la mayoría no vuelven a utilizar.

Así que se rechazó el apartado de registro, ya que para poder comunicarse con un administrador estaba el apartado de contacto. Para conocer a otros usuarios interesados de saber cómo les fue a otros usuarios que ya utilizaron el sistema, están los enlaces a redes sociales incluidas en el servicio web, donde pueden mantener una conversación e interactuar en cualquier momento.

En el apartado de despliegue de sensores, se han añadido algunas herramientas para poder ubicar el mapa en la posición que se requiere, además de la creación y manipulación de áreas. Se pueden cambiar de color con un *click*, especificando que área es más peligrosa y cuál es menos peligrosa, para la colocación de sensores. Además se le ha añadido otra herramienta, una vez seleccionada el área, se pueda ampliar y hacer más pequeña si el usuario lo necesita, seleccionando el área deseada con más detalle. En un recuadro se ha incorporado una herramienta nueva, donde se muestra cuantos m<sup>2</sup> tiene el área, la cual se ha indicado a través de las herramientas de creación de polígonos. Esto es solamente para mostrarlo al usuario, la cual quiere ser mostrada para una mayor cantidad de información.

Se ha añadido otra herramienta muy útil, donde el usuario puede buscar la zona donde se encuentra el área donde desea hacer el despliegue, la búsqueda puede ser posible por el buscador de población o situaciones que tiene Google con esta herramienta. Posiciona al mapa, dirigiéndolo al lugar donde se ha decidido, escribiendo la donde se desea que se posicione el mapa, pudiendo mostrar el área de la población o situación. Además tiene una forma de búsqueda, que al pulsar una tecla automáticamente van saliendo las poblaciones o situaciones a las que corresponden estas letras que se pulsa.

En la forma de guardar y de cargar había un problema, ya que los mapas pueden utilizarse en cualquier dispositivo, se diseñó para que los dispositivos que no sean ordenadores, no tengan la opción de descargar o cargar los mapas KML. Cada dispositivo tiene una forma diferente de abrir el explorador de archivos y elegir el archivo deseado, con lo cual no se le dará la opción.

Hay dispositivos donde no se podrá cargar los mapas, porque esto podría complicar el manejo al usuario que necesite saber en qué posiciones se colocarían los sensores, entonces tendrán que cargarlos desde en un ordenador. Para un completo despliegue, se necesitaría un dispositivo al cual se le proporcionase las posiciones latitud y longitud, con este se podría saber exactamente donde colocar cada sensor en la situación que el sistema de despliegue ha proporcionado.

Estas situaciones se envían al correo del usuario, cuando el algoritmo termine de calcular las posiciones donde va cada sensor, porque depende del área o áreas que tenga que calcular este algoritmo. Puede tardar más de 24H, así que se tomó la decisión de mandar directamente un correo al usuario, con las posiciones donde se colocarían los sensores, además del mapa de despliegue para poder cargar en el servicio web.

Al diseñar desde el principio el servicio web, y por los cambios que se están haciendo, se volvió a empezar a buscar un nuevo diseño, pero esta vez se pensó en ver diseños de web que eran parecidas a redes sociales. Como es (Pinterest), en esta red social es donde se encontró el tipo del servicio web, que más gustó para el diseño. Ésta fue la orientación que hizo que el diseño tenga la forma que tiene, además se le hicieron algunos retoques para que fuese un diseño único y fácil de manejar.

### **3.2.2 Diseño de web**

Para el diseño de este servicio web, se ha utilizado otras fuentes para la recopilación de información, ya que la pantalla no se amoldaba a los apartados del servicio en los dispositivos. Tampoco tenía un diseño apropiado, así que se trabajó en otro diseño, en remodelar los apartados y las herramientas además de en la información que se deseaba mostrar. También se quería reutilizar el máximo código que se había realizado hasta el momento, para no desperdiciar todo el tiempo que se había invertido en la anterior versión.

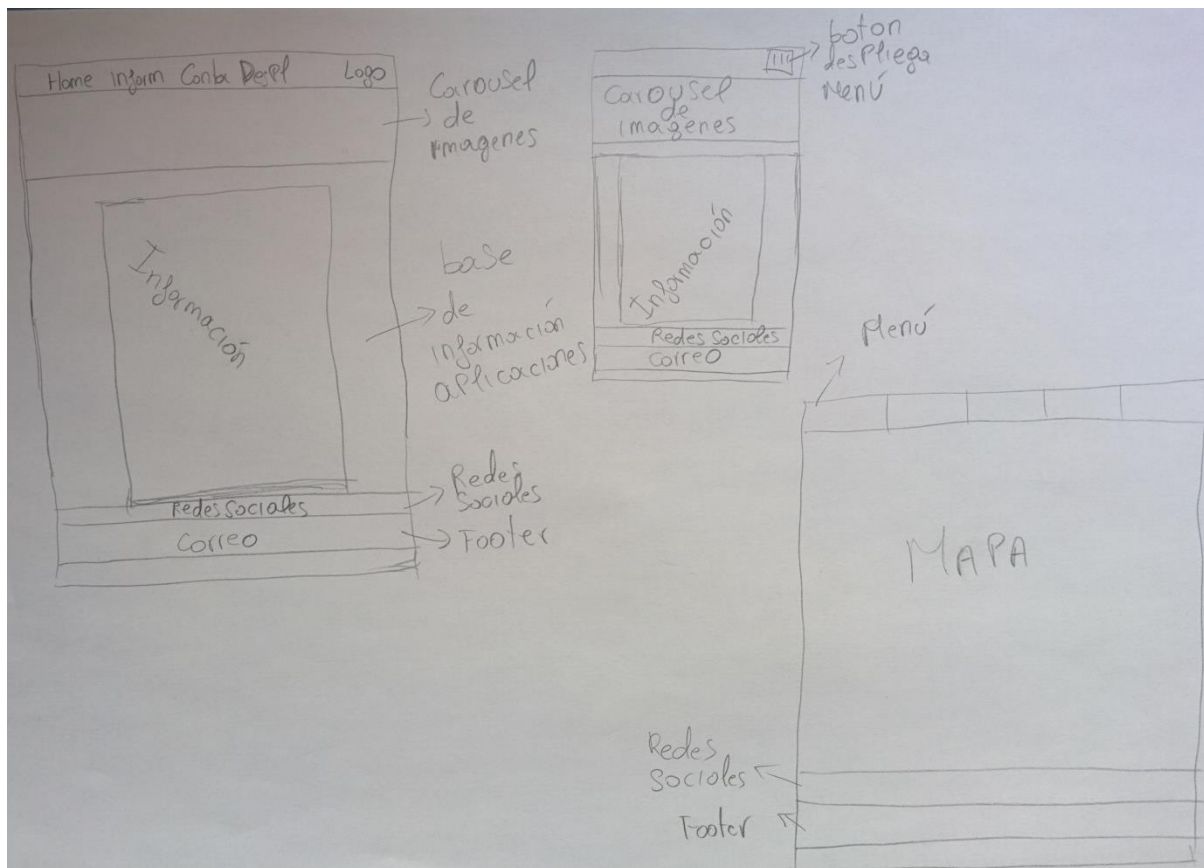
Se empezó por el apartado de Menú, como se ve en el boceto, las pestañas son más estrechas, y se remodeló la parte más alta, ahora no tiene las redes sociales en esa parte y tampoco el botón de registrar. Se trabajó en la distribución del menú, al igual que los apartados son diferentes, en la parte izquierda están todos los apartados con un tipo de letra. En cambio en la parte derecha, que es el nombre del servicio web, *Smart Cultivation*, es el elegido para ponerlo como logotipo, que está en otra fuente. Como también se puede observar en el diseño, el menú se contrae para dejar un solo botón, donde pulsándolo se despliegan los distintos apartados, que son las direcciones a los otros servicios.

En la parte central, es donde se integra la información de los distintos apartados, menos en el apartado de home, porque en este apartado, se ha incluido en la parte superior un carrusel de imágenes para que quede con un diseño más atractivo. Estas imágenes incorporan un texto explicativo, que a los usuarios les puede ser interesantes para información del despliegue de sensores.

En los demás apartados la información está centrada en un rectángulo, para que en todos los dispositivos se pueda ver con el mismo diseño, esto ocurre en todos los apartados menos en el mapa. Este no ocupa totalmente la pantalla por estética, pero es poco el espacio que no utiliza a lo ancho del dispositivo, se realizó para una mejor visibilidad además del manejo del mapa y sus herramientas.

Para la parte del *footer*, se ha optado por mantenerlo en la parte baja del diseño, pero con una distancia que sea como mínimo la más baja del servicio web. Esto es estático, significa que en todos los dispositivos se verá de la misma manera, además por encima del *footer*, también es estático, se encuentran las imágenes donde que en ellas, redirigen a las distintas redes sociales.

Como también se puede observar en la imagen, se amolda todo para que en los dispositivos no haya ningún problema de ver la información o de utilizar cualquier apartado, además de herramientas que están implementadas en el servicio.

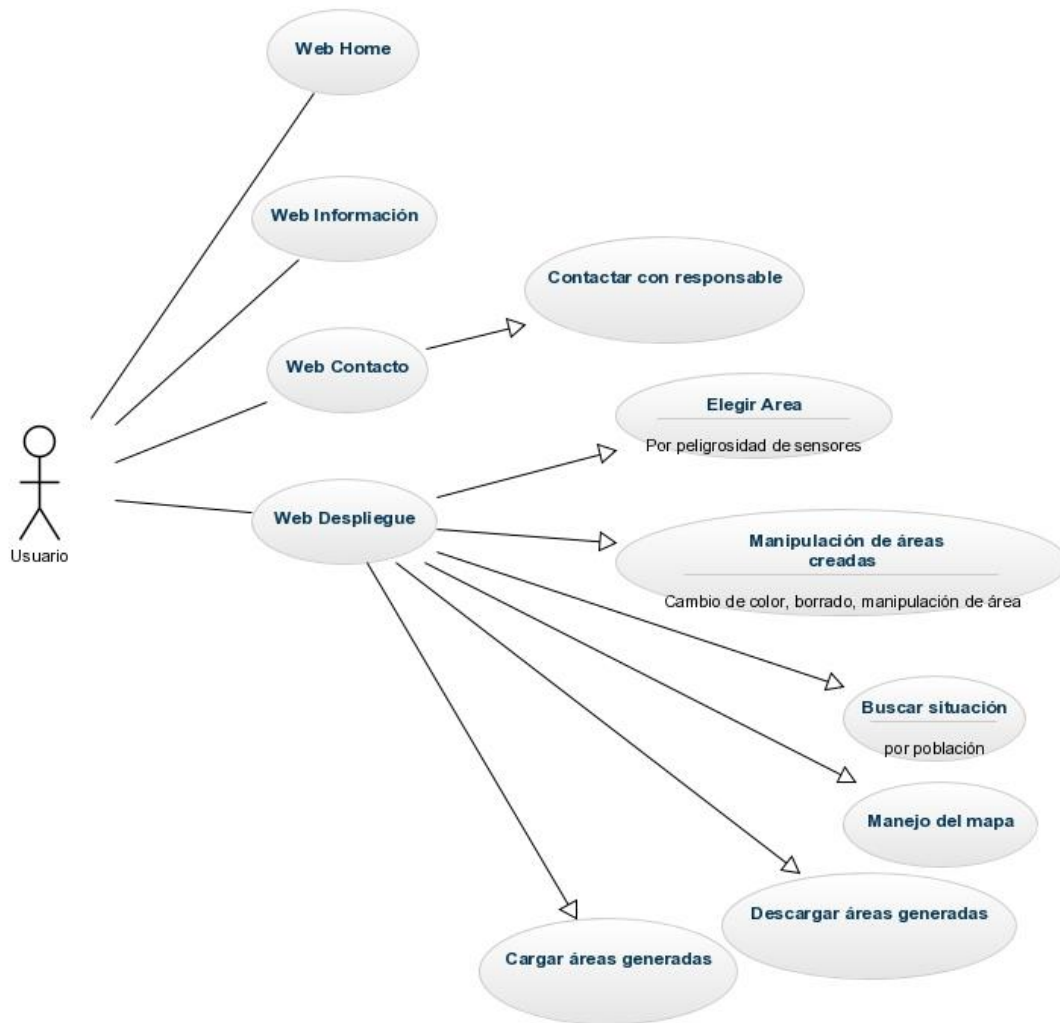


#### 5 Diseño web, segunda fase

### 3.2.3 Diseño casos de uso

En este esquema se desea mostrar las distintas operaciones que se pueden realizar en este servicio web, como se puede observar tiene los cuatro apartados, en contacto y despliegue. Además tienen distintas herramientas que se pueden utilizar.

En contacto la herramienta que se le proporciona al usuario, es poder mandar un correo al responsable desde el servicio web, en cambio, en el mapa se le han añadido más herramientas. El usuario puede operar con más propiedades, y así poder manipular el área con más detalle, para poder realizar el despliegue de sensores.



### 6 Casos de uso, segunda fase

#### 3.2.4 Servicio Web

Para la obtención de la posición de cada sensor, se ha diseñado un servidor que recibe peticiones para realizar este cálculo, esto se ha podido hacer gracias al servicio creado de optimización. Este servidor está activo esperando peticiones del servidor de servicio web, que es el lugar donde se crean las áreas. El servicio web es una de las partes más importantes de este proyecto, ya que es donde se ha utilizado la mayoría de los conocimientos del grado en ingeniería informática. Este servidor está creado para la ejecución en lenguaje Java, está montado en Apache Tomcat/7.0.28, pero la comunicación hasta este servidor es a través de HTTP, la cual se ha utilizado por protocolo POST.

Para el despliegue de este servicio, se han utilizado varias herramientas para la confección de este sistema, además de para la comunicación de un servidor



con otro, y así poder generar el despliegue robusto de sensores de un área determinada. El servidor de optimización está creado para la recepción de variables con *RestFul web services*, ya que este servicio está basado en métodos *HTTP* y tiene el concepto de *REST*. El sistema tiene definido los servicios URI, que para lo que se necesita es lo idóneo, además se pueden enviar *MIME-types*, que son *XML*, *JSON*, etc., y soporta las operaciones *GET*, *POST*, *PUT*, *DELETE*, en nuestro caso solamente hemos necesitado y usado el método *POST*.

Para ello se ha utilizado *JAX-RS* con *Jersey*, ya que soporta la vía *REST* el cual está basado en Java. Al tener que utilizar los algoritmos de optimización en Java para el cálculo del despliegue de sensores, se ha elegido éste, porque tiene una herramienta implementada llamada *JMetal*. La herramienta o librería es la que proporciona las funciones para despliegue robusto, además tiene las clases necesarias para generar la matriz, y se ha utilizado para hacer despliegues robustos.

### 3.2.4.1 Conexiones usadas entre servidores

Para la transmisión de datos, se han utilizado las siguientes partes del código del proyecto:

```
jQuery.ajax({
    type: "POST",
    url: '../Controller/comprobarDespliegue.php',
    data: {email : email,
          tamanos : tamanos,
          timeout: 1000,
          verdes : shapesTotales.length,
          coverteduras : sensores,
          mapaKml : kml,
          parametros : parametros
        },
    success:function(data) {
        //alert(data);
        $("#successRadar").empty();
        var texto = "";
        switch (data) {
            case 'success':

                texto = '<p><h4>Se ha realizado correctamente</h3><br>Recibiré un correo con el despliegue</h4></p>';
                break;
            case 'email':
                texto='<p><h4>El correo electronico no es correcto</h4></p>';
                break;
            case '0':
                texto='<p><h4>No se ha podido conectar con el servidor</h4></p>';
                break;
            default:
                texto='<p><h4>Ha habido un error.</h4></p>';
        }
        $("#successRadar").empty();
        $("#successRadar").append(texto);
    }
});
```

Con este pequeño trozo de código, se envían los parámetros a un archivo en *PHP* para poder ser validados y posteriormente enviados al servidor de

despliegue, además se le envían el email proporcionado, los tamaños de las áreas, la cantidad de sensores máximos, el mapa en KML y los parámetros por los que se registrará el algoritmo.

Servidor web:

```
function post_curl($kmlparse) {  
  
    //set POST variables  
    // $url = 'http://158.49.96.142:8080/SmarCultivation/rest/sensors';  
    $url = 'localhost:8080/SmarCultivation/rest/sensors';  
    //open connection  
    $ch = curl_init();  
  
    $fields = array(  
        'email' => $_POST['email'],  
        'tamanos' => $_POST['tamanos'],  
        'mapaDatos' => $kmlparse,  
        'coverturas' => $_POST['coverturas'],  
        'parametros' => $_POST['parametros']  
    );  
    $fields_string = '&';  
    foreach($fields as $key=>$value)  
        { $fields_string .= $key.'='.$value.'&'; }  
  
    $ch = curl_init();  
    //set the url, number of POST vars, POST data  
    curl_setopt($ch,CURLOPT_URL, $url);  
    curl_setopt($ch,CURLOPT_POST, count($fields));  
    curl_setopt($ch,CURLOPT_POSTFIELDS, $fields_string);  
    //execute post  
    $result = curl_exec($ch);  
  
    $result = curl_getinfo($ch, CURLINFO_HTTP_CODE);  
    //close connection  
    curl_close($ch);  
    return $result;  
}
```

En este trozo de código se puede observar que es en *PHP*, ya se han realizado las validaciones oportunas de los parámetros más sensibles. Con esta función que se utiliza para realizar la llamada, está dirigida al servidor de despliegue de la forma que le llegue a través del protocolo *REST*, con cada uno de los parámetros que se le envió desde el *JavaScript*.

Servidor de despliegue:

```

public Throwable run(String xmlShapes) {
    // xmlShapes="&email=ejemplo@ejemplo.es&tamanos=354.6853669853721!29
    // Recogemos los datos necesarios para la conversión y recogida de d:

    String[] variables = xmlShapes.split("&");
    String[] parametros = variables[5].split("/");
    /**
     * Se comprueba si el despliegue es de forma avanzada o por default
     */
    if (parametros.length==1){
        String[] par2 = new String[7];
        par2[1] = "40";
        par2[2] = "40";
        par2[3] = "30";
        par2[4] = "0";
        par2[5] = "0";
        par2[6] = "250000";
        parametros = par2;
    }

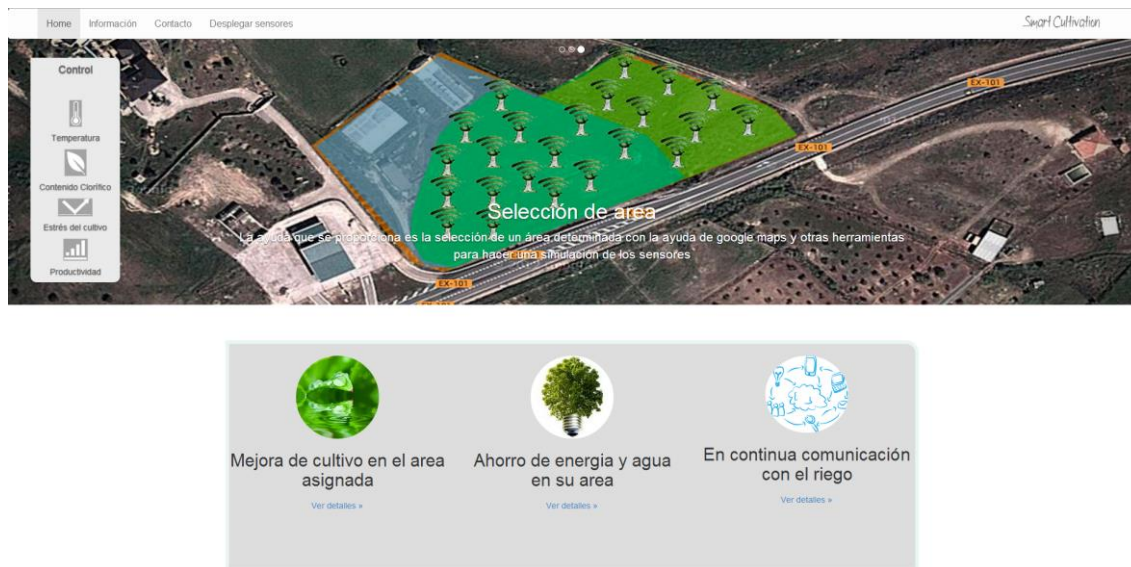
    String[] emailCliente = variables[1].split("=");
    String[] tamanos = variables[2].split("=");
    String[] coberturaSensores = variables[4].split("=");
    String[] cantidadSensores = coberturaSensores[1].split("!");
    int[] coberturas = new int[cantidadSensores.length];
    int cuentaCover=0;
    for (String cobertura : cantidadSensores) {
        coberturas[cuentaCover] = Integer.parseInt(cobertura);
        cuentaCover++;
    }
};

```

En el servidor de despliegue se recibe en un String con los parámetros, seguidamente se tienen que requerir *Split* para separar cada parámetro, y así enviarlos al algoritmo para que haga el cálculo del despliegue. Posteriormente cuando termine de generar el array con la posición de los sensores, se hace una consulta a otra función con el correo del usuario y se le envía los archivos que se describen en el proyecto.

## 3.2.5 Imágenes de del servicio web

### 3.2.5.1 Ordenador:



#### 7 Web Home, ordenador

Como puede observarse en esta imagen, es el apartado Home, está todo como se describió en el apartado de diseño. En la parte de arriba está el menú, que se desliza hacia abajo con el *scroll* del ratón, para tener siempre disponible los distintos apartados del servicio web. Debajo está la clase carrusel que va mostrando imágenes e información sobre ellas, además tiene incorporado botones a la derecha e izquierda para poder pasar de imagen cuando se requiera. En la parte alta de la imagen hay tres botones que son el redireccionamiento de las imágenes que hay, que hace referencia a la información. Un poco más abajo del carrusel, se muestra la información bien formateada, se puede apreciar que tienen unos botones debajo de cada imagen. Al pulsar se desliza hasta donde está la información de este botón, lo que no se puede observar es el *footer*, ya que solo se ve si el *scroll* está lo suficiente abajo, para llegar a ver esta parte.

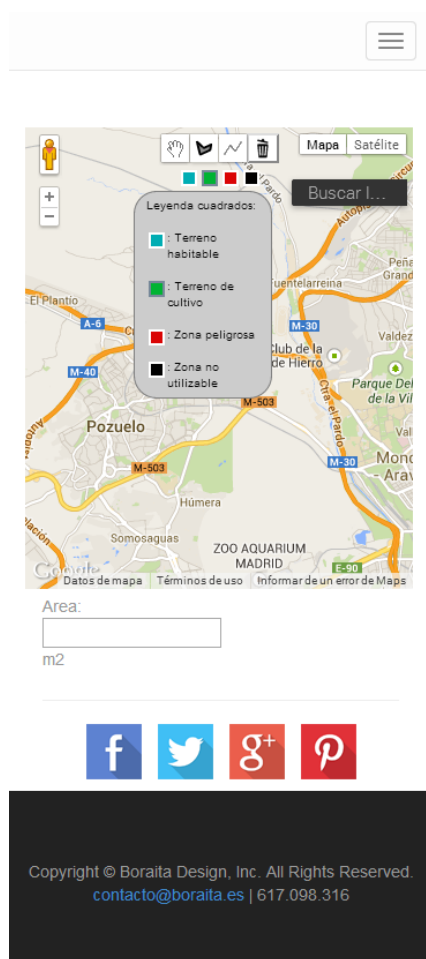
### 3.2.5.2 Tableta:



### 8 web información, tableta

Esta imagen es más reducida, ya que es para una tableta, en las tabletas se ve toda la información y todas las imágenes, pero de una forma diferente, al no tener tanta densidad de píxeles de ancho, se amolda a la pantalla. La información se quedaría un punto debajo de otro, así se puede mostrar todo. Como se ve, el menú se ha estrechado para que quepan todos los apartados, y en la parte del *footer* se queda igual, ya que la fuente de las letras se hace más pequeña para amoldarse a cada dispositivo, como también lo hace en todo el apartado, para mostrar la información correctamente.

### 3.2.5.3 Smartphone:



#### 9 web mapa, móvil

En los dispositivos más pequeños, como son los Smartphone, se amolda perfectamente la información relevante, lo único que no se muestra son imágenes no relevantes para la información de este servicio. Como se puede observar en el menú han desaparecido los distintos apartados, para dejar paso a un botón manejable para los Smartphone, pulsando sobre este se despliega los botones a los distintos apartados del servicio web. En el mapa están todas las herramientas que se pueden utilizar para crear las distintas áreas, además de poder manipularlas como se desee. Se puede observar aquí las imágenes de las distintas redes sociales, y el *footer* con la información debidamente colocada.

El apartado de desplegar mapa, en todos los dispositivos se pueden utilizar las herramientas dedicadas a la creación de áreas. En la parte superior están las

herramientas que pueden utilizarse para crear polígonos, líneas, mover el mapa, borrar un elemento que esté seleccionado y la paleta de colores. También podemos ver, al pasar con el ratón por encima de la paleta, o en su defecto con un dispositivo al tocar un color, saldrá una leyenda en la que se podrá leer que significa la diferencia de color en el polígono de un color u otro. Esta paleta supone una perfecta posición para el despliegue de sensores, ya que en el algoritmo es lo que se utiliza para determinar un despliegue robusto.

Para poder entender algo mejor como se ha realizado el diseño, se mostrarán algunos ejemplos de código.

Ejemplo de generación de *KML* con código *JavaScript*.

```
function GenerateKML(saveShapes){
    if (saveShapes.length!=0){
        //creamos la variable encargada de recoger el kml
        var kml = "";
        kml += '<?xml version="1.0" encoding="UTF-8"?>'+'\n';
        kml += '<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2">'+'\n';
        kml += ' <Placemark>'+'\n';
        for (var i = 0; i < saveShapes.length; i++) {
            var shapeType = saveShapes[i].type;
            var shapeColor;
            if (shapeType == google.maps.drawing.OverlayType.POLYLINE) {
                //Recogemos el color que tiene, comprobamos primero si es un polyline
                shapeColor = saveShapes[i].strokeColor;
            } else {
                shapeColor = saveShapes[i].fillColor;
            }
            var shapePaths = saveShapes[i].getPath().toArray();
            //Generamos los distintos poligonos y polylines y los guardamos en la variable kml
            kml += ' <Element>'+'\n';
            kml += ' <Type>'+shapeType+'</Type>'+'\n';
            kml += ' <Color>'+shapeColor+'</Color>'+'\n';
            kml += ' <Paths>'+shapePaths+'</Paths>'+'\n';
            kml += ' </Element>'+'\n';
        }
        kml += ' </Placemark>'+'\n';
        kml += '</kml>';
    }
    return kml;
}
```

Con este código se genera el *String*, para posteriormente poder ser descargado por el usuario y poder cargarlo, para posteriores operaciones o modificaciones que desee realizar sobre el mapa que generó.

```
@font-face
{
    font-family: 'fuente';
    src: url('../fonts/flower.ttf');
}
```

Código en CSS:

### Capítulo 3 – Análisis

Con este código se adquiere la fuente que se muestra, fue adquirida de una página web de fuentes de letras gratuitas y se adherido a este proyecto. Como se puede observar es algo novedoso y por este motivo se incorporó en el proyecto.





# Capítulo 4

---

## Diseño

Para comenzar con el diseño de este servicio web, se buscaron alternativas de lenguajes de programación, como se ha comentado en anteriores capítulos, los lenguajes elegidos entre tantos, para realizar el diseño de este servicio web son: *HTML5*, *CSS3*, *Javascript*.

Como se puede observar son los lenguajes que se están utilizando por la parte *front end*. Son los más indicados para la parte de usuario, gracias a la unión de estos, se puede crear un diseño bonito además de una funcionalidad muy buena.

Se pueden recurrir a herramientas y librerías que se utilizarán en el proyecto, algunas de las librerías más conocidas, que se van a utilizar son *Jquery*, *Bootstrap* y *AJAX*, además de otras librerías desarrolladas por desarrolladores, se crearon para ser utilizadas de forma gratuita, como son: ([anieto2k](#)), ([Twitter](#)), ([team](#)), además de otras. Con todo esto se ha podido realizar el diseño del servicio web que se quería para el proyecto.

### 4.1 Esqueleto

Para la creación del esqueleto de las páginas web a desarrollar, se desarrollarán en un entorno más sencillo para el usuario, con una carga sobre el dispositivo que sea la menor posible. Se tendrá en cuenta que las imágenes no sean de una cantidad, para que se pueda visualizar desde cualquier red de internet con rapidez. Se desarrollara para ser capaz de visualizarse en cualquier dispositivo, al menos los que se encuentran en el mercado en este momento, también se desarrollará de forma que se amolde a la pantalla del dispositivo.

Se han creado dos diseños como se comentó en el capítulo tres. Para el primer diseño, se crearon las páginas webs sin imágenes para una carga más rápida, utilizando la herramienta SVG.

Es una librería que compagina con HTML 5, en esta herramienta se tienen que programar los movimientos que realice la imagen o aun estando la imagen estática, al igual que cualquier funcionamiento adicional. Con esta herramienta hace que necesite menos ancho de banda para cargar el dibujo, pero en cambio al dispositivo se le da una mayor carga de procesamiento.

Se diseñó una cabecera (*<head>*) sencilla, con los links necesarios para poder utilizar el CSS de *Bootstrap* y el archivo CSS que se ha creado para la implementación del esqueleto.

Para el cuerpo (*<body>*) se le añadieron las clases necesarias de *Bootstrap*, así se hizo uso de esta herramienta en el menú. En la primera fase se añadieron clases como *collapse*, para que se amolde si la pantalla del dispositivo lo requiere. El menú se convierte en un botón, para que se pueda desplegar y así tener más espacio en el dispositivo, con objetivo de poder mostrar la información que se desea presentar. Para la utilización de esta clase, hay que organizar la lista del menú con una etiqueta *<ul>*, la herramienta opera con esta etiqueta amoldándose al dispositivo, administra los botones del menú en orden horizontal. Se añadió la clase *navbar-right* para que el botón de registro y las redes sociales estén en la parte derecha, esto hace que cuando el dispositivo tenga una pantalla algo más pequeña, se van acercando los botones, por el contrario se separan hasta completar la pantalla horizontal.

En la parte central es donde tiene que incorporarse toda la información, para mostrar cada uno de los apartados. Se agregó una etiqueta *<article>*, se le van incorporando otras etiquetas que se van requiriendo para construir el esqueleto, y así mostrar la información que se desea en el servicio web.

Para la segunda fase se decidió incorporar un área de información que ocupara el máximo de pixeles, se programó en CSS para que quedara de forma que los dispositivos de ese tamaño pudieran mostrar el contexto de forma ordenada.

Se decidió incorporar una línea verde claro de 2 *px* y un color gris muy claro para el fondo, por encima de esto va toda la información que se quiere presentar.

Para el *footer*, se incorporó en una etiqueta `<footer>`, se le añadió la clase `footer_menu` de *Bootstrap*, hace que sea el *footer* del servicio web. En esta parte también se tiene que poner una lista no ordenada, para mostrar los diferentes botones que se quieren mostrar, esto fue de la primera fase. En la segunda fase, se decidió no poner botones, solamente la información relevante del servicio web, como es el correo electrónico de contacto, quien la creó o un número de teléfono de contacto. Se añadió una clase más al *footer*, `navbar-static-button`, para que siempre estuviera en la parte más baja, para que una estética mejor, se modificó en el CSS para que tuviera un poco de margen entre el *footer* y la parte más baja.

El correo electrónico que se muestra en la página, se incorporó de forma que no pudiera saber los robot de búsqueda que correo es el que hay, así evitaremos que se nos puedan agregar a listas de envío de correos de spam.

Para terminar el esqueleto, se añadieron todos los links para los enlaces, así se podrá utilizar las distintas librerías que se requieren, para el uso de herramientas que se utilizan en el servicio web. Estos links son script que se crean para hacer las llamadas a los archivos con extensión *js*.

## 4.2 Home

En la primera fase Se creó una imagen principal que mostraba un despliegue de sensores, se hizo utilizando *Canvas*, es una herramienta que tiene incorporado *HTML 5*. Sirve para dibujar dinámicamente imágenes en una página web, esto se crea con lenguaje *Javascript*, el cual se puede crear en un archivo externo, para no llenar un archivo de código y sea así ilegible. Se creó un archivo llamado *Canvas*, éste tenía una función la cual se le proporcionaba una llamada desde la página donde tiene que mostrarse. Supone una carga mucho menor en la red, pero tiene una mayor carga para el procesamiento del

dispositivo al igual que SVG, por esto se prefirió recrear una imagen con *Canvas*.

Finalmente se decidió mejor en la segunda fase, incluir imágenes que mostraran mejor el beneficio que tiene el despliegue de sensores en un área agraria. Se pueden incorporar tantas imágenes como se crean necesarias, esto tiene una mayor carga en la red de internet. Pero los beneficios de mostrar estas imágenes son mucho mayor que un dibujo, así que se decidió incluir un carrusel.

Carrusel es una clase de *Bootstrap* que ya está implementada, para que haga unas funciones determinadas con las imágenes, además de ir amoldándolas, dependiendo del tamaño del dispositivo. Incluye tantos botones como imágenes haya incluidas, estos botones se han colocado en la parte alta de las imágenes. Las clases usadas son las siguientes, *carousel*, *slide* y *carousel-fade*. Tiene una función maneja el tiempo entre intervalos del paso de imágenes, esto se modificó y se creó una propia, porque el tiempo de intervalo no era el adecuado.

También decir que gracias a las clases usadas se incorporan dos botones transparentes, que al poner el ratón encima se oscurecen para pasar de imagen a la izquierda o la derecha. Tiene otra propiedad este *carousel*, cuando se muestra la imagen, también se muestra un texto donde explica el beneficio, que tiene el despliegue de sensores en un área de regadío, esto se crea con la clase *carousel-caption*.

Las clases que también se usan para los botones, que están colocadas en la parte de la imagen a la izquierda y derecha, son las clases, *carousel-indicators* y *carousel-caption*.

Se incluyó también en la segunda fase, un texto explicativo donde se muestran los beneficios de agregar los sensores en el área de regadío, además se modificó, ya que no se creía que estuviera de una buena manera explicado, ni era estéticamente más bonito. Lo que se hizo fue remodelar todo el *body* de la página home, se quería mostrar la información más importante a primera vista. Cuando se carga el servicio web en el dispositivo, como ya estaba el carrusel de fotografías, se incorporaron debajo tres imágenes con los

beneficios que este despliegue supone. Estas imágenes se incluyeron con una clase llamada *center-block*, las imágenes y el texto explicativo forman un bloque y se quedan centradas en la pantalla.

Para más información sobre los beneficios se ha aportado un botón debajo de cada imagen, tienen un ancla que a su vez tiene una función, por cada una de las imágenes.

La función que está creada en un archivo aparte es en *Jquery*, lo que realiza esta función es un desplazamiento uniforme vertical hasta donde está incorporada el ancla, que es en la información más detallada de la imagen que se desea ver. Para cargar las imágenes, se ha necesitado otra clase, *Img-circle*, esta clase transforma a la imagen que se muestra en redonda. Además se ha usado para información más detallada debajo de esta, pero para la imagen se ha creado más grande para que se viese mejor.

Se ha usado otra clase para que se coloque la imagen a la derecha o a la izquierda, y el texto de información en la parte contraria de la imagen. *Pull-right* o *pull-left*, son las clases usadas en esta parte, el texto se ha considerado poner el más importante en un tono más oscuro, otro no tan importante en un tono más claro, para una visión directa y que el usuario vea la información más relevante.

Esta es la presentación de del servicio web de HOME:



### 10 Presentación home

## 4.3 Información

La finalidad para la que se creó este proyecto era para información y dar servicio al usuario, pero no cuál iba a ser el diseño para el servicio web, se empezó incorporando un texto de información. Cuáles eran los beneficios de daba desplegar una red de sensores robusta en una zona agraria, de una forma explicativa y fácil de entender para cualquier usuario. Se redactaría unos pasos a seguir, para el buen manejo de las herramientas creadas y así poder hacer el despliegue de sensores. Se requirió añadir un video explicativo, donde quede claro cuál es el beneficio de ahorro de agua y el aumento de producción sobre la zona agraria. Esto se encerró en una etiqueta div con las

clases *container*, que es la clase la cual va a contener toda esta información. Dentro de esta hay una etiqueta *article*, las clases que incluye esta etiqueta son *container-fluid* y *center*. La función de estas clases son las siguientes, Mantener la información dentro de la etiqueta *article*, esto consigue mostrar en cualquier dispositivo la información deseada y así poder leerla el usuario. Dentro de *article* se incorporan cada una de las etiquetas, que son necesarias para poder mostrar el video y la información. Se incorpora un *div* por cada uno de los grupos de información para el video, se le pone la etiqueta *video*, que también es nueva etiqueta en HTML 5, Se compone del tamaño del video.

La clase que va a usar es *col-lg-12*, esto hace que se ocupen doce columnas que crea *Bootstrap*, con esta clase se va amoldando al dispositivo para que no haya problema de visualización.

El problema viene al ser un video, no se amolda perfectamente a la expansión de cada uno de los dispositivos, se ha tenido que crear unas reglas en CSS. Se incorporan una programación por cada tamaño de pantalla, las cuales se estipula que puedan ser las medidas más usuales que se vayan a utilizar, se han creado cuatro tipos de medidas, con esto se cubren todos los dispositivos.

Para los otros dos apartados que se quieren añadir, se crea por cada uno un *div*, donde se colocan *container-fluid* y *col-lg-6*. La primera clase hace que se amolde el texto al tipo de pantalla, la segunda clase crea 6 columnas *Bootstrap*. Estas columnas no se ven para el usuario, es para ordenar el tipo de información, el máximo de columnas que se pueden añadir son 12, como la clase que se ha añadido en la etiqueta *video*. Se ha escogido una clase con 6, pues para la otra parte de información se han elegido la misma, para que puedan verse cuando la pantalla lo permita. Cuando la programación de *Bootstrap* decida que el dispositivo es más pequeño, se reorganiza y se pondrá uno debajo de otro, para poder mostrar toda la información, de forma ordenada y fácilmente leíble.

En la primera fase no fue así, se le pusieron al video una de 6 y a los otros *div* de 4, para que se amoldase, se pretendía que el video se viera justo al lado de la información, cuando la pantalla lo permitiese. Hacía algo parecido a lo que ahora está haciendo, ponerse uno debajo de otro, pero antes el video estaba el



ultimo, como se ve en este momento el video está el primero, ya que es más relevante para la información del usuario.

Esta es la presentación del servicio web de INFORMACIÓN:

Home Información Contacto Desplegar sensores Smart Cultivation

### Video Demostrativo



1:29

#### Información

- Con esta web, se quiere poner a disposición del usuario, una herramienta que donde podría dibujar áreas con una herramienta de Google Map, llamada polígono, gracias a esta herramienta se pueden dibujar encima de un área determinada del dicho mapa, con esto además de saber cuánto m2 mide el área que ha seleccionado, se han agregado más herramientas para dibujar estos polígonos, se puede elegir un color en ellos para diferenciar, un área de peligro, un área de regadío o un área donde no se puede hacer nada de esto, además se pueden guardar estos polígonos, se genera un archivo el cual se puede cargar de nuevo en esta web para unos próximos arreglos de estos polígonos
- Con estas herramientas dichas en el punto anterior, se pueden hacer un despliegue de sensores, gracias a un algoritmo sofisticado, el cual

#### Pasos a seguir

1. Para hallar el área que se quiere seleccionar, este mapa tiene la opción de, recibir la ubicación del dispositivo desde el que se está trabajando, o bien por un buscador integrado en la parte arriba derecha del mapa, este proporciona la posibilidad de buscar una población o situación que tenga almacenado la base de datos de Google.
2. Una vez encontrado el área que se quiere seleccionar, se debe de seleccionar con la herramienta que esta situada en el centro arriba de la pantalla, cuando este seleccionado, se deberá ir marcando por clic, de ratón si es en el ordenador o por touch si es un dispositivo como tableta o Smartphone, hasta que se tiene todo el área deseado, si no se ha hecho con mucha perfección, no importa, ya que el mismo polígono se puede agrandar por partes, tocando uno de los puntos del medio y noniéndolo en el sitio deseado, y si aun así no se ha puesto

### 11 Presentación información

## 4.4 Contacto

En este apartado el diseño era lo importante, ya que el contenido era bastante lógico, así que se creó una etiqueta *article* donde se incorporaron las clases *container-fluid* y *center*. La primera clase es para que se amolde en cualquier dispositivo y mantenga toda la información a mostrar. La segunda clase es para que la información esté centrada, esto se ve en una pantalla de ordenador, ya que en los dispositivos más pequeños ocupan toda la pantalla.

Con el *name* del campo se recogen los datos de los campos y con la etiqueta *action* se envían los datos para procesarlos, seguidamente se enviarán a un

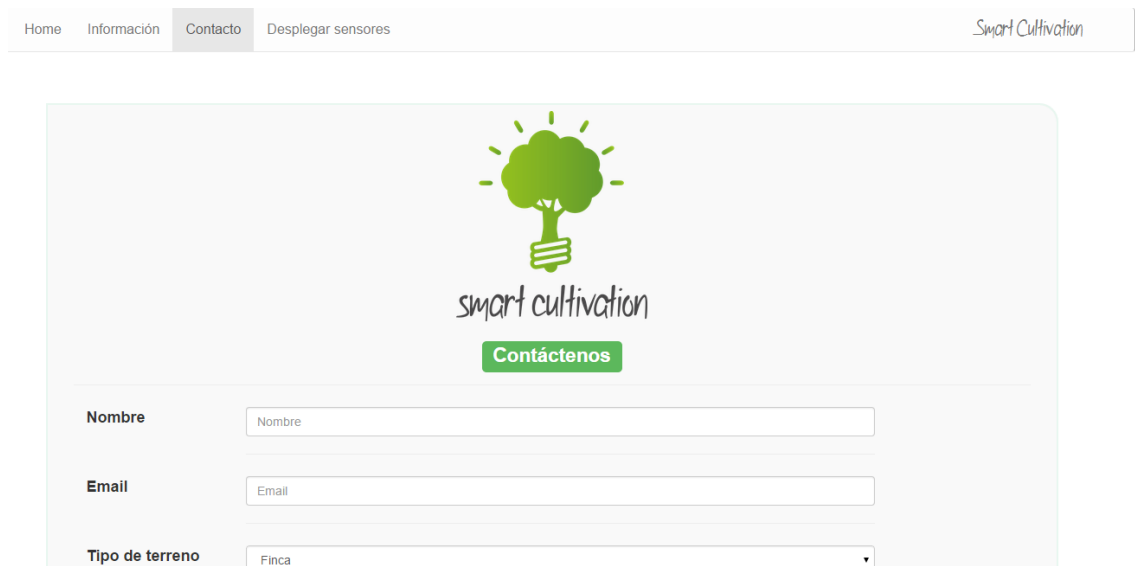
*PHP* para ser validados. Además cada *label* que contiene el formulario está compuesto por clases, para que no haya ningún problema a la hora de visualizarlos en ningún dispositivo. En cada input se colocan los datos para poder mandarlos, tiene cada uno su tipo para que los datos se recojan de distinta manera y ordenada.

Una ventaja que tiene HTML 5 es que, los dispositivos móviles muestran el teclado a razón del tipo de campo que se vaya a rellenar. Otra propiedad es que si no se escribe correctamente el dato del tipo, al dar al botón enviar muestra un mensaje en el propio input, los datos introducidos no son los correctos, se tiene que poner el dato correspondiente. También se ha incorporado un *option select*, la clase que se ha incluido es *form-control*, para que tenga control de cuando un elemento es seleccionado, y además de cambiar estéticamente al input.

También el botón tiene una clase incorporada para que sea estéticamente bonita llamada *btn-primary*, una vez rellenado todos los datos que están en el formulario y están validados cada uno de ellos, se pulsa el botón de enviar. Este formulario manda la llamada con el protocolo POST a un archivo en *PHP*, al recibir los parámetros lo siguiente es adquirirlos por el método *POST*. Una vez que se tienen los parámetros se vuelven a validar, es aconsejable también validarlos en el servidor, con esto se van comprobando cada uno de los datos que se han enviado y si están correctamente formados.

Se llama a la función enviar correo, este email le llega formateado con un estilo, para que al usuario pueda leerlo de forma correcta y saber que ha pasado con el despliegue que envió. Por el caso contrario hay alguno de los input mal puestos, se muestra un mensaje en la pantalla donde aparece que no ha podido ser mandado, que vuelva a intentar enviarlo.

Esta es la presentación del servicio web de CONTACTO:



### 12 Presentación contacto

## 4.5 Desplegar sensores

Esta es la parte más importante del proyecto, ya que para poder desplegar sensores, es necesario primero crear el área donde se va a realizar el despliegue. Para esto se han utilizado la herramienta de los mapas de Google, ya que es una gratuita y además de tener mucha información, es el que más herramientas y desarrolladores trabajan con esta. Esto da una gran ventaja por buscar problemas que surgen durante el desarrollo con la herramienta del mapa de Google.

Este apartado ha sufrido grandes cambios durante el diseño por la colocación del mapa y sus diferentes herramientas, además de incluir nuevas opciones que no vienen definidas en el mapa de forma estándar. Vamos a empezar por el diseño exterior, después seguiremos describiendo cada una de las herramientas que se han proporcionado en el mapa de Google.

Se ha colocado como en los demás *article* con la clase *container-fluid*, para que pueda amoldarse al dispositivo, para ello se crean contenedores que son *div* con un *id* alojado. Estos se crean antes en el servicio web donde mostrarán, para después ser implementados por el árbol DOM con la librería *Jquery*. Se ha creado un *div* para el mapa y otro para la paleta de colores.

Esta paleta se encarga de decidir qué color tiene el polígono que se está creando, además se ha creado un botón para borrar los dichos elementos teniendo que estar creados en el mapa. Para dar más información al usuario se ha creado un contenedor donde tiene alojado un input, en este se muestran los m2 del área que esté seleccionado, esto engloba al que se acaba de crear.

También se han incorporado unos contenedores que aparecerán u ocultarán cuando se necesiten, para la manipulación de los archivos que se generan o que se desean subir al servidor. Una vez creados las áreas en el mapa, se pueden descargar.

Esto es lo necesitado para incorporar las herramientas, ya que con *Jquery* se van insertando elementos en el árbol DOM donde se verá a continuación.

### 4.5.1 Mapa

Para la carga del mapa en contenedor asignado, se ha creado un div asociado a un id, este id es el que se usará para poder identificarlo en el árbol DOM. Para la carga en la etiqueta *body*, se ha puesto un comando, haciendo una llamada a una función la cual hace que cuando carga el *body* sea el primero en cargar. Se ha tenido que hacer un link al archivo que contiene dicha función, que es la encargada de llamar para la incorporación del mapa, a esta función se le ha llamado mapa.

La función contiene todo lo esencial para cargar además del mapa, todas las distintas funciones que se van realizando mientras carga las demás herramientas. Lo primero que hace en esta función es crear un array de marcadores, después se crea la variable donde va incorporado el mapa, además integra en el id a través de *Jquery* el contenido. Dicho mapa y las opciones del mapa que va a tener esta función: `new google.maps.Map(document.getElementById('map-canvas'),mapOptions);` para que el mapa tenga una situación, una vez que se cargue en el contenedor se le añaden dos posiciones, ya que si se le añade solamente una, el mapa al cargarse obtiene el máximo zoom posible. Así al tener que mostrar las dos posiciones se queda a una distancia el mapa que se creyó recomendable.

Antes de incorporarlas al mapa se añade a una variable especial la cual recoge la longitud y latitud. Después de esto, La función sigue y le pide al navegador la posición del dispositivo que está accediendo a este mapa, aquí el usuario puede elegir el aceptarlo o denegararlo. Al aceptarlo, se pondría en el mapa esta situación, pero si se deniega, el mapa no se mostraría, al no tener ninguna ubicación. Este fue el motivo de haber puesto antes una posición por defecto. A continuación se cargan algunas herramientas.

### **4.5.1.1 Búsqueda de situaciones**

La primera herramienta que se carga, que además se coloca en la parte superior derecha del mapa, es la herramienta de búsqueda de situaciones o poblaciones en el mundo. A esta herramienta se le incluye otro comando que necesita de Google para que pueda buscar en su base de datos, además a esta herramienta se le añade un evento. Este evento consiste en, que cada vez que se pulse una tecla cuando tiene el foco sobre él, hace una consulta a Google con las situaciones que coinciden, con las letras que están en esta herramienta.

Seguidamente, se integra en el árbol DOM la paleta de colores y el botón de borrar, esto se guarda en una variable que se recoge a la hora de utilizarse. El siguiente comando, es colocar la imagen que se ha incorporado al encontrar la situación que se proporciona al mapa, aquí se define qué tipo es y el tamaño que se le quiere proporcionar. Una vez se utilice la herramienta para localizar una población o lugar, se llama a una función donde obtiene la situación que devuelve Google, se añade el título que tiene la imagen que se ha descrito antes, además del tamaño del mapa y el mapa donde se carga. Todo esto se incorpora en la “marca” la cual es la función que se desarrolló, se encarga de colocarla en el mapa y dirigir la situación de este hasta la “marca”.

### **4.5.1.2 Herramienta de polígonos**

Una vez terminada con esta parte de la función, pasa a cargar las herramientas necesarias para la manipulación de polígonos, se empieza implementando las opciones que van a tener. Después se crea la variable que adquiere todo lo relacionado con dibujar polígonos, líneas o mover el mapa, se añade la posición que va a estar colocada la herramienta. Se añaden las

opciones y en el mapa que va a cargarse, es la variable que se creó al principio del servicio web. Una vez terminado con la implementación de la herramienta, se crea un evento *listener*, la cual espera las llamadas que se hacen, se reduce la escucha cuando uno de los polígonos o líneas se ha colocado o cerrado en una parte del mapa.

En esta función recoge el polígono o la línea correspondiente que se guarda en una variable, estos datos pasan por una serie de sentencias condicionales, para ubicar la variable en el sitio que le corresponde. Una vez que ha pasado de este proceso, se sabe que es un “*Shape*” válido, entonces se guarda en un array de “*Shapes*”, seguidamente se hace la condición de si es un polígono o una línea. Una vez las sentencias lo llevan a ser un polígono comprueban si tiene menos de tres puntos, si no es así se borra directamente la variable, del array de “*shapes*” y del mapa. Si en vez de ser un polígono, es una línea, y tiene menos de dos, ocurre lo mismo de antes, se borra la variable, del array y del mapa. Por el contrario pasan las sentencias y además cumple que tiene los puntos suficientes, entonces se llama a una función con la variable llamada “*Shape*”.

En esta función se le manda la variable del nuevo “*Shape*”. Primero se llama a una función donde si hay un “*Shape*” seleccionado antes del que se ha creado, se le quita la propiedad editable y lo deselectiona. Después de esto, pone al nuevo “*Shape*” como seleccionado y tendrá la propiedad editable activa, más tarde recoge el color que tiene para añadirsele como color principal al polígono o línea.

### **4.5.1.3 Cálculo de área**

Se creó una función donde tiene que pasar por la condición en el cual tiene que ser un polígono, se manda el *Shape* para el cálculo de área en m<sup>2</sup>, entonces se hace uso de un comando que se encontró en (Developers). Se hace una llamada a Google al devolver el valor, al dar un id al input se insertan el valor devuelto por el árbol DOM y gracias a las funciones *Jquery*.

### **4.5.1.4 Borrar Shape del mapa**

Ahora lo que hace la función es crear eventos a los distintos botones para que tengan una función que hacer, se crea el botón pero en la función es donde se

le da la utilidad, tiene que hacer el click en el para que se accione dicha función.

Uno de los botones es la herramienta de polígonos, otra es deseleccionar el polígono que este seleccionado al tocar en el mapa, y otro es borrar un polígono del mapa, la cual llama a una función para borrar. Manda el array de Shapes y comprueba si hay un Shape seleccionado, se le borra el dato al input que muestra los m<sup>2</sup> del polígono que está. Se busca cual es el Shape seleccionado y se guardada en una variable, seguidamente se borra del array de Shapes y también del mapa.

### **4.5.1.5 Paleta de colores**

La función hace una llamada para crear la paleta de colores, que son los utilizados para la elección cuan es la facilidad del polígono en caerse, y los cuales se le proporcionan a los “Shapes”. Primero el div generado con el id para la paleta de colores se le agrega a una variable, para poder añadirle la funcionalidad. Hay una variable la cual contiene los colores, se quieren integrar en la paleta. Se añade el código del color en otra variable, que es la que se encarga de agregar el color en la paleta, así va ocurriendo con cada uno, hasta que ya no quedan más colores que agregar. Seguidamente se selecciona el color que se haya determinado por defecto en la paleta de colores.

Hubo que añadir una leyenda a esta paleta de colores, ya que cada color significa una peligrosidad para el cálculo de los sensores. Se crea con el árbol DOM un nuevo div y se le incluye una función, encargada de mostrar la leyenda de cada color que está en el div, además se le añaden más funcionalidades. Tiene un código en CSS añadido para que salga en una posición de la web, para que no moleste al usuario al trabajar mientras lee la leyenda. Además se le añade la funcionalidad para que muestre este div solamente cuando esté el ratón encima de la paleta de colores, al retirar el ratón o tocar en otro lado del mapa con el *touch* hacemos que el div se oculte.

### **4.5.2 Herramienta guardar mapa**

Cuando la otra función ha terminado, se carga la herramienta generar mapa, que crea un evento donde espera un clic de botón, el cual se creó en el servicio

web. Cuando se pulsa el botón se lanza una función donde se le envía el array de “Shapes”, la primera sentencia que se encuentra es si hay al menos un Shape, al ser así se genera una variable con una cadena. Esta cadena es lo que se mandaría para crear el archivo, se crea la cabecera del archivo, se incorporan cada Shape que haya, se recoge en una variable para después incorporar al String.

En este archivo los parámetros que se guardan son, qué tipo de “*Shape*” es, que color es el que tiene y el array de puntos que tiene el polígono o línea. Todo esto se añade por cada uno que haya en la variable tipo String, y finalmente se incorpora el pie del archivo. Una vez creado el String, con todos los componentes necesarios, se hace uso de *Jquery* con *AJAX*. Ya que no es aconsejable hacer las operaciones de crear archivos en el cliente, para generar un archivo en el servidor se ha utilizado el lenguaje *PHP*, pues se manda por un conector que es el encargado de utilizar *AJAX*.

Se llama a una función que su propósito es generar el archivo con String que se está enviando. Esta función se encarga de desanclar el archivo llamado mapa.kml antes de incorporar el nuevo archivo, posteriormente los datos los incorpora en un archivo con el mismo nombre en String. Una vez que se ha creado el archivo, con *AJAX* se hace una llamada a la dirección donde se encuentra este archivo, esto hace dar al usuario la opción de guardarlo en el ordenador. Con el archivo en posesión del usuario puede ser cargado en el servicio web, pero Si el array de Shapes está vacío, simplemente no hace nada.

### **4.5.3 Cargar Mapa**

Para la carga de los polígonos que se hayan realizado en otro instante con el servicio que proporciona este proyecto. Primero se ha crea en el menú un botón donde tiene un id asociado para poder acceder desde el árbol DOM, con una función que pueda asociarse al botón. Para que la carga sea de una forma más novedosa, se le ha añadido una librería que ha sido desarrollada para ser utilizada por terceras personas. Esta librería de autor aquí nombrado (Klinggaard), está creada en *Jquery*, esta función es creada en el servicio web de despliegue de mapas. Al cargarse el archivo al botón de carga de mapas, se le añade una función la cual cuando se pulsa añade a un elemento que se



haya creado en el servicio web, para el despliegue de sensores con un id asociado para ubicarlo en el árbol DOM más fácilmente.

Una vez insertado el elemento en la carga de esta función se puede enviar el archivo, cuando el servidor recibe el archivo lo procesa, una vez que el algoritmo termina de hacer las operaciones para incorporar los polígonos y si incluye los sensores. El sistema se posiciona encima del mapa desplegado, con esto se podrá observar el despliegue de una forma más real.

### **4.5.3.1 Popup**

La librería es la encargada de mostrar el elemento como un *Popup*, donde es un elemento que se muestra en la pantalla de una forma peculiar. Se ha programado para que salga deslizándose de izquierda a derecha, además de oscurecer la pantalla que está por la parte trasera de este elemento y así no poder utilizar la trasera. Esta librería tiene la propiedad que si se pulsa en otro sitio que no sea el elemento mostrado, este elemento desaparece deslizándose desde donde está, hacia la derecha. Tiene incorporado un botón, que está colocado arriba a la derecha del elemento para que haga esta misma operación.

### **4.5.3.2 Upload**

El elemento al que se ha proporcionado la propiedad de *Popup*, tiene alojados varios elementos para proporcionar otras herramientas. La función que tiene alojada de *Upload*, primero hace segura la transferencia dándole la propiedad a un elemento con un comando en *Jquery*. Después se inicializan variables, más tarde llega la función donde se revisa el archivo que se ha arrastrado hasta el elemento *Popup*, aquí se comprueba si el archivo tiene la extensión deseada, que en este caso es *KML*. Si el archivo no tiene esta extensión se mostrará un mensaje en el elemento donde se está mostrando, poniendo que tiene que ser un KML. Si lo es se muestran los botones para poder subir el archivo, además se muestra el botón de borrar e imágenes para que sea más atractivo estéticamente.

Si se pulsa el botón de borrar, se borrará el archivo y además se ocultan los botones y las imágenes que se mostraban, para solo quedarse el elemento de arrastrar el archivo. Si se pulsa el botón de cargar, este botón tiene una función a la que es llamada para la carga del archivo, además de mostrarse

una barra de carga para ver como se está subiendo el archivo. Se ha programado para que una vez que esté subido el archivo, automáticamente se carguen los polígonos en el mapa. Para esto Se envía el contenido del archivo a una función en *PHP*, para que las operaciones se hagan en el servidor y así se recogen las referencias del archivo.

Con una función en *PHP*, se inserta todo el contenido del archivo en una variable, para poder sacar los elementos de los distintos Shapes, este contenido se convierte en un String. Esto hace más fácil el manejar funciones en *PHP*, el motivo es retirar del archivo todo lo que no sea el color, el tipo y los puntos de los Shapes, esto se hace con cada uno de los Shapes que haya. Con toda esta información se devuelve a la función desde la que se ha hizo la llamada, además se mostrará un texto que muestra subido exitosamente, y se almacena en un array con los distintos apartados de los Shapes. Al recoger el mensaje, separa el array que ha devuelto la llamada para convertirlo con la función *Split*, para poder sacar solo la información necesaria. Se hace un bucle para recoger cada uno de los Shapes, el cual saca la información de ellos y generando un Shape con cada uno.

Seguidamente se incluyen en el array donde se encuentran todos los Shapes en el mapa, pero antes se comprueba si el Shape que se va a incorporar está ya en el mapa, si es así lo descarta y no añadirlo en el mapa. Si por el contrario no está, lo incluye la variable mapa que se creó al principio en el apartado mapa. Para insertarlo en el mapa de Google creado, además de que se pueda operar con cada uno de los Shapes que se están insertando, se le añade un evento igual que cuando se crea desde el mapa. El evento es para cuando el ratón o un *touch* hacen clic sobre este, así el Shape pueda manejarse con todas las herramientas. Una vez que están cargados e insertados todos los Shapes cargados del archivo, se añade al mapa la nueva ubicación donde tiene que dirigirse, que es donde los *Shapes* creó el usuario cuando los hizo, además se le añade al mapa una altura para poder visualizarlos debidamente.

### **4.5.4 Despliegue de sensores**

Una vez que se han generado las distintas áreas con sus respectivas peligrosidades, aquí aplicaría pulsar en el botón de carga de mapas, para

generar el despliegue y se pueda insertar el mapa que se ha creado. Cuando se pulsa, aparecerá una pantalla donde vienen los tamaños de las distintas áreas verdes creadas. Ya que son los áreas que se van a tomar como referencia, para incorporar los sensores dentro. También aparece la distancia entre sensores, que el usuario quiere proporcionar en estas áreas. Algo muy importante, es el correo electrónico donde, después de hacer el despliegue robusto de los sensores, se le enviará un KML.

Esto aparecería de forma por defecto, esta forma se ha categorizado como la normal, ya que las variables con las que el algoritmo va a desplegar están por defecto. Como se puede observar en esta pantalla, se puede encontrar un radio *button* donde se puede elegir el modo avanzado. Este modo deja que el usuario pueda desplegar los sensores de una manera más profesional, en la cual además de las normales se pueden elegir qué valor tiene cada opción, así es como trabajará el algoritmo.

Cuando se ha mandado el mapa para desplegar, posteriormente podrá cargar este en la aplicación, con cada una de las posiciones de los sensores generados.

Para el despliegue de sensores, se genera un archivo *XML*, en el cual se incorporan todos los polígonos generados en el orden de creación, para poder procesarlos de esa manera. Este archivo *KML* se procesa antes para ser enviado al servidor posteriormente, se comprueba que los datos que se van a enviar son correctos, esto se procesa en una función *PHP* para una mayor seguridad. Después de comprobar que todo está correcto, se hace una llamada con tecnología Ajax, donde se utiliza un servicio *Restful*, el cual trabaja con Jersey. Con la llamada que se hace desde Ajax se envían varias variables, pero se destaca que una de ellas es la cadena compuesta para el procesamiento, en esta cadena van los polígonos, sus características y las variables que el usuario eligió. Todo esto se envía al servidor donde se recoge a través de Post, y esto hace que se llame al servicio *RestFull*, en el servidor recoge todas estas variables y después el algoritmo calcula el despliegue con las diferentes clases.

Mientras se iba creando el proyecto, se fue cambiando algunos conceptos, para un mejor rendimiento y un mejor procesamiento. Se empezó a generar el cálculo de las áreas en la parte del cliente, ya que con las herramientas que se

tenían del mapa, se podían hacer algunos cálculos en la parte del cliente de los de las áreas, estos se hacían de forma más sencilla. Pero con los cálculos que había que procesar eran demasiado procesamiento para un dispositivo con esas características. Esto hacía que el procesamiento total del despliegue no fuese eficiente, ya que este procesamiento tendrían que hacerlo también en los dispositivos móviles. Por esto se determinó el enviar los datos al servidor, que allí se hiciesen todos los cálculos necesarios además del despliegue.

Una vez enviado al servidor, se utilizan funciones para crear las matrices necesarias, por cada una de las áreas verdes que el usuario ha generado y enviado. Se divide cada una de estas áreas dependiendo de la distancia que eligió el usuario anteriormente entre sensores. Además saber qué peligrosidades hay dentro de los puntos que pertenecen al polígono principal, además se necesita saber la posición que se encuentran, y así generar la matriz con esta información, la cual se enviará al algoritmo de despliegue de sensores. El algoritmo buscará la forma robusta de colocar cada uno de los sensores, y así tener cubierta todo el polígono principal.

Una vez que se ha generado el despliegue robusto devolverá una matriz, donde se recorrerá para la generación de un archivo *KML* con las distintas áreas. Para después enviar al correo que proporcionó el usuario para el despliegue, se le enviará el archivo *KML*. Para que este pueda cargar el mapa y así ver el despliegue en las áreas determinadas, también se le envía al usuario un archivo *TXT*, con cada una de las posiciones exactas de los sensores. Así podrá implantar cada uno de estos sensores en la posición exacta del área que se generaron, solo muestra la longitud y latitud de cada sensor. Dependiendo de cuantas áreas verdes haya generado el usuario, así será la cantidad de archivos que se le envíen al usuario.



## 4.6 Servidor de optimización

Se ha creado un servidor de optimización, porque hay demasiados cálculos que realizar para el despliegue robusto de sensores, además es demasiado elevado computacionalmente para calcular en la parte del usuario. Se realizó una búsqueda para saber, cuál podría ser el servicio que se debía utilizar para este proyecto, además debíamos prever que el sistema soportara el procesamiento este sistema web necesita.

Para la realización del servicio web, nos hemos apoyado en un proyecto de fin de carrera, realizado por Jiménez Espada, Jesús. Se trata de la resolución de problemas de optimización robusta, esto ayudó a decidir en qué lenguaje y de qué forma se iba a implementar para el cálculo del despliegue de sensores.

Se decidió Java para el servicio web, ya que tiene varias herramientas para esta aplicación y utilización de matemáticas. JMetal es una de las librerías que se utilizarán, proporciona las funciones necesarias para que el despliegue de sensores sea robusto, lo hace de forma evolutiva y esto le daba un valor añadido. Significa que va creando despliegues y descarta las peores evoluciones, así partirá de nuevo con las mejores evoluciones, eligiendo cual es la mejor evolución la cual se queda como candidata, por esto es un cálculo tan elevado en la utilización de este cómputo por la parte del cliente.

### 4.6.1 Comunicación entre servidores.

Para la comunicación entre servidores, se ha utilizado el protocolo Post por HTTP, con lo que se ha hecho necesario determinar cómo enviar datos de forma organizada además de segura. La transmisión de datos se hará una vez estén validados además se le dará un formato, que si alguien lo llega a conseguir, no sabría bien como utilizarlo.

Los lenguajes utilizados en cada una de las partes del sistema son:

- desde la parte del cliente se utiliza *Javascript*, que se necesita para la generación de las áreas en *Google Maps*.
- En la parte del servidor web se utilizó *PHP*, para las validaciones necesarias, esta transmisión se hizo a través del método Ajax. Que hace

llamadas al servidor, de forma que el cliente no tiene por qué darse cuenta de las operaciones que se están realizando. La llamada es a una función en *PHP*, la comunicación a este archivo es a través del protocolo Post. Ya que no se muestran en la barra de direcciones como hace el método GET, además se puede transmitir más datos, en esta función se validan los datos necesarios para el despliegue de sensores.



### 15 Formulario de despliegue, Normal y Avanzado

Una vez se han validado los datos, se hace el *parse* del KML enviado también por Ajax, en *PHP* se procesa para poder ser enviados posteriormente al servidor de optimización. Después que se han procesado y se ha realizado el *parse* para poder enviarse, se hace una conexión desde *PHP* con una herramienta que tiene incorporada llamada CURLOPT. Esta herramienta proporciona la posibilidad de enviar datos a través de Post a un servicio web. La llamada a este se realiza a través de una URI, esto hace que llegue al servidor de optimización con varias variables si la llamada se ha hecho correctamente.

En el servidor se ha generado un algoritmo para *desparsear* los datos que se envían, para que posteriormente estar recogidos en variables y ser utilizados. Se ha creado un algoritmo que obtiene las áreas con sus respectivos parámetros, se crea una matriz con los distintos puntos, de cada una de las áreas y peligrosidades. Cuando se ha creado esta matriz, se repite tantas veces como áreas verdes se hayan enviado.

Seguidamente se envía esto al algoritmo de optimización JMetal, en el cual se generan las matrices para el cálculo de la colocación de los distintos sensores, que se necesita para cubrir el área respetando las áreas con más peligrosidad. Una vez generada la matriz donde se tienen que colocar cada sensor, se genera un KML y un archivo por cada área verde que se haya enviado anteriormente. Con la posición exacta de cada sensor para un despliegue real en el área deseada.

Estos archivos, se envían al servidor de correo del usuario, y así tener la posibilidad de ver donde se colocan los sensores, si desea ver de una forma como quedarían los sensores, puede cargar el mapa KML en la aplicación web, Utilizando las herramientas proporcionadas para la carga de este.

#### **4.6.2 Cálculo de despliegue de sensores**

El cálculo se ha generado en un array de matrices, que son cada una de las generadas por el usuario. Se calculan en el servidor de optimización para la generación de datos, una vez concluido este cálculo y generadas las matrices, se le envía a esta herramienta. Se ha tenido que modificar para que pudiese recoger, calcular y devolver lo que se necesitaba en este proyecto, para posteriormente seguir con el proceso de envío de estos sensores desplegados.

Una vez que se hace el envío desde el servidor de optimización a la herramienta de despliegue, se comienza con el procesamiento y generación de un despliegue robusto. Como el usuario puede decidir de forma normal o avanzada para enviar los parámetros, se toman en cuenta los que están por defecto en el servidor o recoge los del usuario. Además esta función necesita parámetros adicionales que no pueden ser modificados por el usuario. Aquí se llama a la clase *despliegueSensores*, se van a enviar los parámetros más relevantes que se le puede enviar al algoritmo, los cuales son del usuario.

Son los siguientes:

- *rcom*: Opcional, sirve para el cálculo con la comunicación de sensores, por defecto es 40m.
- *rSen*: Opcional, es para el radio entre sensores, por defecto 40m.



## Capítulo 4 - Diseño

- *rEnergySensors*: Opcional, es para la energía de cada sensor, por defecto 30 de energía.
- *neighbors*: Opcional, es para saber con cuantos vecinos tiene que estar en comunicación al mismo tiempo, por defecto 30 vecinos.
- *nodesToUndeploy*: Generada por el sistema, es para saber cuántos nodos no tiene desplegar por cada evolución.
- *HECN*: Generada por el sistema, posición del sensor padre, este nunca se cae, es por donde sale y entra la información de los sensores, por defecto en el medio de cada área verde.
- *maxSens*: esta es proporcionada por un cálculo a través del usuario y la distancia entre sensores.
- *TerrenoUsuario*: es el área verde, con cada uno de las áreas de diferentes colores generados por el usuario.

Como se ha visto, hay varios parámetros para hacer el cálculo del área que se envían a esta clase, y se le hacen tantas llamadas como áreas verdes haya enviado el usuario.

Hay otros parámetros que están relacionados con las evoluciones del procesamiento:

- *populationSize*: Con cuanta población empieza a desplegar, por defecto 300.
- *maxEvaluation*: Opcional, Cuantas evoluciones va a generar para el despliegue robusto, por defecto 250000.
- *feedBack*: Cuantas veces hace *feedback* con las evoluciones.

Todos estos son importantes, pero el que más concierne al producto deseado, son las evoluciones que tendrá en cada área, para esto se le ha decidido que tenga 250000 evoluciones por defecto.

Esta clase llama a una función que se encuentra en la clase *RobustoSensores*. Es la que ejecuta el algoritmo de cálculo del despliegue de sensores, gracias a los parámetros que se le proporcionan y las funciones de las que hace uso, va generando distintos mapas de despliegues hasta encontrar una solución.

Primero se le envían las posiciones del área que está en la primera posición, y se llama a una clase que es la encargada de generar la matriz a partir de esos puntos enviados. Genera una matriz rectangular, lo suficientemente grande para que abarque toda el área verde que se envía, se compone de 0, 1, 2 y 3, para las diferentes peligrosidades que se encuentran en los otros polígonos que envía el usuario.

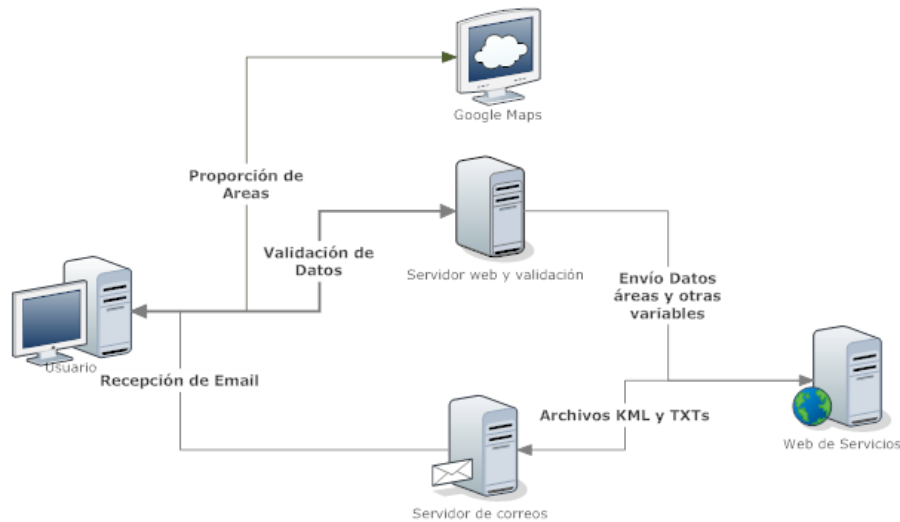
Otra clase importante que se llama es a Sensor, ya que esta es la que decide en qué posición va a estar el sensor y si estará desplegado, la posición que se genera del sensor es aleatoria, y aquí se determina en que área se está desplegando ese sensor. Esto significa que se decidirá dónde se va a desplegar el sensor, y cual tiene prioridad, de esto depende el área de peligrosidad donde se encuentre ubicado.

Una vez ha terminado, guarda las posibles soluciones en un archivo que tiene la generación de despliegues, en este momento se hace uso de una función. Esta función lee los archivos que ha generado el algoritmo, uno de ellos tiene más de una posible solución, se elige la mejor solución gracias a otro archivo que genera el cual tiene la mejor solución. Cuando averigua cuál es la mejor solución, genera la matriz a partir del archivo que tiene las soluciones.

Esto se hace tantas veces como áreas verdes ha mandado el usuario, una vez terminado cada una de estas áreas, se generan los archivos KML y TXT, igualmente se generan tantos como áreas principales. Una vez generados los archivos se procede al envío de ellos al correo electrónico que nos proporcionó el usuario, además se responde al servicio web con una respuesta de generación y envío de archivos correcta. Si por el contrario la generación de despliegues de sensores, no ha sido favorable porque no se han puesto unos parámetros idóneos para esas áreas, también se le mandaría un correo al usuario diciéndole que la generación no ha sido favorable, que proceda a intentar poner otros parámetros.

### **4.6.3 Esquemas comunicación entre servidores**

Este esquema muestra las llamadas que se realizan para poder generar el despliegue robusto generada por el usuario.

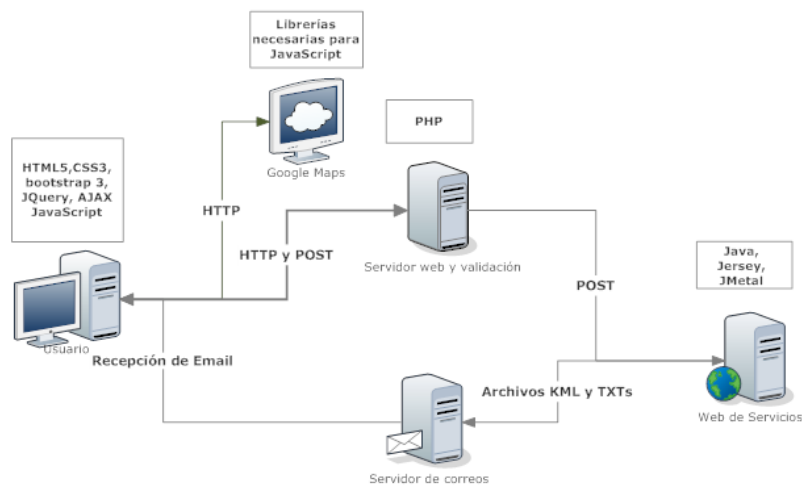


### 16 Comunicación servicio Web

#### 4.6.4 Esquema de lenguajes utilizados

En este apartado se quiere mostrar cuales son los lenguajes utilizados en las distintas partes del proyecto, cuáles han sido en el servidor del servicio web y en el servidor de optimización.

Se mostrará cada parte, donde está alojada y como se hacen las llamadas entre los distintos lenguajes, además de mostrar el funcionamiento entre servidores y el ordenador del usuario.



### 17 Lenguajes utilizados

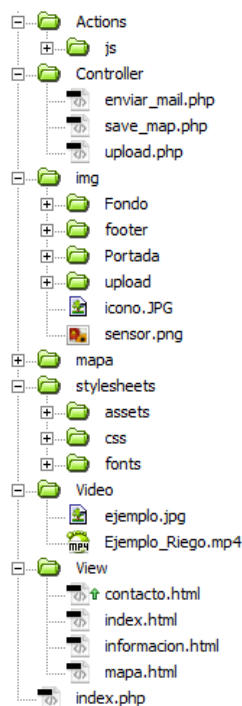
## 4.7 Esquema de archivos

Aquí se mostrará la organización de los archivos que se han generado de forma MVC (Modelo Vista Controlador), esto significa que están separados en capas por el controlador, la vista y del modelo. El modelo no se ha necesitado, ya que no se han creado clases para el control de usuarios finalmente, porque esta parte está más orientado a la carga de datos desde una base de datos. Que la página estuviera mucho más orientada a ser una red social, pero en este proyecto no se han incluido.

Al realizar la llamada a la raíz, se encuentra un archivo en *PHP* que redirige a la vista con una redirección 301, se encarga de mandar a la primera página que es home llamada *index.html*. Estas webs están relacionadas entre sí, por vinculaciones. Los controladores de las aplicaciones están en una carpeta aparte llamada *controller*. Aquí es donde se requieren las llamadas de algunas funciones para una buena organización de las carpetas.

Se han creado más en la raíz, *img* contienen todas las imágenes del servicio web, dentro de ella como se puede observar hay varias carpetas, donde se encuentran las imágenes de los distintos sitios de del servicio web. Dentro encontramos con otra carpeta llamada *video*, aquí se ha incluido el video de información que se está ubicado en el apartado Información. También hay una imagen que se muestra antes de reproducirse el video como foto principal, para no estar el elemento de video en negro. En *Stylesheets* es donde se encuentran todos los elementos de estilo, para el diseño que se le proporciona al servicio web, además de los estilos que se necesitan de las librerías de otros desarrolladores. En carpeta *Actions* incluidos todos los *Javascript* necesarios para las acciones y funciones, que se utilizan durante la manipulación de cada uno de los apartados de del servicio web. La carpeta *mapa* es la utilizada para el tránsito del mapa, como es guardado del mapa.

Imagen del esquema de archivos relacionados con del servicio web Smart Cultivation:



### 18 Esquema archivos

## 4.8 Optimización de Página web

Para la optimización del servicio web se van a emplear distintas herramientas que se han obtenido de internet, se arreglarán fallos que en la WWW los muestra como errores, fallos o problemas, para esto se ha utilizado la herramienta *w3c validator*.

### 4.8.1 Corrección de fallos HTML

El entorno de trabajo Dreamweaver tiene una herramienta en la cual se va a utilizar para encontrar fallos de código en HTML5, esta herramienta nos proporcionara la información necesaria para averiguar si tiene algún error, y así poder reparar el error que tenga constancia la (Consortium), como entidad oficial.

Para ello se han revisado cada una de las páginas y corrigiendo cada error que indicaba la W3C, esto es un proceso importante a la hora de cargar las paginas, pero aún más importante para el robot de Google. El cual siempre está buscando nuevos enlaces a páginas web, lo que hace es posicionar los servicios web, para que cuando se haga una búsqueda en su buscador tengan una posición, las páginas que tienen menos errores suben en este ranking.

Uno de los fallos más comunes que se producen en estos casos es la codificación de la web, también se producen otros como el cierre de etiquetas, cosa que HTML5 deben estar bien validados, tienen que estar bien organizadas las etiquetas ya que es un lenguaje de etiquetas.

### **4.8.2 Carga en dispositivos**

Gracias a la herramienta que proporciona Google, es relativamente nueva pero con ella se va a comprobar todos los errores que tiene el servicio web. Para que la carga de este servicio sea más rápida, esta herramienta también proporciona la capacidad de carga en cada uno de los dispositivos. Esto hace un gran beneficio este diseño de web, ya que está creada para un amplio mercado de dispositivos de distintos tamaños de pantalla.

#### **4.8.2.1 Optimizaciones de imágenes**

Primero hay que optimizar las imágenes, ya que esto es lo que más tarda en cargar del servicio web, con esta herramienta que nos proporciona un tutorial, se seguirá para una optimización de estas imágenes.

Como se ha propuesto en esta herramienta, se ha procedido a la reducción de imágenes que se realizó en Photoshop, ya que tiene una herramienta bastante específica, que es guardado de imágenes para páginas web. Con esta herramienta, se ha podido reducir notablemente la carga de todas las imágenes, y así reducir la carga completa en el servicio web para su visualización en los dispositivos.

#### **4.8.2.2 Optimización archivos**

Después se aconseja aprovechar el almacenamiento en cache del navegador, esta web tiene un tutorial que al seguirlo propone hacer algunos cambios en las diferentes páginas del servicio.

Una de las recomendaciones que tenía esta herramienta, es interesante si el servidor donde está alojado el servicio web, no tiene agregada la expiración de los archivos, estos servicios no son comunes en los servidores que alquilan sus servicios. En este caso el servidor no tiene este servicio activado, además no se puede activar, ya que el servicio contratado no lo permite, entonces esta optimización no se ha podido realizar.

También esta herramienta aconseja que no se provoquen bloqueos entre archivos CSS, pues esto causa un retraso en el procesamiento de la página que se está cargando.

La forma que se solventa esto es que cargue el archivo con menos relevancia el último, para esto es en la parte inferior del servicio web, una sustitución del archivo para que este el link en otra parte del código, con esto se ha solventado.

Otra forma de aumentar la velocidad de carga con los archivos CSS es minificarlos, esto es otra forma de contribuir a la aceleración.

Para esto se comprime con herramientas que están preparadas para reducir el código, esto hace que tengan menos caracteres y además optimiza la carga, pero no ha sido posible hacerlo. Habría que modificar archivos en el servidor donde tienen que dar permisos ya que donde se aloja no ha sido posible.

Con toda esta reducción que se ha podido realizar, y lo poco que no se ha podido hacer, se ha optimizado en un 30% en la carga de los dispositivos, por la reducción de estas, también ha aumentado la carga al hacer que los archivos CSS se carguen en un orden correcto, para que no haya solapamientos al cargar.

### **4.8.2.3 Conclusión**

Al haber realizado todos los cambios que se han obtenido del servicio web (Developers), se ha incrementado la velocidad de carga considerablemente. Los resultados mientras que antes de hacer las modificaciones tardaba entre 7 y 10 segundos en cargar, desde que se carga por primera vez entre 2.5 y 3 segundos para la carga sobre cache, donde ya estaban cargadas todas las imágenes y archivos. Al realizar la carga con la cache borrada la primera vez que se carga, ha pasado a una carga entre 4 y 4.5 segundos, a una carga con todos los elementos, se ha reducido a entre 0.5 y 0.7 segundos. Como puede comprobarse se ha optimizado bastante, sobre todo para la carga de dispositivos móviles.





# Capítulo 5

---

## Resultados

Para realizar estos resultados se ha apoyado en un ordenador de sobre mesa, Un *Ipad* mini y en un móvil *nexus* 5. Con esto se quiere demostrar que en todos los dispositivos se pueden visualizar y además inspeccionar todos los apartados sin problema, pudiendo leer y ver toda la información que se muestra en ellas. Se demostrará que se pueden utilizar las herramientas que se han estado creando durante todo el proyecto. Se empezará mostrando una imagen de cada dispositivo y con cada página que se vaya desarrollado, se mostrarán algunas más si son del dispositivo móvil, para una mayor visión de ellas.

### 5.1 Dispositivo ordenador sobremesa

Con este dispositivo se van a comprobar los resultados, por ello se va a mostrar una imagen de cada una de los apartados, para ver la diferencia que hay entre un dispositivo y otro. Con este dispositivo vamos a utilizar todas las herramientas que se han creado, la carga mapas y descargar mapas, para comprobar que todas están funcionando completamente.

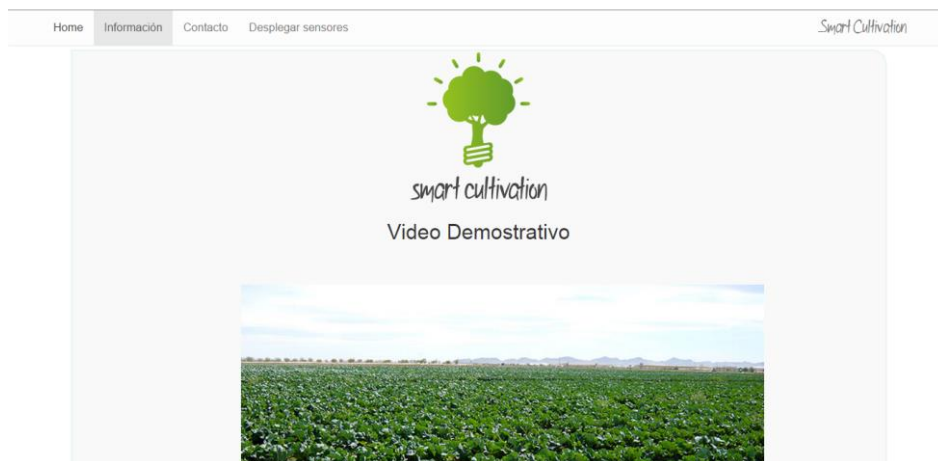
#### 5.1.1 Home



19 Home Ordenador

Aquí se muestra toda la información además de todas las imágenes con sus botones para más detalles, el menú adherido a la parte superior como se ha explicado en otros apartados, tiene la función los botones de deslizar la página hasta la información que corresponde a ese botón.

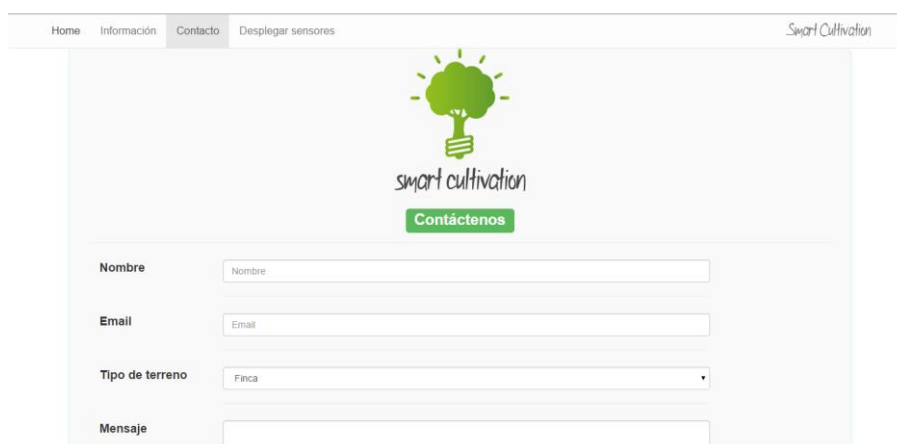
### 5.1.2 Información



#### 20 Información Ordenador

Aquí se puede observar que también está toda la información que tiene este apartado, además bien ordenada y como está explicado en el diseño de esta memoria. La funcionalidad de poder visualizar el video, pararlo o desplazar la barra de tiempo en este.

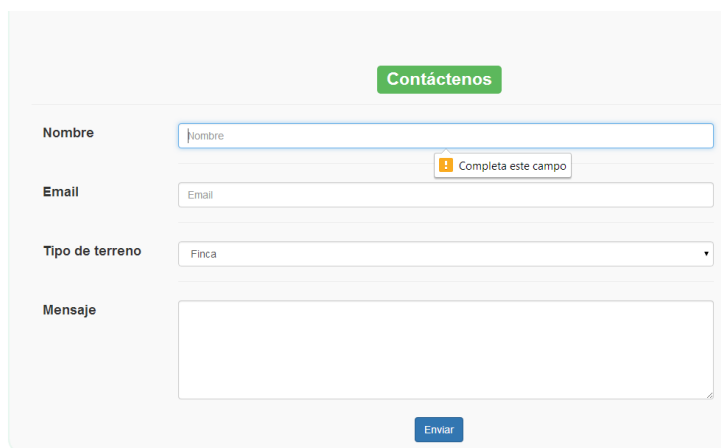
### 5.1.3 Contacto



#### 21 Contacto Ordenador

Como muestra la imagen también se ve todo, ya que en el ordenador todo se muestra y se expande al tamaño de la pantalla que se está visualizando. Se ve todo el contenido ya que esta página no tiene mucha información, solamente un formulario para el envío de correo electrónico a la persona que está encargada de recibirlo.

### 5.1.3.1 Prueba de correo



The screenshot shows a contact form titled "Contáctenos". It contains four main sections: "Nombre" with a text input field containing "Nombre" and a "Completar este campo" tooltip; "Email" with a text input field containing "Email"; "Tipo de terreno" with a dropdown menu showing "Finca"; and "Mensaje" with a large text area. An "Enviar" button is located at the bottom right.

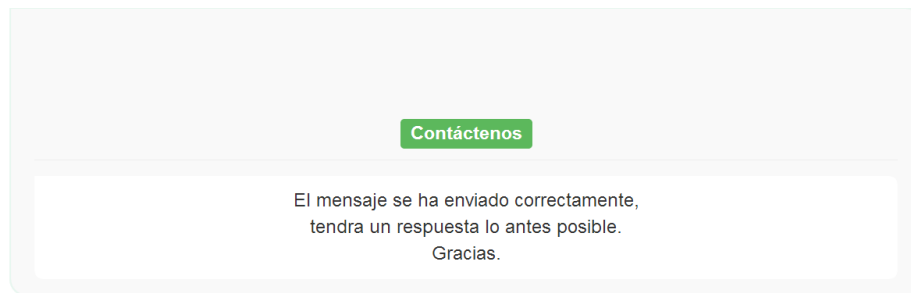
#### 22 Prueba correo campo incorrecto



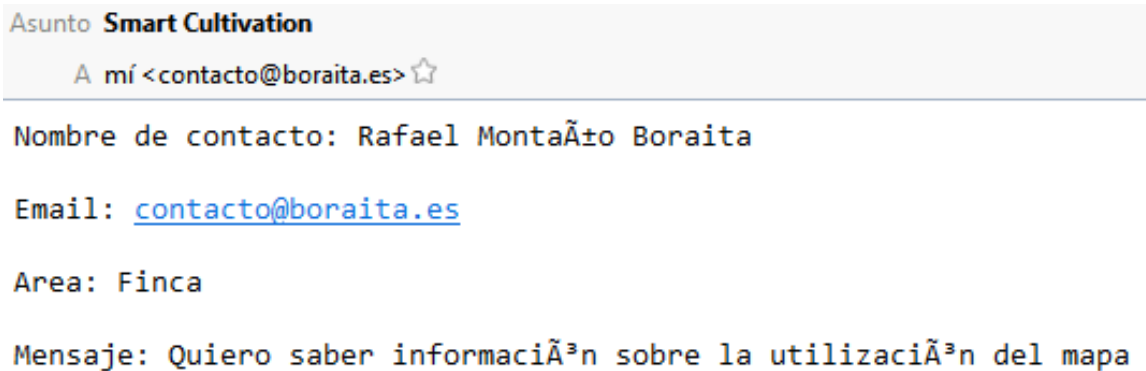
This screenshot shows the same contact form, but with an error message. The "Nombre" field contains "Rafael Montaña Boraita". The "Email" field contains "contacto" and is highlighted with a red border. A tooltip points to the field with the text: "Incluye un signo '@' en la dirección de correo electrónico. La dirección 'contacto' no incluye el signo '@'." The "Enviar" button is disabled.

#### 23 Prueba correo campo incorrecto

Como aquí se puede comprobar no se puede enviar un mensaje hasta no rellenar todos los campos correctamente, esta validación la tiene por defecto HTML5, es bastante útil para que el usuario se de cuenta si tiene algún fallo en el formulario.



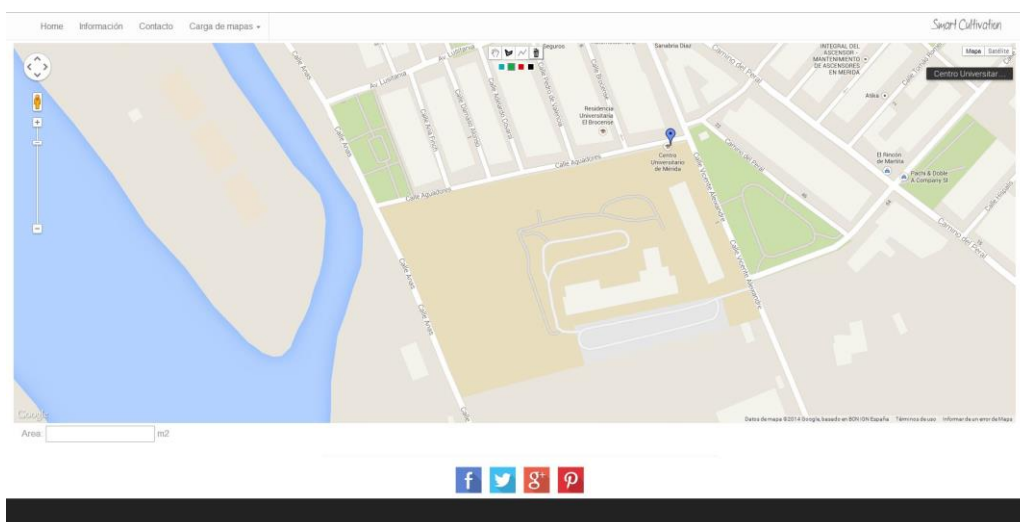
#### 24 Correo enviado correctamente



#### 25 Correo recibido correctamente

Como se puede ver en este se ve la transmisión correctamente creada, el problema de las letras no traducidas correctamente es problema del servidor de correo al enviarlo, ya que no hace el cambio entre el estandar UTF-8.

### 5.1.4 Desplegar sensores



#### 26 Despliegue sensores ordenador

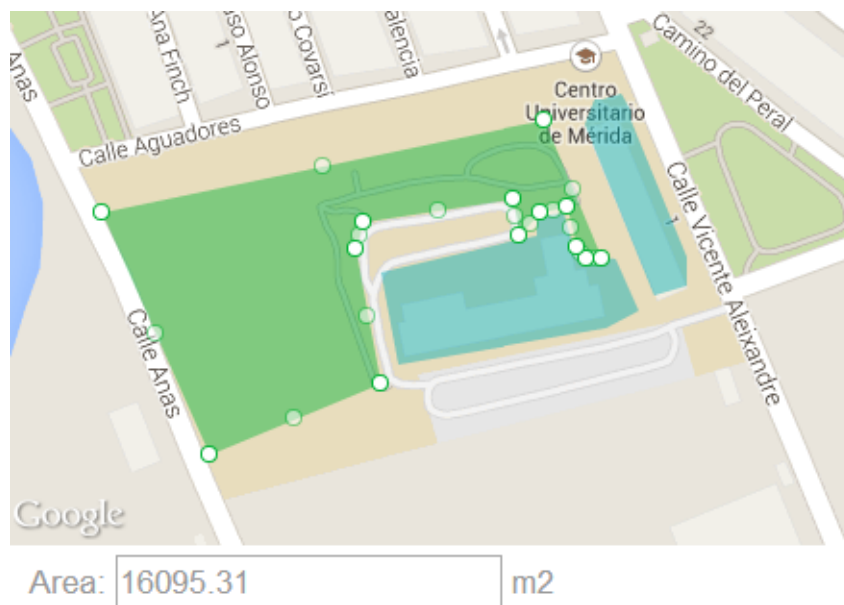
## Capítulo 5 - Resultados

Como se observa, se ven todas las herramientas, se ha usado la de búsqueda para ubicar el CUM, al poder utilizarse esta herramienta podemos darnos cuenta de lo potente e interesante que es tenerla para hacer una búsqueda con ella.

### 5.4.1.1 Crear polígonos



#### 27 Polígonos creados ordenador



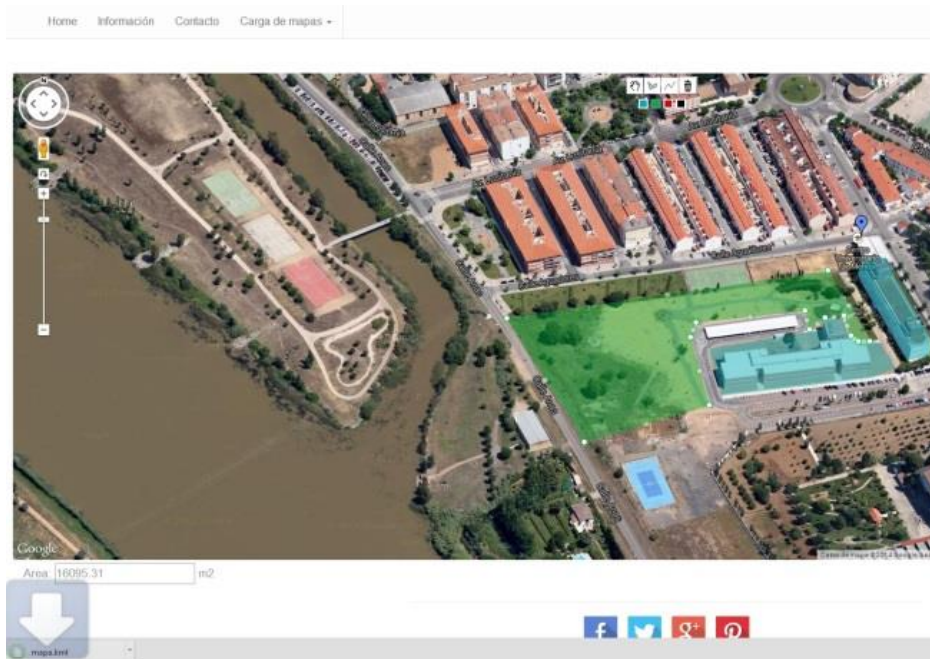
#### 28 Polígonos creados ordenador

Se han utilizado distintos colores para los polígonos para la utilización de esta herramienta, además se está utilizando una herramienta para el cálculo del área, el tamaño es del polígono que está señalado, al tener un ratón en este dispositivo se puede detallar mucho más las posiciones del polígono.

### 5.4.1.2 Descarga archivo mapa



29 Descarga mapa ordenador



30 Mapa descargado ordenador

Se han creado polígonos que son áreas para el despliegue de sensores, con esta herramienta se han podido descargar los polígonos, como veremos más adelante solo se podrá utilizar esta herramienta de descarga de mapas para este dispositivo. Ya que no se creyó conveniente agregarla a los dispositivos más pequeños, porque esto se ha diseñado para poder modificar o ampliar las áreas que se están diseñando. Esta descarga sirve para hacer modificaciones

en un futuro, con esto podrá cargar en el mapa y que aparezcan igual que los terminó para poder editarlos si así lo cree conveniente el usuario.

### 5.4.1.3 Borrar polígonos



31 Polígonos borrados ordenador

Se puede ver que se han borrado dos de tres áreas que se han creado antes con la herramienta borrado, se puede hacer esto si ya el usuario no necesita esta área o para una creación nueva del área.

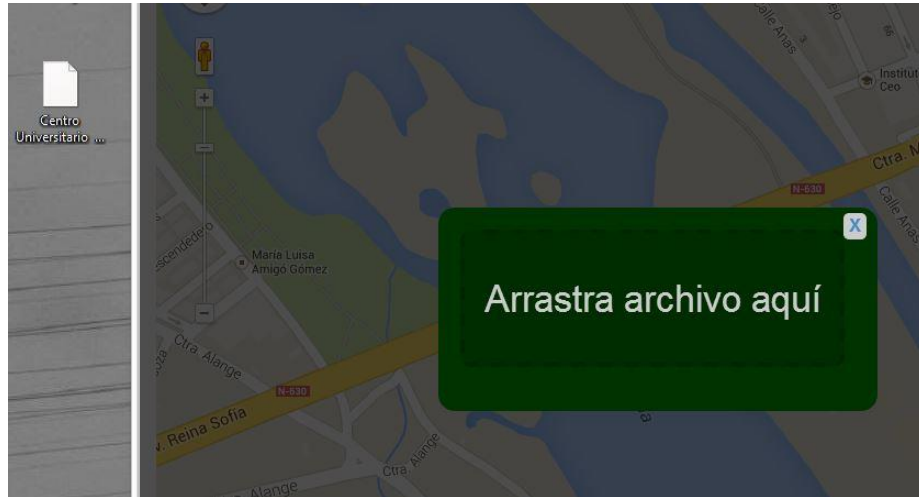
### 5.4.1.4 Carga de áreas



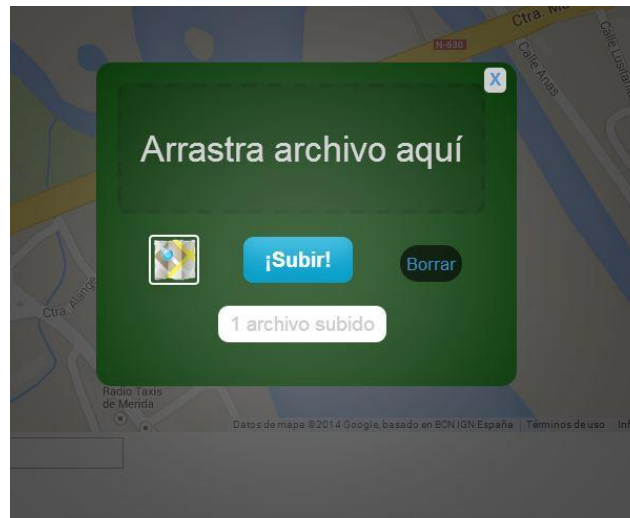
32 Carga archivo áreas ordenador

## Capítulo 5 - Resultados

Como se puede observar en esta imagen, en la parte de detrás del cuadrado verde (DIV), dos de los polígonos están borrados, al cargar se añaden al mapa solo los que no estaban, esto hace que sea una gran herramienta, ya que puede volver a cargar el mapa sin que se añada uno que ya tiene realizado.

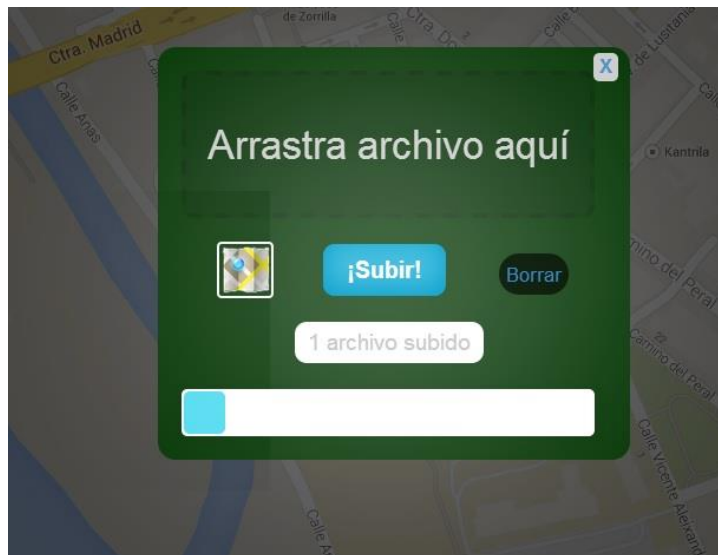


### 33 Arrastrando archivo ordenador



### 34 Archivo cargado ordenador





### 35 Subiendo archivo ordenador



### 36 Áreas cargadas ordenador

Como se ha visto en el proceso la carga se ha realizado con éxito, se han incorporado los dos polígonos que faltaban en este mapa, si en el archivo hubiera polígonos de otra parte del mundo también los añadiría, esto sirve por si alguna persona se dedica a poner los sensores en distintos campos, así no tiene que estar creando varios archivos.

### 5.4.1.4 Sensores desplegados

#### 5.4.1.4.1 Mapa sin peligrosidad sin robustez



#### 37 Selección de área sencilla

### Despliegue Sensores

Perfiles para despliegue: Normal  Avanzado

Radio Comunicación(20/100):

Radio Sensor(20/100):

Vecinos comunicados(1/50):

Estrategia: N° 1  N° 2

HECN (Sensor principal): Centro  Posición segura

Iteraciones(20K/300K):

Terreno/s:

Distancia en metros (unidad de discretización):

#### 38 Parámetros de área sencilla



#### 39 Sensores desplegados Sencillo

Como se puede apreciar aquí se ha creado un mapa en campo, se ha quedado sin peligrosidad y se ha desplegado de una forma muy básica. Como se puede apreciar en el despliegue los sensores, están desplegados y al no tener mucha conexión por los parámetros enviados no tienen el despliegue robusto que buscamos.

##### 5.4.1.4.1 Mapa con peligrosidad sin robustez



#### 40 Área con distintas peligrosidades

En esta área se han insertado zonas con diferentes peligrosidades, ya que se está simulando además una zona donde no se pueden colocar sensores. Con este despliegue se demostrará que hay mayor cantidad de sensores en las zonas con mayor riesgo.

### Despliegue Sensores

Perfiles para despliegue: Normal  Avanzado

Radio Comunicación(20/100):

Radio Sensor(20/100):

Vecinos comunicados(1/50):

Estrategia: N° 1  N° 2

HECN (Sensor principal): Centro  Posición segura

Iteraciones(20K/300K):

Terreno/s:

Distancia en metros (unidad de discretización):

Email:

La petición ha sido enviada correctamente.

#### 41 Parámetros de baja robustez



#### 42 Despliegue peligrosidad y baja robustez

Como se puede observar con unos parámetros mediocres, la discretización y robustez no es adecuada, por esto se ha hecho el método normal, para el usuario que no entienda que parámetros son los adecuados.

## Capítulo 5 - Resultados

El tiempo que ha tardado en hacer este despliegue han sido sobre 5 minutos, ya que no tienen muchas evoluciones, ni calcular la posición con sus vecinos ya que tiene puesto 1, además de los otros parámetros que son bajos para un buen despliegue.



### 43 área sin peligrosidad y robusta

Este es el mapa que se va a utilizar en las dos pruebas, la diferencia va a ser que uno de ellos tendrá polígonos con peligrosidades y tendrá otros donde no se pueden incorporar ningún sensor.

Despliegue de sensores con un alto grado de robustez, para esto se va a decidir hacerlo de forma avanzada y poner los parámetros que se cree conveniente. Para este despliegue se tarda alrededor de 10 horas, ya que es un área muy grande y se le ha enviado que necesita hacer muchas evoluciones.

Una captura de pantalla de una interfaz de usuario con el título 'Despliegue Sensores'. El formulario contiene los siguientes campos y valores:

Perfiles para despliegue:	Normal <input type="radio"/> Avanzado <input checked="" type="radio"/>
Radio Comunicación(20/100):	40
Radio Sensor(20/100):	40
Vecinos comunicados(1/50):	10
Estrategia:	Nº 1 <input checked="" type="radio"/> Nº 2 <input type="radio"/>
HECN (Sensor principal):	Centro <input checked="" type="radio"/> Posición segura <input type="radio"/>
Iteraciones(20K/300K):	100000
Terreno/s:	296940.95
Distancia en metros (unidad de discretización):	100

### 44 Parámetros despliegue robusto

Como se observa en esta imagen, tiene un grado alto de robustez para la práctica que se va a hacer, estos parámetros son lo que se van a usar para proceder al despliegue robusto.

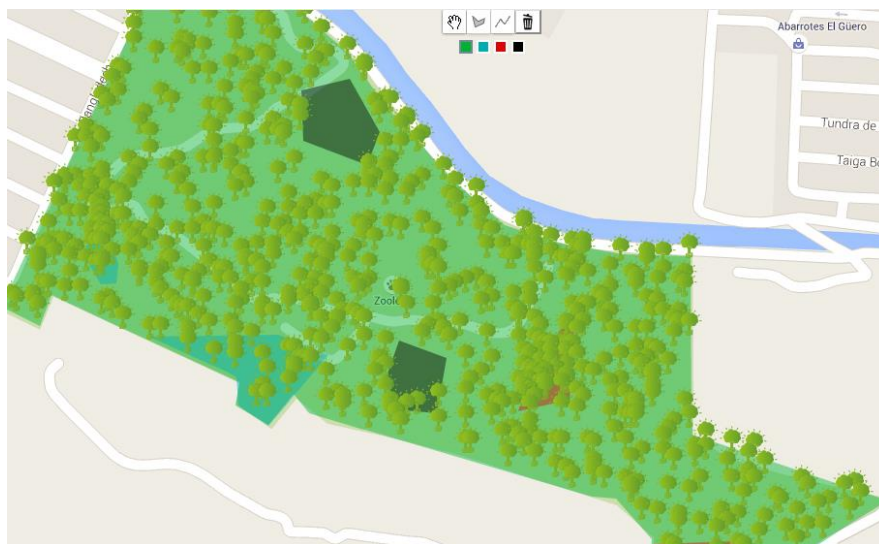
### 5.4.1.4.1 Mapa sin peligrosidad con robustez



#### 45 Sensores desplegados sin peligrosidad

Como podemos visualizar, los sensores están distribuidos de un forma uniforme, cada uno de ellos tiene que estar conectado con 10 vecinos, como se propuso en los paramentos.

### 5.4.1.4.1 Mapa con peligrosidades con robustez

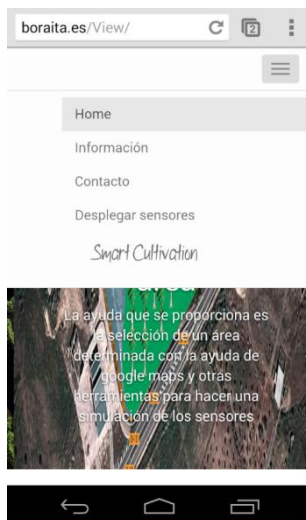


#### 46 Sensores desplegados con robustez y peligrosidades

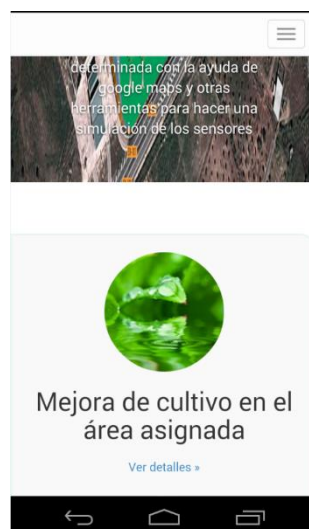
En esta imagen se puede ver que en la zona donde hay peligrosidades hay más cantidad de sensores, también se ve que en las zonas negras no hay ningún sensor, ya que es una zona que no puede haber.

## 5.2 Dispositivo Nexus 5

### 5.2.1 Home



47 Menú en home



48 Home móvil



49 Home móvil

Como se puede observar en las imágenes están todos los datos que integran en el servicio web de home, pero como también se puede observar faltan algunas imágenes con un tamaño más grande, como se menciona se eliminan las imágenes menos relevantes. En este caso son las grandes, para dejar espacio a las más pequeñas, ya que estas si son posibles poder visualizarse en los dispositivos más pequeños.

## 5.2.2 Información



50 Información móvil

51 Información móvil

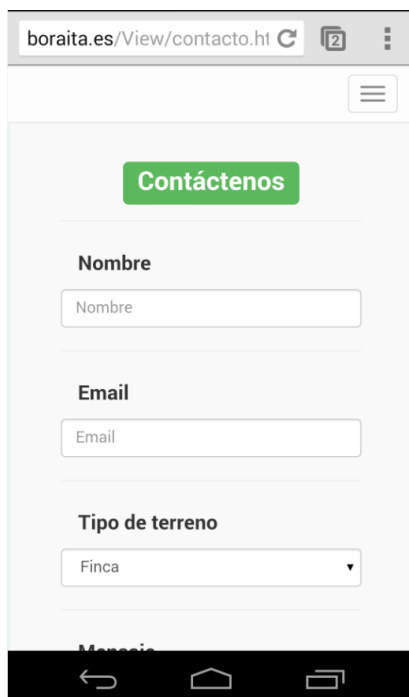
Observamos en las dos fotografías dependiendo del ancho de la pantalla se amolda la información, el video se queda igual pero se puede reproducir al mismo tiempo que se va cambiando de posición. Tiene un botón de ampliación o disminución para la reproducción del video, esto hace que el usuario elija la forma que desea ver el video.



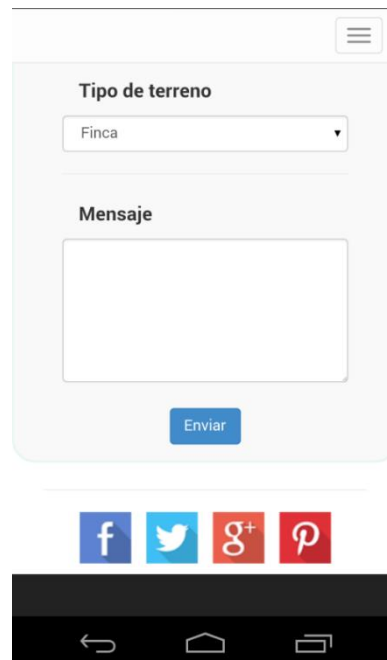
52 Reproduciendo video completo



## 5.2.3 Contacto



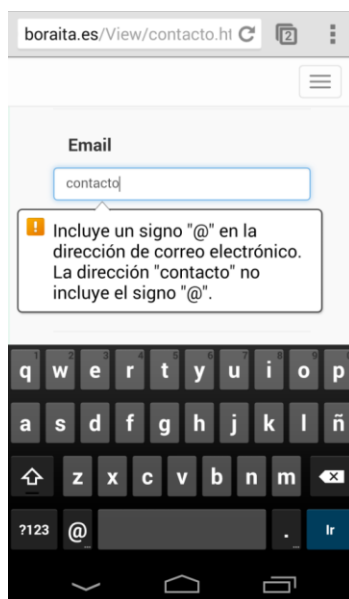
53 Contacto móvil



54 Contacto móvil

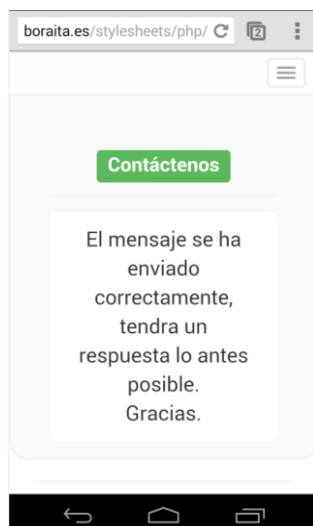
En estas imágenes los datos que se quieren mostrar se ven perfectamente, además se puede usar cualquier elemento. Dependiendo del campo del formulario que se seleccione, saldrá un teclado u otro, esto es otra herramienta que se está utilizando de HTML5, hace que el usuario pueda utilizar las teclas que necesita por defecto, para una mayor facilidad y que se escriban menos errores en el envío de datos.

### 5.2.3.1 Prueba contacto en móvil



#### 55 Error enviar mensaje móvil

Como se puede ver en el móvil también tiene la herramienta, para detectar el fallo de no introducir correctamente en los apartados, con los elementos mal escritos no puede enviarse el correo, ya que para esto se incorporó la validación en *PHP*.

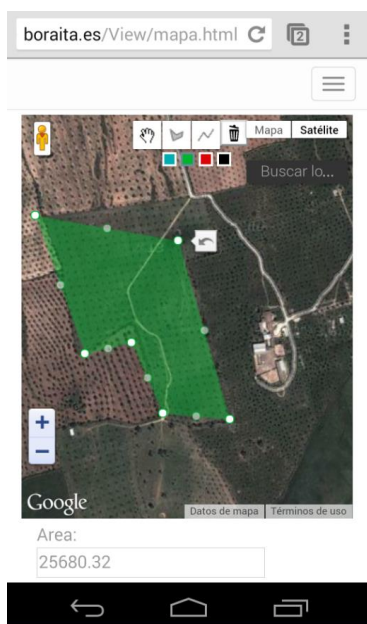


#### 56 Envió correcto móvil

Aquí se han introducido los datos correctamente, por esto se muestra el mensaje de correo enviado correctamente.

## 5.2.4 Desplegar sensores

### 5.2.4.1 Herramientas de áreas



57 Creando polígono móvil



58 Ampliando área móvil

Observando, vemos que funciona la herramienta de ampliación de polígono, una vez que el área se ha creado funciona en todos los dispositivos como se mencionó en el diseño. Se ve también la herramienta la cual se muestra si uno de los puntos del área se ha movido de posición, se puede hacer un retroceso de este punto a donde estaba. Se puede observar que la herramienta de cálculo de área también está haciendo su desempeño, el mostrar cuanto mide el área señalada. Esto nos hace ver que las herramientas se ven y funcionan correctamente en cualquier dispositivo.



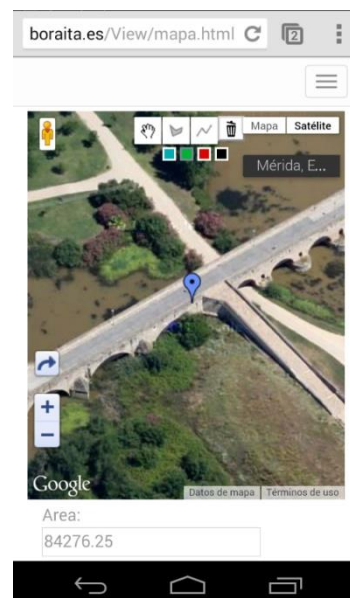
59 Leyenda colores móvil

Aquí se puede ver como se muestra la leyenda de la paleta de colores, pulsando encima de uno de los colores saldrá qué diferencia hay en poner un color u otro para el despliegue de sensores, esta desaparecerá una vez que se le pulse fuera del área que tiene contemplada esta leyenda.

### 5.2.4.2 Herramienta búsqueda situación



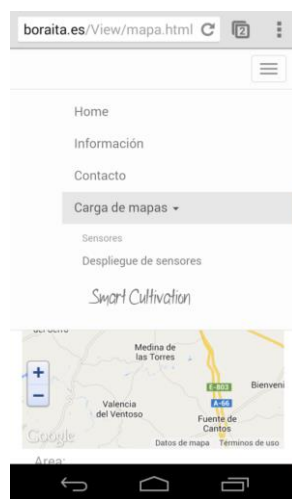
60 Búsqueda móvi



61 Sitio encontrado móvil

Se puede utilizar la herramienta para buscar una situación que esté en la base de datos de Google, como se observa tiene una flecha donde se puede voltear el mapa, ya que esta ciudad tiene la propiedad de estar en 3D, si no lo tuviera no se mostraría. Al poder utilizar esta herramienta nos facilita ir a la posición que queremos, ya que en estos dispositivos tenemos una pantalla más pequeña.

### 5.2.4.3 Guardar y cargar archivo



#### 62 Despliegue menú carga mapa móvil

Como se puede observar en los dispositivos más reducidos, no se permite el cargar ni guardar los mapas, por esto no aparecen los botones correspondientes a estas herramientas. Solo se puede mandar a desplegar los sensores, ya que en el dispositivo no se permite hacer otro trámite que no sea enviar el área para el despliegue de los sensores, este haría el proceso y después mandaría el correo correspondiente. Tiene también la herramienta de avanzado porque no necesita hacer el cálculo, ni cargar el mapa desde este dispositivo.

## 5.3 Dispositivo Ipad Mini

### 5.3.1 Home



63 Home Ipad



64 Home Ipad

En las imágenes este dispositivo tiene el suficiente tamaño para poder mostrar toda la información y todas las imágenes, pero esto cambia si se pone en modo vertical. En esta posición no puede mostrar las imágenes más grandes, así que lo que hace es recolocar las pequeñas, para adquirir la forma apropiada y así poder visualizar todo el contenido.

## 5.3.2 Información

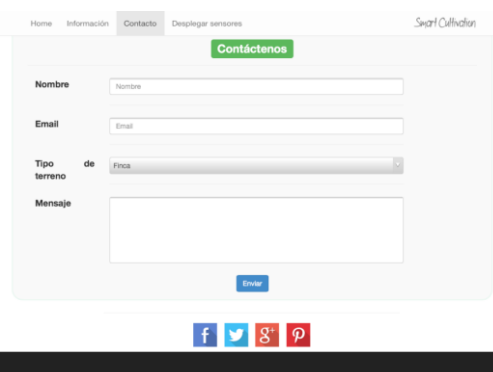


### 65 Información Ipad

### 66 Información Ipad

Se puede ver claramente la diferencia que hay entre la pantalla del ordenador y la de un dispositivo mediano, al no ser lo suficientemente grande en vez de mostrar la información en dos columnas, se amolda para que se vea la mayor información posible. Al colocarse un texto debajo de otro también puede reproducirse el video, además se le puede manejar como en los otros dispositivos, ya que las herramientas del video está diseñado para que se puedan manipular en cualquier dispositivo.

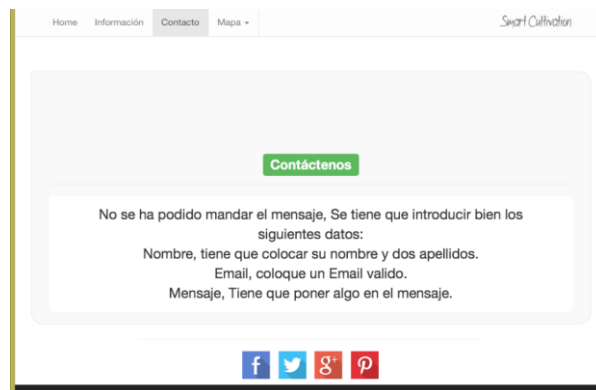
## 5.3.3 Contacto



### 67 Contacto Ipad

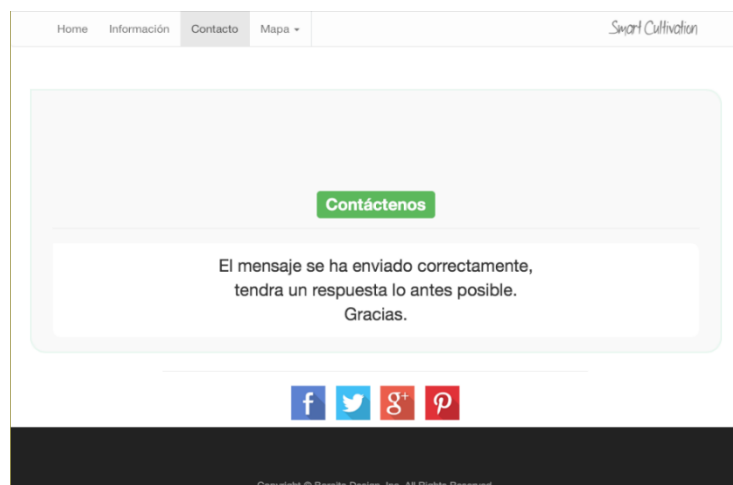
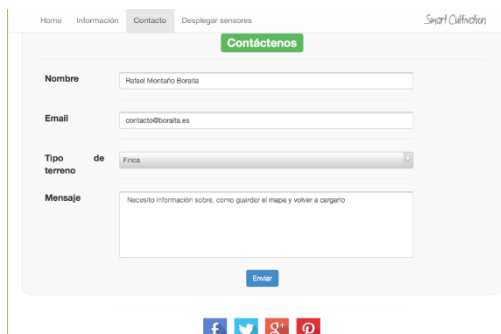
Como se demuestra en esta imagen se reproduce todo perfectamente, además que se pueden ver los campos para poder escribir, está formateado de la misma manera que en el dispositivo de ordenador, ya que tiene la envergadura de una pantalla lo suficientemente grande para poder visualizar todo a la vez.

### 5.3.3.1 Prueba contacto Ipad



#### 68 Fallo contacto Ipad

En el iPad vemos que no funcionan las reglas que se han incorporado para la validación de los campos en su totalidad, por lo tanto la validación tiene que hacerse por *PHP* en el servidor. Como se muestra aquí, al enviarlo sin ningún campo relleno se recibe un mensaje de fallo en el envío, ya que el navegador que se está usando en el iPad no está compatibilidad completa con HTML5.



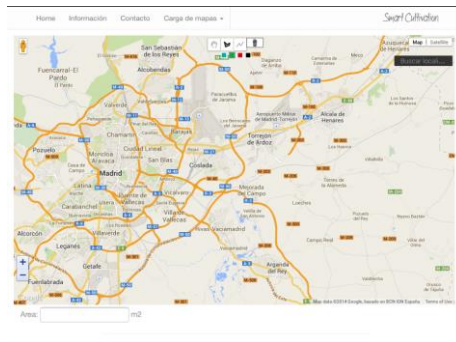
#### 69 Relleno de contacto Ipad

#### 70 Confirmación contacto Ipad

En estas imágenes al rellenar los campos debidamente se envía la pregunta al contacto correspondiente, se muestra el mensaje donde el correo se ha enviado correctamente, aquí habrá que esperar a la respuesta.

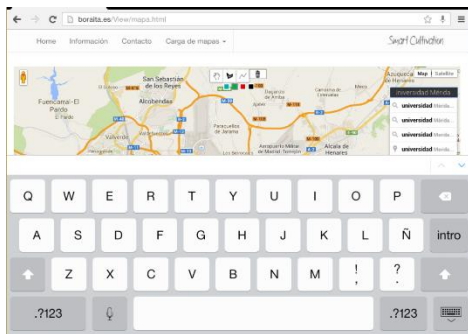


### 5.3.4 Desplegar sensores

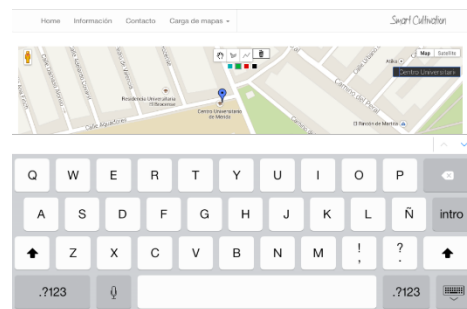


71 Desplegar sensores Ipad

#### 5.3.4.1 Herramienta buscar situación



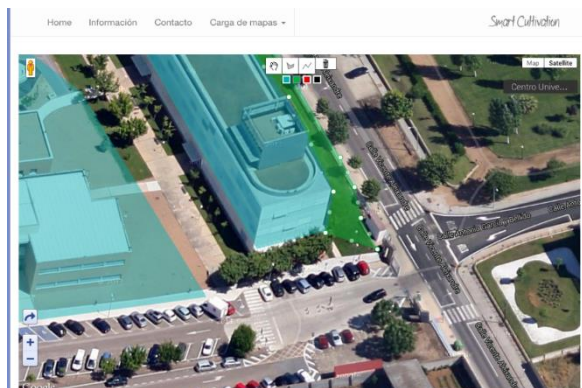
72 Herramienta situación Ipad



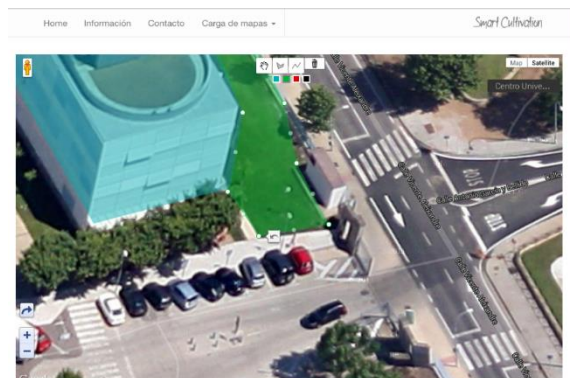
73 Situación encontrada Ipad

Se ha buscado el Centro Universitario de Mérida, al encontrarlo se pulsa la tecla intro y el mapa se reubica, en la posición donde está el punto de la situación, como podemos observar esta herramienta se puede utilizar en cualquier dispositivo.

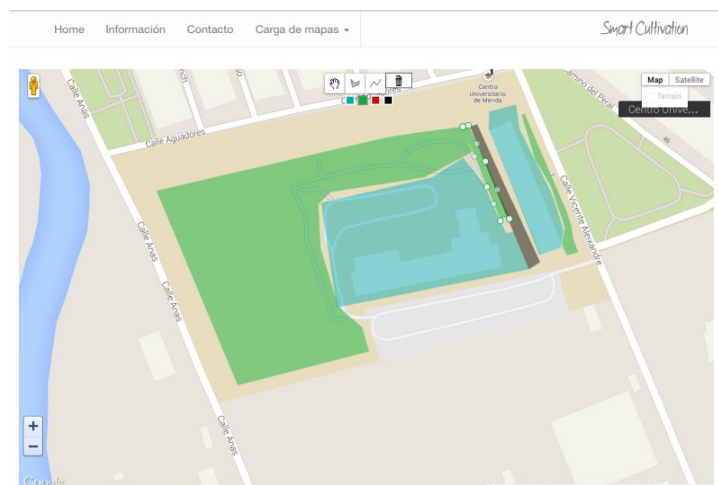
### 5.3.4.2 Herramienta crear áreas



74 Creación de área Ipad



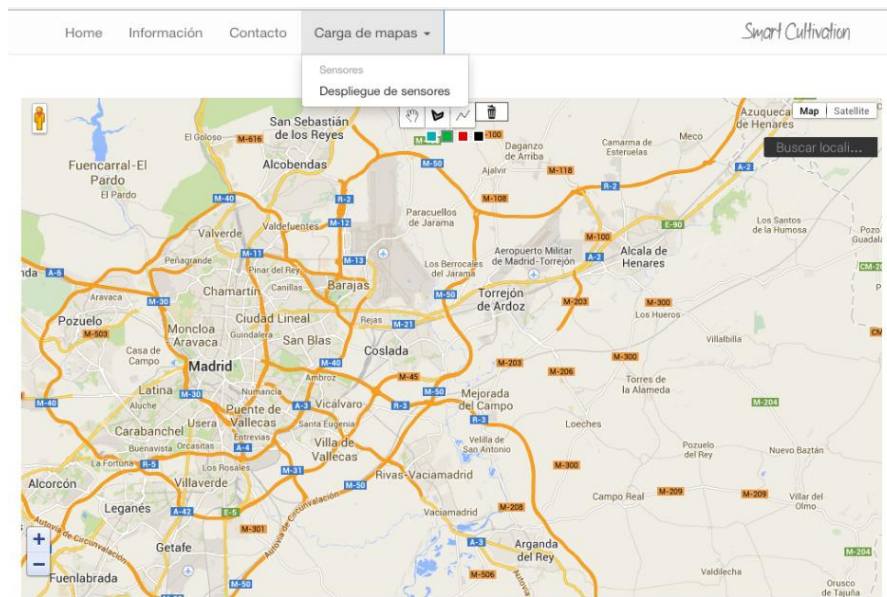
75 Modificación de área



76 Muestra de áreas creadas Ipad

Se puede hacer cualquier modificación con los áreas, moverlas, redimensionarlas, borrarlas, cambiarles el color, tiene todas las herramientas igual que en el ordenador y en el dispositivo *Nexus 5*, por eso es tan versátil este servicio web, y así se quería hacer.

### 5.3.4.3 Cargar y guardar mapa de áreas



#### 77 Carga de mapas Ipad

Como se demuestra en la imagen los botones de cargar mapa y descargar mapa, no están disponibles en los dispositivos con un tamaño más reducido que el de un ordenador, por el problema que se explicó anteriormente.



## Capítulo 6

---

### Conclusiones y trabajo futuro

En este capítulo procederemos a la exposición de las conclusiones obtenidas tras la realización de este proyecto de fin de grado. También se verán las posibles funcionalidades que se pueden incorporar al trabajo realizado. Finalmente se declararán los agradecimientos.

#### 6.1 Conclusiones

1. La búsqueda y análisis de distintos lenguajes de programación de diseño web además de las librerías que se han recopilado, ha sido una labor ardua pero necesaria para poder afrontar el desarrollo del servicio web.

Con este sistema se ha conseguido que adquiriera el valor que se deseaba, gracias a la elección que se hizo en las decisiones de incorporar estos lenguajes, gracias a esto podrá actualizarse y ampliarse este proyecto. La implementación de estos ha supuesto la mayor labor y parte del tiempo, ya que fue necesaria una compatibilidad entre los lenguajes además de una fácil modernización del sistema.

2. La creación de este servicio web ha demostrado cómo pueden ser combinados varios lenguajes de programación, añadiendo librerías para la aplicación y sus operaciones además de librerías de diseño. Para una mayor compenetración se han elegido de forma correcta la colocación e integración de estas. El diseño del servicio web ha sido un punto importante, ya que el usuario además de tener funcionalidad, puede interactúa mejor si la aplicación es estéticamente bonita, este punto se ha conseguido en este proyecto.
3. La creación de distintas funciones ha sido laborioso, ya que al utilizar librerías de otros desarrolladores hay que estudiarlas a fondo, lo que supone mucho tiempo en el proceso de aprendizaje de estas para ir avanzando en el proyecto. Hay muchas pruebas que realizar para que dé el resultado esperado, también hay que comprobar que vayan

funcionando en los navegadores más usados, que es lo que se pretendía que hiciese este servicio web, además de en los dispositivos con distintos tamaños. Ya que era otra parte importante de este proyecto, añadir y modificar las funciones necesarias para las llamadas entre ellas fuese rápida y segura.

4. La comunicación que se ha conseguido entre las funciones de distintas librerías aunque estén en otro servidor son realmente rápidas. Por esto se han incluido funciones que están en otro servicio web, donde se aloja la librería necesaria aun así tiene un retraso mínimo de respuesta. Como se ha podido ver en la parte de velocidad de carga, esto hace mucho más versátil que el sistema esté en cualquier servidor del mundo y utilizar las herramientas desde otro.
5. Se han encontrado diversas dificultades con el diseño para todos los navegadores y los dispositivos posibles. Por lo que se ha necesitado invertir tiempo en ir probando cada uno de los dispositivos si todo funcionaba correctamente, además que la información quedara en las posiciones que se quería.

Hay un estándar en navegadores para la reproducción del sistema, pero en realidad cada uno lo reproduce a su manera, esto entorpeció mucho en el diseño del servicio, al igual que se han encontrado problemas al aplicar funciones, ya que no todos tienen incorporados los mismos *plug in*. Están probados en muchos de los navegadores y se ha solventado los problemas que se han ido encontrando. Con los dispositivos no ha habido tantas diferencias para el diseño y las funciones, aun siendo dos sistemas completamente diferentes y navegadores diferentes.

6. Se ha realizado una gran labor al optimizar al máximo la carga de los distintos dispositivos, lo cual es un gran avance para que los usuarios no tengan retraso en el tiempo de carga, lo cual ocurre con otras páginas web.
7. Se ha creado una herramienta muy útil para usuarios que puedan necesitar un riego eficiente, ya que cada vez la tecnología avanza más, sería un gran avance el poder usarla para la ayuda de una producción con más calidad. Este servicio web creado puede ayudar a visualizar el entorno donde se quiere desplegar de forma real, además de visualizar los sensores colocados en los sitios exactos. Cuando un sistema como este genera un despliegue de sensores de forma robusta en un área

decidida, supone un menor gasto en la implantación del sistema de riego. Ya que no será necesario contratar a alguien que orientase de la misma manera que el sistema, con la única diferencia sería el comprar dichos sensores e implantarlos en el terreno, de esta manera tendría un despliegue de sensores eficiente y robusto.

8. La comunicación entre los diferentes sensores no ha sido complicado, pero si se ha necesitado mucho tiempo, ya que se necesitaban conectar diferentes tecnologías. Para esto fue necesario hacer muchas pruebas, además de ir probando cual es la manera más eficiente de enviar los parámetros de forma segura y que el sistema necesitaba.
9. Con el algoritmo de optimización que calcula el mejor despliegue de sensores posible, se ha invertido también mucho tiempo, ya que se ha reutilizado de otro trabajo previo. A sido muy laborioso el estudiar la parte del trabajo que se necesitaba, además de cada una de las clases necesarias para este proyecto, en un principio fue complicado encontrar como este sistema hacía los cálculos.

Este sistema se creó para algo parecido, por esto hubo que modificar las funciones del sistema, para que así pudiese recibir y devolver aquello que se necesitaba. Después que se hicieron los cambios, hubo que añadir clases para poder exportar al mapa ya generado, además de enviar los archivos con las áreas del usuario.

## 6.2 Ampliaciones y trabajo futuro

Entre las posibles de ampliación que se pueden realizar a este trabajo y que no han podido afrontarse por falta de tiempo, vamos a destacar algunas:

1. Crear una gestión de usuarios que pudiese acabar en una red social, donde pudieran intercambiar las experiencias que han tenido aquellos usuarios registrados en este servicio web.
2. Utilizar este servicio web para una ampliación y convertirla además en una tienda online, donde se pudiera vender todo lo necesario para que los usuarios desplegasen físicamente los sensores, y así se podría calcular el precio que conlleva todo el despliegue. La realización de este despliegue de sensores en el área decidida también podría requerir soporte y asistencia, previo pago de transporte y mano de obra.

3. Crear una aplicación móvil que se comunicara con el servicio web para saber en qué sitios colocar los sensores, además de ir con el dispositivo móvil para colocar en el área adecuada y en los sitios proceden a la colocación exacta mediante GPS.





## Capítulo 7

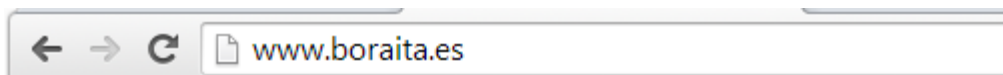
---

### Manual del usuario

Este capítulo es el encargado de explicar cómo se utiliza todo el proyecto creado, abordaremos cada apartado y dentro de éste se abordaran las distintas herramientas que estén implementadas en cada uno.

#### 7.1 Acceder a Smart Cultivation

Para acceder al servidor donde se encuentra alojada este servicio web se necesita un navegador, en la barra de direcciones se le pone como aquí pone.

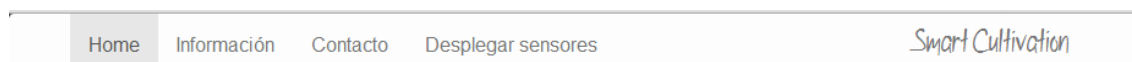


##### 78 Barra dirección

Una vez en está el cursor en la barra de direcciones se tiene que escribir la siguiente dirección, [www.boraita.es](http://www.boraita.es) o como *mirror* está también alojada en [www.boraita.freeiz.com](http://www.boraita.freeiz.com), una vez haya accedido a este servicio web, se redirige para acceder a la página home.

#### 7.2 Menú y Footer

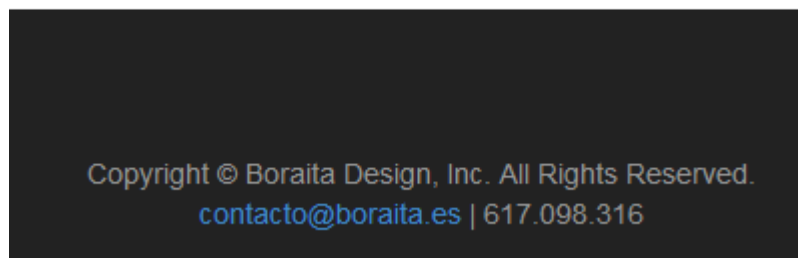
En este apartado como se puede observar tiene un menú, desde donde puede dirigirse a cualquier apartado de los que hay, pulsando encima de cualquiera de ellos.



##### 79 Menú de navegación

Para el manejo del *footer*, hay cuatro imágenes donde se pueden hacer clic o *touch* sobre ellas, haciendo esto se redirigirá a la red social que corresponde el

logotipo que se haya elegido, también está la información referente a la persona quien ha desarrollado este servicio.



### 80 Footer

## 7.2 Home

Para observar la información que hay en Home, solo hay que utilizar el *scroll* del navegador para subir o bajar que se está visualizando, también se puede utilizar los botones que están justo debajo de las imágenes más pequeñas, donde se verá más información. Esto se deslizará suavemente de forma vertical hacia la parte donde está la información, la cual corresponde al tipo de información que se desea ver.



### 81 Botones información Home

## 7.3 Información

Este apartado está creado para la información sobre la utilización del sistema, que como se puede observar, para la visualización solamente hay que desplazar el *scroll* del navegador, para ir leyendo la información sobre que tiene este servicio web y como utilizar el mapa de Google para la creación de áreas y hacer el despliegue de sensores.

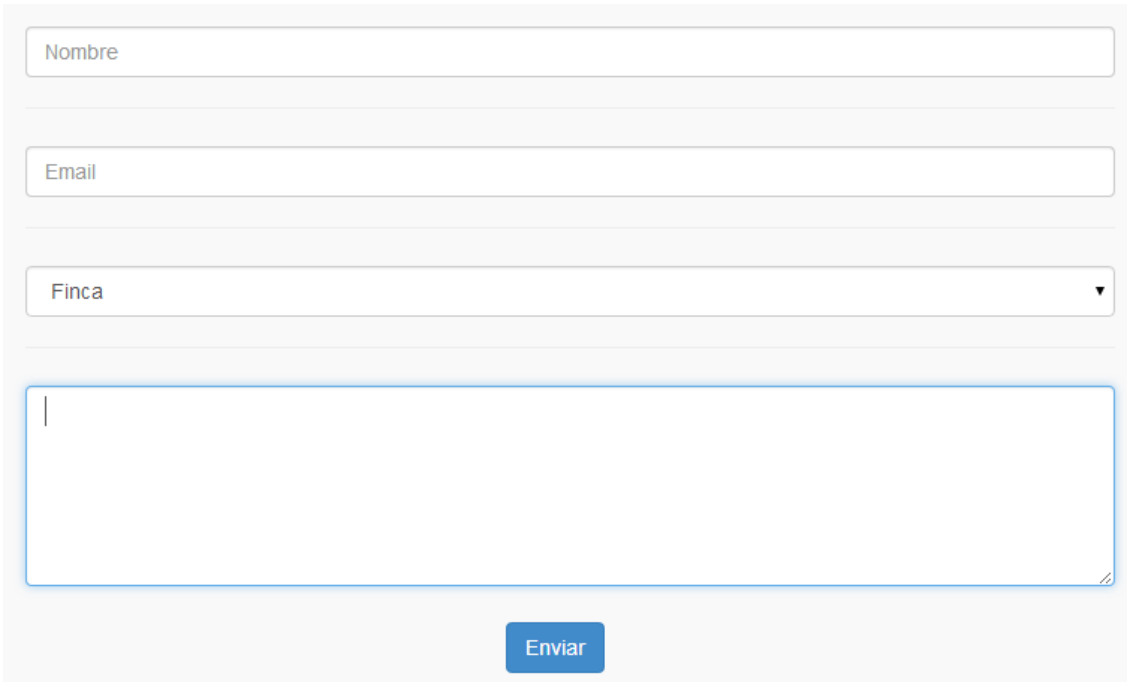
En este apartado hay un vídeo en el cual pulsando en el medio del recuadro empezará a reproducirse, tiene controles adicionales para la manipulación del vídeo.



### 82 Controles de vídeo

## 7.4 contacto

Para poder hacer uso del apartado contacto, hay que rellenar un sencillo formulario, tiene indicado cada uno de los campos a rellenar pulsando encima de ellos con cualquier dispositivo, ya sea con un clic del ratón o un *touch* de un dispositivo móvil. Se podrá escribir en estos campos solamente a razón del texto que requiere explicada en su parte izquierda. Una vez rellenado los datos correctamente, se le pulsará al botón enviar, si todos los datos están correctamente rellenados, se enviará la consulta al sistema.



Formulario de contacto con los siguientes campos:

- Nombre
- Email
- Finca
- Área de texto grande
- Botón Enviar

### 83 Formulario de contacto

## 7.5 Desplegar sensores

Aquí se podrán realizar las operaciones con áreas y líneas que requieran para hacer un despliegue de sensores, además de poder guardar o cargar todas las áreas que se hayan creado con esta herramienta. Se podrá utilizar el mapa para la localización de lugares o situaciones que el usuario crea conveniente, además de poder manipular el aspecto en el que se muestra el mapa donde está seleccionando el área.

### 7.5.1 Manipulación de mapa

Para empezar con la manipulación de mapas, primero hay que saber que cuando se carga este apartado de desplegar sensores, el navegador pedirá al usuario si quiere proporcionar la dirección donde se encuentra el dispositivo con el cual se está accediendo. El mapa mostrará la situación donde se encuentra el dispositivo si se pulsa que si decide proporcionársela, mostrará por defecto una dirección ubicada que es en Madrid al no proporcionar una dirección.

Las herramientas que proporciona el mapa de Google son para acercar o alejar la altura en la que se encuentra el mapa, además de si el usuario requiere dirigirlo en alguna posición.



#### 84 herramientas Google maps

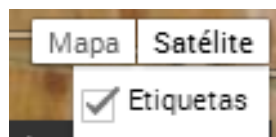
Esta propiedad también la tienen el ratón o el *touch* del dispositivo móvil, manteniendo el cursor del ratón pulsado con el botón izquierdo, moviendo el ratón se conseguirá que el mapa se mueva en esa dirección. Si es un dispositivo móvil, pulsando con el *touch* y sin retirar el dedo, arrastrándolo por la pantalla el mapa se moverá en esa dirección. Con la rueda del ratón, la posibilidad de acercar o alejar en distancia al mapa donde tiene colocado el cursor.

Con un dispositivo móvil tiene que hacerse con dos dedos, si se pulsa con dos dedos la pantalla y se arrastran, para acercar el mapa, tienen que separar la distancia entre dedos, para acercar el mapa es justo al contrario, acercando los dedos entre sí.

Para cambiar de vista el mapa, porque el usuario así lo quiere o le es más sencillo la manipulación en un tipo de mapa que en otro para la creación del área, hay dos botones alojados en la parte superior derecha. Estos dos botones, al pulsarlos o hacer *touch* en uno de ellos, el mapa cambia para

pasar a ser un mapa de un aspecto diferente. El aspecto es para ver situaciones de carreteras o lugares que se destacan más en esta situación, al pulsar encima da la opción de decidir si lo prefiere con relieve o sin relieve.

En cambio el otro botón es un aspecto de satélite, es la visualización más real ya que son fotografías tomadas desde un satélite en el espacio. Con este aspecto se puede distinguir las delimitaciones que hay con los campos cercanos del área que pueda elegir el usuario, este botón también tiene un check, que al tener activo muestra las etiquetas de los lugares en el mapa. Como pueden ser los nombres de las poblaciones, los nombres de las carreteras o nombre de situaciones que se ven desde la altura, si se deselecciona no se mostrará nada de eso.



85 Botón satélite con etiquetas



86 Botón Mapa con relieve

## 7.5.2 Herramientas de áreas



87 Herramientas áreas

Como se puede observar en la imagen 70 esto tiene cuatro recuadros que son herramientas para el procesamiento de las áreas, además de cuatro recuadros de colores son los que obtienen los polígonos que se dibujan.

Empezando por la herramienta de la izquierda, se puede mover el mapa si no se está encima de algún área, sí por el contrario se pulsa encima de un área con esta herramienta se moverá hacia la posición que se mueva el ratón o touch.

La segunda herramienta por la izquierda es un polígono, donde teniendo seleccionada esta herramienta se podrán hacer polígonos, pulsando en tantos puntos como sea necesario tener seleccionada el área deseada, con un mínimo de tres puntos. Cuando se desea que el polígono se cierre hay dos maneras, una de ellas es pulsando sobre el primer punto que se puso para empezar el área, y la segunda manera, es dando doble clic con el ratón o el *touch*, en la parte que se decida que desde ahí se haga el cierre. Esto supone que el último punto que se ha pulsado, se haga la unión con el primer punto que se creó a través de una línea recta, aquí se le deja un ejemplo.



#### 88 Área creada

Como se puede ver en la imagen tiene dos tipos de puntos, unos de color blanco y otros son de color blanco transparente, los puntos blancos son los sitios que se han marcado para ir creando el área. Estos puntos pueden volverse a pulsar, para agrandar o disminuir el área de una forma más exacta. Para los puntos de un blanco transparente, tienen la misma utilidad que los otros puntos, con la diferencia que al pulsar sobre ellos y arrastrarlos hacia otra posición para hacer el área más exacta, se convierte en un punto blanco, además aparecen otros dos puntos de forma blanco transparente.

Refiriéndonos al punto arrastrado, sus dos vecinos son de blanco transparente, en el cual se puede hacer lo mismo con estos para hacer el polígono más exacto. Otra herramienta que esta oculta si no se arrastra



ningún punto blanco es la de retroceso, sirve para poder rectificar el último punto arrastrado, en el caso de no haberlo colocado bien en la posición se puede retroceder al paso anterior, esta herramienta solo aparece en este caso.

Como se puede ver también en esta imagen, en el recuadro aparece la dimensión del área en m<sup>2</sup>.

Este recuadro se rellena si se pulsa encima de un área, para saber cuánto es la medida que tiene, se borra una vez se haga *click* en otra parte del mapa en la que no hay un polígono creado.

Area: 46756.64 m<sup>2</sup>

#### 89 Dimensión de área

### 7.5.3 Herramienta Línea

La herramienta que está en la tercera posición por la parte izquierda de la imagen 70, es para dibujar una línea con tantos puntos como se desee, pero tiene que tener al menos dos para ser útil y se pueda ver en el mapa, sirve para restringir un área que requiera resaltar el usuario. Esta línea tiene las mismas propiedades que un área menos la propiedad de la dimensión, la utilización de esta herramienta es igual que la de un área.



90 Línea dibujada

### 7.5.4 Herramienta Borrar elemento

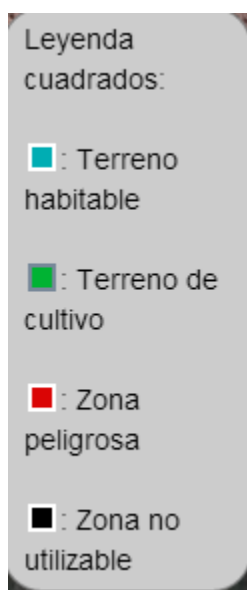
Para borrar un elemento, primero tiene que estar debidamente seleccionado. Hay que tener seleccionada la herramienta movimiento que es la primera por la izquierda de la imagen 70, hay que hacer clic o *touch* encima del elemento que se desea borrar, a este elemento le volverán a aparecer los puntos blancos y blancos transparentes. Al pulsar la herramienta papelera que es la cuarta por la izquierda de la imagen 70, automáticamente desaparecerá el elemento seleccionado, y esto hará que se quite también del array de polígonos creado, así no será guardará ni se utilizará para el despliegue.



91 Área eliminada

### 7.5.5 Herramienta paleta de colores

La paleta de colores son los cuadrados que están en la parte de debajo de la imagen 70, cada uno de estos colores significa una peligrosidad para el despliegue de sensores. La leyenda de estos colores se muestra al pasar el ratón por encima de la paleta o haciendo *touch* encima de esta, una vez que el ratón no esté encima de la paleta o se haga *touch* con un dispositivo móvil en otro lugar, se ocultará de nuevo la leyenda para dejar una mejor visión del mapa que se está utilizando, este es el ejemplo de la leyenda.



### 92 Leyenda paleta de colores

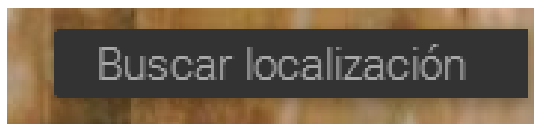
Para elegir cualquiera de los cuatro colores que hay en esta paleta, hay que hacer clic o *touch* encima de ella. Si se quiere cambiar el color de un área que ya está en el mapa, primero hay que seleccionar el área que explicado en el punto 7.5.2, cuando esté seleccionada simplemente hay que hacer clic o *touch* encima del color al cual se quiere cambiar. Si lo deseado es hacer el área o la línea de un color en concreto, primero hay que seleccionar el color y después se selecciona la herramienta crear polígono o línea, puntos 7.5.1 y 7.5.2 respectivamente, Aquí está el ejemplo de cambiar de color la línea.



### 93 Línea cambiada de color

### 7.5.6 Herramienta búsqueda de lugar

Esta herramienta se encuentra en la parte superior derecha de la pantalla del mapa, es un rectángulo negro, pulsando sobre este con un clic del ratón o con el *touch* del dispositivo se podrán poner letras para buscar un lugar o situación que se desee. Esta herramienta tiene la propiedad de desplegar una lista de los sitios o lugares, las cuales empiezan por letras que se han pulsado. Si aparece el lugar en esa lista, solo hay que pulsar con el ratón o el *touch*, directamente en el nombre. El mapa se moverá hasta esa situación, si por el contrario no aparece en esta lista, se le puede dar *Enter* en el teclado del dispositivo que se esté utilizando entonces se moverá hasta la posición. Si hay más de una posición, el mapa se alejará lo suficiente como para mostrar todos los puntos que ha encontrado.



94 Herramienta buscar



# Agradecimientos

Después de todo el esfuerzo realizado durante estos 4 años de carrera, tengo mucho que agradecer a muchas personas, no podría nombrar a todas así que aquí me gustaría recordar algunas de las más significativas.

A la persona que más tengo que agradecer es a Daniel Lanza García, él ha sido un gran amigo, compañero de carrera al igual que de piso y profesor en muchas ocasiones, estoy seguro de que si él no hubiese estado en la carrera como compañero, no me hubiera podido obtener el grado de ingeniería de la manera que he llegado a conseguir.

También tengo que dar las gracias a mis compañeros de piso, Carlos Verdejo Merchán y Jorge Sousa Villafaina, ellos me han soportado y me han hecho tener a conseguir unos amigos para siempre, tengo que agradecerles el apoyo que me han otorgado, al igual de los momentos inolvidables vividos a su lado.

Quiero agradecer a mis directores de proyecto Francisco Luna Valero y Rafael Marcos Luque Baena, que a pesar de haber estado en el final de la carrera, también he podido comprobar lo buenos profesores que son.

La mejor elección que he tomado es haber estudiado en Universidad de Extremadura en Mérida, no he podido elegir un mejor lugar, ni una mejor carrera. Ya que los profesores que allí están impartiendo, además de tener un conocimiento amplio en las materias que necesité para adquirir, también han hecho que sacara lo mejor de mí. Todo esto no se puede agradecer lo suficiente en especial a Luís Arévalo Rosado, Francisco Chávez de la O, Héctor Sánchez Santamaría y Miguel Ángel Andugar Hernández. Ellos han sido los profesores con los que más he aprendido además de guiarme profesionalmente.

Por último y no menos importante, quiero dar las gracias a toda mi familia y amigos, que me han estado apoyando en todo momento durante la carrera, en especial a Elena Aguilera Balboa que gracias a ella, he podido empezar mi carrera profesional de la forma que yo deseaba. A toda mi clase, ya que nos hemos mantenido unidos y me llevo buenos amigos de esta etapa de la vida.

Con la mano en el corazón, Rafael Montaña Boraita

## Bibliografía

**anieto2k.** anieto2k. [Online] <http://www.anieto2k.com/>.

**Azure, Carlos. 2013.** Empezando con AngularJS. [En línea] 9 de Septiembre de 2013. <http://carlosazaustre.es/blog/empezando-con-angular-js/>.

**Bidelman, Eric.** html5rocks. [Online] <http://www.html5rocks.com/en/tutorials/file/dndfiles/>.

**BlitzMap.** [Online]

**bootply, Team.** bootply. [Online] <http://www.bootply.com/>.

**Consortium, World Wide Web.** W3C. [Online] <http://www.w3c.es/>.

**desarrolloweb.com.** desarrolloweb.com. [Online] <http://www.desarrolloweb.com/articulos>.

**Developers, Google.** Google Developers. [Online] <https://developers.google.com/maps/documentation/javascript/geometry?hl=es>.

**Espada Jiménez, Jesús.** *Trabajo fin de carrera.* [Documento digital]

**Experts, jQuery Communitu. 2010.** *jQuery Cookbook.* s.l. : OReally media, 2010.

**Flanagan, David. 2011.** *jQuery.* s.l. : OReally Media, 2011.

GitHub. <https://github.com/twbs/bootstrap/releases>. [Online]

**Google.** Google Nexus 5. [Online] <http://www.google.es/nexus/5/>.

**jmcunningham. 2013.** Github. [Online] Enero 2, 2013. <https://github.com/jmcunningham/AngularJS-Learning>.

**Klinggaard, Bjoern.** dinbror. [Online] <http://dinbror.dk>.

**Knol, Alex. 2013.** *Dependency Injection with AngularJS*. s.l. : Packt Publishing, 2013.

**Malopinsky, Ivan.** imskv. [Online] <http://imsky.co>.

**MDN.** Mozilla Developer network. [Online] <https://developer.mozilla.org>.

**Pinterest.** Pinterest. [Online]  
<http://es.pinterest.com/search/pins/?q=website%20design&rs=rs>.

**Robbin, Scott.** srobbin.com. [Online] <http://srobbin.com/jquery-plugins/>.

**Sebadri, Brad Green & Sbyam. 2013.** *AngularJS*. s.l. : O`Reilly Media, 2013.

**Simpsons, Johnny.** Inserthtml. [Online] <http://www.inserthtml.com/>.

**Spurlock, Jake. 2013.** *Bootstrap*. s.l. : OReilly, 2013.

**Stackoverflow.** Stackoverflow. [Online] <http://stackoverflow.com/questions>.

**Surguy, Maks.** bootsnipp. [Online] <http://bootsnipp.com/>.

**team, Core.** getbootstrap. [Online] <http://getbootstrap.com/>.

**Twitter.** bootstrap-transition. [Online]  
<http://getbootstrap.com/2.3.2/javascript.html#transitions>.

**Varios.** Jquery write less, do more. [Online] <http://jquery-manual.blogspot.com.es/>.

**w3schools.** w3schools. [Online] <http://www.w3schools.com/>.