



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

CAMPUS D'ALCOI

# Aplicación web para el control de la concentración de desinfectante residual en redes de agua de consumo humano

---

Trabajo Final de Grado Ingeniería Informática

Rubén Canals Català

Tutor: Javier Esparza Peidro

Defensa: Septiembre de 2017

## Tabla de contenido

Introducción .....	1
1.1 Motivación .....	1
1.2 Estudio del mercado.....	1
1.3 Objetivos del proyecto.....	2
1.4 Estructura del resto de la memoria .....	2
2. Análisis o definición del problema .....	4
2.1 Proceso actual.....	4
2.2 Solución propuesta.....	5
2.3 Requisitos.....	5
2.3.1 Funcionalidades de usuario .....	5
2.3.2 Requisitos indispensables.....	6
2.3.3 Requisitos deseables .....	7
2.4 Planificación .....	7
3. Solución del problema.....	8
3.1 Arquitectura de la aplicación .....	8
3.2 Tecnologías utilizadas.....	9
3.2.1 Lenguajes de programación.....	10
3.2.2 Frameworks / Herramientas .....	14
3.3 Capas .....	22
3.3.1 Capa de datos .....	23
3.3.2 Capa de negocio .....	26
3.3.3 Capa de presentación .....	27
4. Resultado o Tour por la aplicación.....	29
4.1 Login.....	29
4.2 Panel de administración .....	30
4.3 Vistas.....	35
4.4 Panel de gestor .....	41
4.5 Control de datos .....	42
4.6 Diseño .....	44
4.6.1 Vista desde pantalla de escritorio .....	45
4.6.2 Vista desde teléfono móvil .....	45
7. Buenas prácticas.....	47
8. Conclusiones y trabajos futuros .....	47
9. Bibliografía.....	49

## Introducción

### 1.1 Motivación

Este proyecto se basa en la creación de una aplicación web para el control de la cantidad de desinfectante residual en el agua destinada al consumo humano, con el fin de ofrecer mayores garantías de calidad a los consumidores.

Este proyecto viene motivado por la necesidad de automatizar el proceso de toma de muestras y de gestión de los elementos que lo integran, ya que hasta este momento, este proceso se realizaba de forma manual, tomando notas y comunicando los resultados a través de fax.

Las principales desventajas del modo en que se realiza el proceso actualmente son, la cantidad de tiempo que se invierte en la toma de datos, el envío de los mismos para su almacenamiento, la cantidad de papel que se utiliza durante el proceso, y la actualización de dichos datos en el sistema. Por tanto, con el desarrollo de esta aplicación se pretende cubrir la necesidad de automatización del proceso en cuestión, reduciendo los tiempos de actualización de los datos y mejorando la comunicación entre el centro de control de datos, en este caso, el centro de salud de Alcoy, y las personas encargadas de tomar las muestras.

Se consigue un “feedback” entre el equipo de administración y los técnicos encargados de tomar las muestras casi instantáneo.

Hoy en día, existe un gran abanico de tecnologías que nos permiten abordar este problema para obtener una solución de garantías las cuales estudiaremos en los siguientes apartados.

### 1.2 Estudio del mercado

En otras zonas de España ya notaron la necesidad de automatizar el proceso de control del agua, y desarrollaron sus propias aplicaciones informáticas para cubrir dicha necesidad.

Pese a que el funcionamiento de dichas aplicaciones difiere en algunos aspectos al de este proyecto, el fin es el mismo.

El primer ejemplo de aplicación similar que se ha encontrado, se desarrolló en la provincia de Alicante en 2014. La aplicación en cuestión se denomina iCAP y fue desarrollada por la empresa “Aqualogy Aqua Ambiente”.

<http://www.hidraqua.es/es/noticias/22/la-calidad-del-agua-del-taibilla-a-traves-de-una-aplicacion-informatica>

Como se puede observar en la noticia, esta aplicación está basada en la recogida de datos de la red de distribución y genera alertas vía email y móvil a los correspondientes técnicos encargados de su mantenimiento en caso de que los valores no sean aceptables.

Otra aplicación que ha sido desarrollada con fines similares es la siguiente:

<http://www.euskaditecnologia.com/deustotech-colabora-con-open-data-euskadi-para-desarrollar-una-aplicacion-sobre-la-calidad-de-las-aguas-en-euskadi/>

Es un programa que muestra de forma interactiva mediante un mapa, las diferentes tomas de agua existentes en el País Vasco, dando información del estado del agua en tiempo real.

Fue desarrollada por investigadores de la unidad “DeustoTech-INTERNET” con el fin de informar a los consumidores, ya que según estos investigadores, el agua puede ser potable o no dependiendo de la época del año.

### **1.3 Objetivos del proyecto**

A la hora de desarrollar cualquier proyecto, el primer paso fundamental es analizar detenidamente los objetivos que se pretenden satisfacer con dicha aplicación antes de abordarlo. El objetivo principal por el cual se elabora este proyecto, es el de diseñar y desarrollar una aplicación informática basada en tecnología web con el fin de satisfacer la necesidad de informatizar un proceso de toma de datos que actualmente se realiza de forma manual.

Además del objetivo de informatizar dicho proceso, también se plantea la creación de un panel de control dentro de la misma aplicación, para que los datos recogidos durante el proceso, puedan ser gestionados por las personas que correspondan.

Más adelante se describirán con más detalle, los requisitos que necesita cubrir la aplicación para que ofrezca una solución de garantías al problema planteado.

.

### **1.4 Estructura del resto de la memoria**

Todo lo explicado en los puntos anteriores ha sido una breve introducción al proyecto, para dar a conocer al lector, la problemática que se planteaba y el motivo por el cual se ha decidido implementar este proyecto.

A partir de este punto, se empieza a describir los diferentes aspectos de la aplicación con un mayor detalle. De manera general, se expondrán los siguientes apartados:

### **Análisis previo**

Consiste en un estudio de las necesidades y requisitos que se pretenden satisfacer para que la aplicación sea de utilidad, buscando qué aspectos son más relevantes y plantear posibles soluciones para cubrir las necesidades vigentes cada uno de ellos.

### **Herramientas utilizadas**

En este apartado, se busca elegir las tecnologías disponibles en el mercado que permitan desarrollar una solución al problema. También se realizará una explicación de los motivos por los cuales se han elegido dichas tecnologías, dando a conocer algunos datos relevantes sobre ellas.

Además también se utilizarán documentos gráficos como soporte, para una mayor comprensión y sobre los que se expondrán las correspondientes explicaciones.

### **Desarrollo**

Esta parte del proyecto se centra en explicar, con datos más técnicos, el proceso de desarrollo de la aplicación. Se expondrá, de la manera más clara posible, el funcionamiento de cada elemento que conforma el proyecto, el uso de funciones, control de errores, diseño adaptado...

Este apartado tendrá un gran soporte gráfico para una mayor facilidad de comprensión del lector.

### **Conclusión**

Finalmente se redacta una conclusión, en la que se describen opiniones y sensaciones obtenidas a lo largo de la elaboración del proyecto, si se ha conseguido el objetivo propuesto y una valoración del resultado final obtenido.

## 2. Análisis o definición del problema

El problema que se plantea en este proyecto y que se pretende resolver, es la recogida de datos acerca del estado del agua de la red de suministros para llevar un control de la calidad del agua destinada al consumo.

La problemática principal no reside en el hecho de recoger los datos, ya que actualmente los datos son recogidos y recopilados. El problema que cabe resolver es el modo que se emplea para recoger y almacenar dichos datos.

A continuación se describe el proceso que se sigue actualmente para realizar esta tarea y posteriormente indicaré la solución que se pretende implantar con este proyecto.

### 2.1 Proceso actual

El centro de salud de Alcoy es la institución encargada de la recogida, análisis y almacenamiento de datos que revelan el estado del agua destinada al consumo humano.

Este centro cuenta con dos tipos de personas/usuarios para desarrollar esta tarea.

- Gestor: Persona encargada de la toma de muestras en distintos puntos de la ciudad.
- Inspector: Persona que realiza la misma función que el gestor, pero a modo de revisión, para controlar que el gestor realiza correctamente su trabajo.

El gestor tiene asignadas zonas de la ciudad dónde hay puntos donde tomar las muestras. Estas asignaciones las realizan los encargados del centro de salud. El gestor acude a los puntos asignados, toma muestras de agua, las analiza y toma anotaciones en papel de los resultados. Posteriormente estos datos se envían al centro de salud por medio de fax para su almacenamiento.

Cada cierto tiempo, un inspector puede tomar muestras para comprobar que los datos recogidos por el gestor son correctos.

El principal problema que existe es el modo de recolectar datos, ya que al tomar anotaciones en papel y ser enviadas por fax, el proceso de actualización y mantenimiento de los datos tiene un valor en tiempo y en uso de papel demasiado alto.

## 2.2 Solución propuesta

La solución que se propone en este proyecto para hacer frente al problema existente, es la creación de una aplicación web para conseguir automatizar el proceso en la medida de lo posible. El uso de esta aplicación, ofrecerá las siguientes ventajas:

- Reducción de uso de papel.
- Reducción del tiempo empleado en la ejecución del proceso.
- Mejor aprovechamiento del espacio físico (datos informatizados).
- Actualización inmediata de los datos.
- Búsqueda de datos más rápida y eficaz.

## 2.3 Requisitos

El paso fundamental para conocer los requisitos que debe cumplir la aplicación que se quiere desarrollar es hablar con las personas implicadas en el proceso, para que ofrezcan información lo más detallada posible de todos los aspectos a tener en cuenta.

### 2.3.1 Funcionalidades de usuario

En el sistema existen dos tipos de usuarios (inspectores y gestores). Los inspectores tienen un rol en el sistema que les permite realizar acciones administrativas que obviamente un usuario gestor no puede realizar. Además tiene acceso a una mayor cantidad de vistas de la aplicación.

En la siguiente tabla se muestra las funciones correspondientes a cada tipo de usuario y sobre que tablas puede realizar dichas funciones.

<b>Usuario</b>	<b>Función</b>	<b>Tabla</b>
Inspector	Insertar Actualizar Listar Borrar	Municipios Muestras Usuarios Tomas Zonas
Gestor	Insertar	Muestras

### 2.3.2 Requisitos indispensables

A continuación se describen, de manera más detallada, los requisitos a cubrir en cuanto a la funcionalidad de cada usuario.

- **Inspector/Admin**

El usuario inspector es el administrador del sistema, por lo tanto, tiene todos los permisos sobre los datos de la aplicación. Las principales funciones que debe poder desempeñar este usuario son:

- Gestionar (dar de alta/baja/listar) los municipios de los que es responsable un centro de salud.
- Gestionar (dar de alta/baja/listar) las zonas de abastecimiento incluidas en cada municipio.
- Gestionar (dar de alta/baja/listar) los puntos de toma de muestra incluidos en cada zona de abastecimiento.
- Gestionar (dar de alta/baja/listar) gestores de agua, pudiendo asignarlos a distintas zonas de abastecimiento.
- Gestionar (dar de alta/baja/listar) las muestras de agua tomadas por los gestores en cada punto de toma de muestra.
- Gestionar (dar de alta/baja/listar) los agentes de control que pertenecen a un centro de salud.
- Gestionar (dar de alta/baja/listar) las muestras de control de agua tomadas por los agentes de control en los punto de toma de muestra.

- **Funcionalidades para el gestor**

El usuario “gestor” es un usuario sin privilegios, que únicamente se le permite insertar datos en la tabla de muestras. Este usuario se crea para que sea utilizado por los técnicos que toman las muestras en las diferentes tomas de la ciudad. Las funciones que debe cumplir este usuario son:

- Dar de alta nuevas muestras

### 2.3.3 Requisitos deseables

Los requisitos mostrados anteriormente, son aquellos que se necesitan cubrir obligatoriamente, puesto que dan solución a la esencia del problema.

Como requisitos deseables, para dar una mayor funcionalidad al sistema estarían los siguientes:

- Detectar incumplimientos por parte de los gestores en cuanto a:
  - la periodicidad de toma de muestras
  - el nivel de los parámetros de calidad del agua.
- Automatizar la notificación de incumplimientos vía e-mail, tanto a los responsables del incumplimiento (los gestores), como a los responsables del control (el centro de salud).
- Contar con mecanismos de importación/exportación de datos, a partir de documentos Excel, PDF.
- Visualización de gráficas que muestren el nivel de cloro de distintos puntos de manera cronológica, para poder evaluar su evolución.
- Contar con un panel de control que mostrara las incidencias más relevantes, y el estado actual del sistema.

## 2.4 Planificación

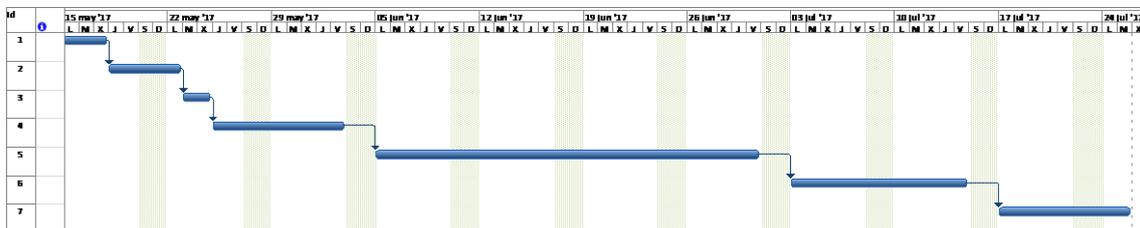
La planificación es un punto a tener muy en cuenta a la hora de desarrollar un proyecto, ya que es una buena forma de organizar las distintas tareas a desempeñar.

Para este proyecto, antes de iniciar el proceso de desarrollo, se ha creado un pequeño calendario con todas las tareas que se tienen que realizar, acompañadas del tiempo estimado que se va a invertir en cada una de ellas, para obtener de forma aproximada, el tiempo total que se va a utilizar para crear el proyecto completo.

En este caso se puede observar un diagrama de Gantt con la distribución de las distintas tareas. Según la planificación del calendario, el proyecto debe estar completado el 25 de julio aproximadamente, sin embargo y como suele pasar en todos los proyectos, el tiempo estimado no suele corresponder con la realidad, ya que se tiene que hacer frente a imprevistos y problemas.

Cabe destacar que la estimación de tiempo total se ha cumplido casi con exactitud. Únicamente ha variado el tiempo invertido en las tareas.

Modo de	Nombre de tarea	Duración	Comienzo	Fin
	Recopilación de información	3 días	lun 15/05/17	mié 17/05/17
	Estudio del problema	3 días	jue 18/05/17	lun 22/05/17
	Análisis y elección de tecnologías y herramientas	2 días	mar 23/05/17	mié 24/05/17
	Desarrollo capa de datos	7 días	jue 25/05/17	vie 02/06/17
	Desarrollo capa de negocio	20 días	lun 05/06/17	vie 30/06/17
	Desarrollo capa de presentación	10 días	lun 03/07/17	vie 14/07/17
	Memoria	7 días	lun 17/07/17	mar 25/07/17



### 3. Solución del problema

#### 3.1 Arquitectura de la aplicación

La aplicación que se pretende desarrollar en este proyecto, se basa en la división de las funcionalidades en tres capas (Presentación – Negocio – Datos) aunque se podría considerar de dos capas ya que la capa de negocio y de datos se ejecutan en el servidor y la de presentación en el cliente. Para desarrollarla se utiliza el famoso patrón MVC (Model-View-Controller) o en español, modelo-vista-controlador, el cual tiene la siguiente estructura:



Imagen: <https://sites.google.com/site/aunaris2/programacion/modelo-vista---controlador>

Este patrón basa su idea en la separación de código según su funcionalidad. A continuación se describe cada una de las partes que conforman el patrón MVC.

- **Modelo**

En el modelo, se desarrolla el código que representa la estructura de los datos, por ejemplo, los campos de una tabla existente en una base de datos. Esta estructura nos servirá para crear funciones con el fin de manipular los datos.

- **Vista**

Es la parte visual de la aplicación. Es la parte accesible por el usuario y en ella se muestran los datos de la base de datos. Mediante la intervención del usuario con las diferentes vistas de la aplicación, se podrán modificar/actualizar/insertar/borrar los datos.

Las vistas necesitan la funcionalidad del modelo y el controlador para poder ser útiles.

- **Controlador**

En esta parte de la aplicación se desarrollan las funciones de inserción/modificación/eliminación de datos. La vista llama al controlador y este se encarga de realizar la tarea solicitada en la vista, conjuntamente con el modelo, que es donde está la estructura de los datos a los que queremos acceder.

Este sistema que se pretende desarrollar, cumple con las necesidades de un sistema CRUD (Create-Read-Update-Delete) o en español (Crear-Leer-Actualizar-Borrar).

Un sistema de este tipo nos permite hacer permanentes los datos en nuestra base de datos y manipularlos según las necesidades a satisfacer en cada momento.

### 3.2 Tecnologías utilizadas

Cuando se pretende desarrollar un proyecto de esta envergadura, se necesita tener conocimientos de distintas tecnologías e intentar apoyarse en recursos que faciliten la tarea, ya que normalmente, proyectos de este tipo suelen ser encargados a un conjunto de programadores que trabajan de forma colaborativa, no a una sola persona.

A continuación se describen todas las tecnologías utilizadas para el desarrollo de este proyecto.

### 3.2.1 Lenguajes de programación

Puesto que el proyecto está basado en tecnología web, se procede a describir todos los lenguajes utilizados que están enfocados a esta tecnología.

- **HTML**

Conocido por sus siglas como HyperText Markup Language es un lenguaje standard para la elaboración de páginas web.

Nos permite generar la estructura de la web, creando secciones, donde posteriormente se muestra la información.

Como es un estándar, se va a poder utilizar prácticamente en cualquier plataforma. Este lenguaje no va a permitir introducir colores, ni crear una web visualmente atractiva para el usuario, para ello se necesita hacer uso de otros lenguajes como CSS.

En este proyecto se ha utilizado la versión 5 de HTML, que es la más reciente.



Imagen: [https://www.w3.org/html/logo/downloads/HTML5\\_Logo\\_512.png](https://www.w3.org/html/logo/downloads/HTML5_Logo_512.png)

- **CSS**

*Cascading Stylesheets* u *Hojas de estilo en cascada* es un lenguaje enfocado a la parte visual de la web, que trabaja conjuntamente con HTML para el desarrollo de una web atractiva visualmente para el usuario.

Este lenguaje se basa en la creación de hojas de estilos donde se indican aspectos de la web como colores, posicionamiento de elementos, efectos...

Para que html pueda trabajar conjuntamente con CSS, se debe añadir un link a la página HTML. Por ejemplo:

```
<link rel=stylesheet href="archivo CSS"/>
```

Con esta sentencia, HTML implementa las características descritas en el archivo css a los elementos con el mismo id o el mismo tipo.

Para este proyecto se ha utilizado la última versión (versión 3).



Imagen: <https://rolandocaldas.com/wp-content/uploads/2013/05/css3.png>

- **JAVASCRIPT**

Conocido por sus siglas JS, es un lenguaje de programación orientado a web que basa su funcionamiento en el uso de scripts.

Un script es una secuencia de código que se ejecuta en un momento determinado.

Para ejecutar scripts Javascript en documentos HTML, se necesita insertar la referencia al archivo .js en el código html de la siguiente manera:

```
<script type="text/javascript" src="ruta del archivo"></script>.
```

Este lenguaje suele ir enfocado a la parte del cliente, sobretodo se utiliza para la modificación de imágenes, validación de datos introducidos en un formulario, cambios dinámicos en el contenido...

Aunque también se puede ver implementado en la parte del servidor. Un ejemplo sería Node.js.

Actualmente es uno de los lenguajes de programación más utilizados según la lista publicada por Bussines2community.

(<http://www.business2community.com/tech-gadgets/top-20-popular-programming-languages-2017-01791470#tiV042Lw4evRZ2ky.97>)

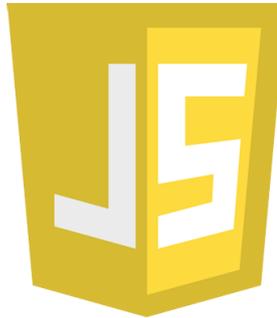


Imagen: [https://www.codementor.io/assets/page\\_img/learn-javascript.png](https://www.codementor.io/assets/page_img/learn-javascript.png)

- **PHP**

Este es otro de los lenguajes más utilizados para desarrollo web en la actualidad, como podemos observar en la lista facilitada anteriormente.

Es un lenguaje de programación enfocado a la parte del servidor y de código abierto. Sus siglas significan *PHP Hipertext Preprocessor* y su funcionamiento de PHP es el siguiente:

Desde la parte del cliente (navegador) se envía una petición al servidor para obtener unos datos determinados. El servidor ejecuta el código PHP necesario, y crea una respuesta HTML que es devuelta al cliente con la información que había sido solicitada para que sea visualizada en el navegador.

PHP trabaja conjuntamente con bases de datos. Es un lenguaje de programación en el que es fácil iniciarse, en comparación con otros, y que ofrece mucho potencial.

En este proyecto se ha utilizado la versión 5.6.

Toda la información referente a este lenguaje de programación se puede consultar en su página oficial <http://php.net/>



Imagen: <https://upload.wikimedia.org/wikipedia/commons/thumb/2/27/PHP-logo.svg/1200px-PHP-logo.svg.png>

- **SQL**

Por sus siglas *Structured Query Language* es un lenguaje de programación desarrollado con el fin de crear, manipular y borrar bases de datos. Utiliza queries (consultas) para tratar con la base de datos.

Estas queries pueden ser de cuatro tipos principalmente:

- Create: Crear tablas y base de datos.
- Insert: Insertar datos en la base de datos.
- Update: Actualizar datos ya insertados.
- Delete: Borrar datos.

Para la utilización de este lenguaje, se ha escogido Mysql, que es un sistema de gestión de bases de datos relacionales.

En combinación con PHP, permite enviar datos al navegador con el fin de mostrárselos al cliente.



Imagen: <http://4.bp.blogspot.com/-eHpEiIM6Nss/VRs-EqQizYI/AAAAAAATSo/tpEFUn3QrZY/s1600/SQL-Server.png>



Imagen: <https://upload.wikimedia.org/wikipedia/en/thumb/6/62/MySQL.svg/1200px-MySQL.svg.png>

### 3.2.2 Frameworks / Herramientas

- **Laravel**

Es un framework PHP creado en 2011 con el fin de ayudar a los desarrolladores en la creación de aplicaciones web. Sigue el patrón MVC aunque posee una estructura más compleja, ya que alberga muchos más archivos que ofrecen una mayor funcionalidad al proyecto.

Es uno de los frameworks PHP más utilizados del mercado como podemos ver en diferentes listas publicadas en Internet.

<https://openwebinars.net/blog/los-5-frameworks-php-mas-usados/>

En una de ellas aparece un gráfico explicativo del porcentaje de uso de los frameworks PHP más relevantes y se puede observar que Laravel tiene una posición dominante junto a Symfony.

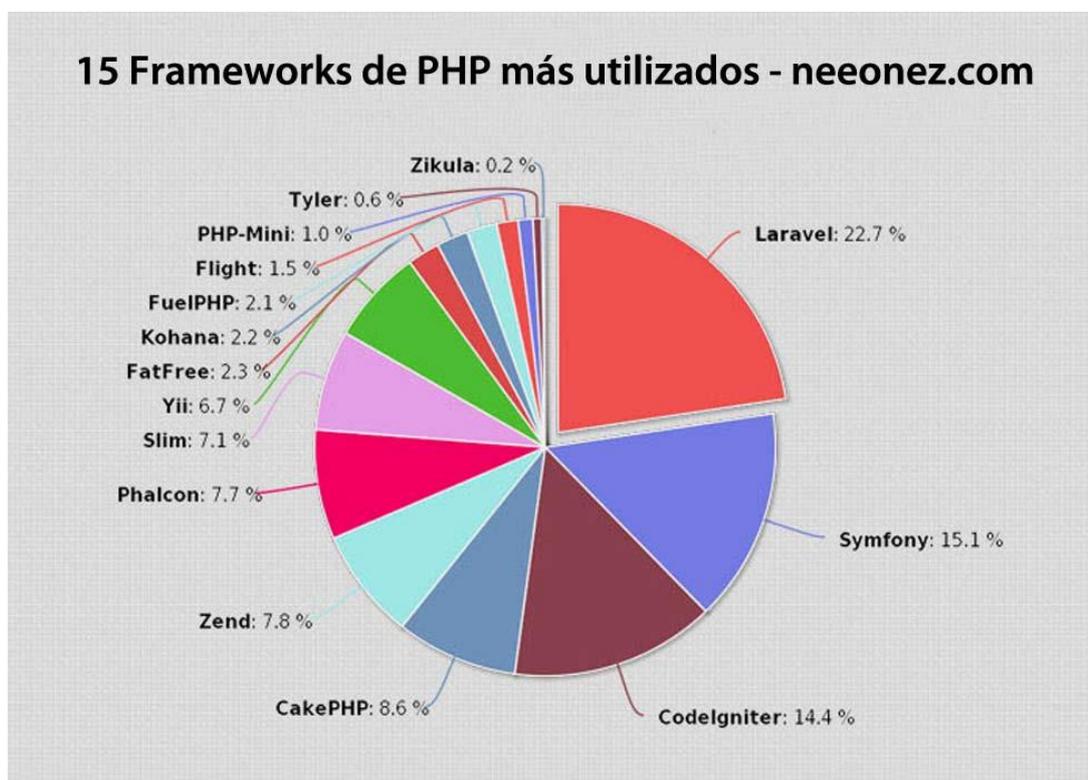
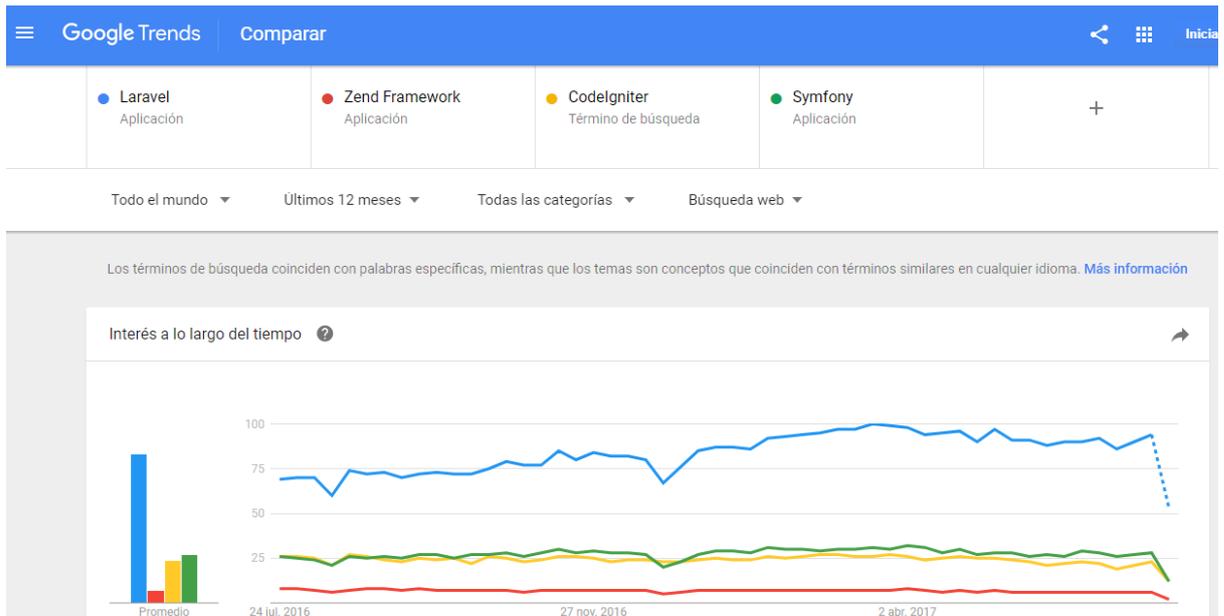


Imagen: <http://www.neeonez.com/wp-content/uploads/2016/07/grafico-frameworks-php.jpg>

Se pueden obtener más datos de este tipo realizando una consulta en Google Trends, que es una página que estudia los conceptos más buscados y utilizados. En este caso realizamos la búsqueda siguiente.



Las dos principales opciones a la hora de desarrollar este proyecto son Symfony y Laravel. Tras estudiar las ventajas y desventajas de cada uno, se ha optado por utilizar Laravel, puesto que ofrece un rendimiento muy bueno, por la gran comunidad de desarrolladores que le da soporte y porque la línea de aprendizaje es más asequible que ofrecida por Symfony.

Estas tres características provocan que finalmente sea Laravel el framework utilizado. En este proyecto se ha hecho uso de la versión 5.4 de Laravel.



Imagen: <https://manuais.iessanclemente.net/images/b/be/Laravel-5.png>

- **Vue.js**

Este es otro de los Frameworks que se han utilizado en este proyecto, para utilizar como soporte en la generación de código JavaScript. Vue permite dar dinamismo al proyecto, lo cual es una característica fundamental para conseguir una web atractiva. Ofrece un gran rendimiento y es muy fácil de utilizar.



Imagen: <http://victorroblesweb.es/wp-content/uploads/2017/03/vuejs2-victorroblesweb.jpg>

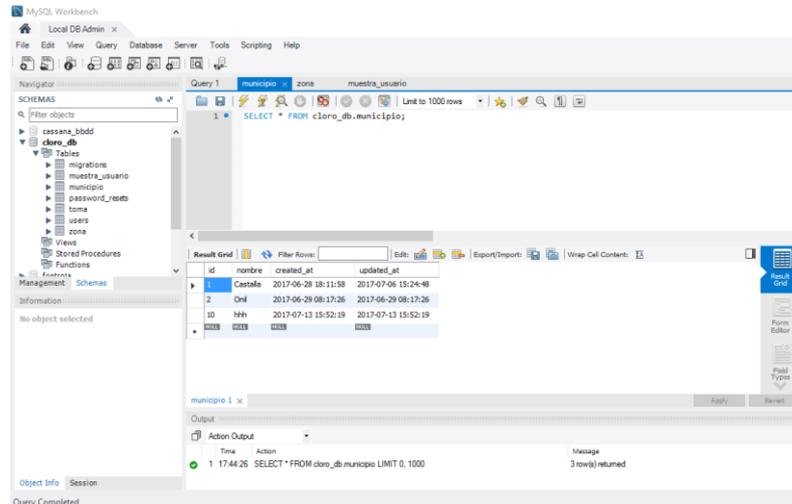
Se utiliza en combinación con Laravel para conseguir una funcionalidad completa en el proyecto.

Existen otros frameworks de este estilo como pueden ser Angular, ReactJS o Ember. Aunque Angular es el framework más utilizado a nivel global, Vue.js cuenta con una gran comunidad de desarrolladores, es eficiente y da un resultado similar al que se puede conseguir con Angular, de un modo más sencillo.

- **Mysql Workbench**

Es una herramienta que permite gestionar bases de datos. Mediante esta aplicación, se ha creado la base de datos necesaria para el proyecto, junto con el diagrama de su estructura.

Permite ejecutar scripts con extensión sql para ejecutar acciones sobre la base de datos. El entorno de desarrollo tiene este aspecto.



Es una de las herramientas más utilizadas para la gestión de base de datos en la actualidad.



Imagen: <http://ignaciojusticia.es/wp-content/uploads/2016/05/logo-mysql-workbench.png>

- **BOOTSTRAP**

Es un framework creado en 2011 que tiene como finalidad el desarrollo del diseño de sitios web. Se centra en la maquetación de la aplicación web (front-end), mediante la utilización de tres lenguajes de programación (HTML, CSS, JAVASCRIPT). Su característica más atractiva, es que permite crear sitios web responsive, es decir, proyectos que se adaptan al dispositivo desde el cual se visualizan (PC, móvil, tablet).

Esto se consigue introduciendo en las etiquetas HTML que se refieren a contenedores (div) ciertas sentencias que indican el tamaño del contenedor en base al dispositivo.

Bootstrap se basa en un sistema en el cual, divide el contenido HTML en una especie de rejilla (columnas). Se recomienda que el ancho total del contenido sea de 12 columnas, por lo tanto se debe asignar un total de 12 columnas a los distintos contenedores de nuestro HTML para que se

adapte. A continuación se muestra un ejemplo básico de funcionamiento de Bootstrap para la creación de una web Responsive sacado de la siguiente web:

[https://ajgallego.gitbooks.io/bootstrap-3/content/capitulo\\_rejilla.html](https://ajgallego.gitbooks.io/bootstrap-3/content/capitulo_rejilla.html)

Aquí se puede ver una tabla con las referencias de las etiquetas a utilizar y los tamaños según los diferentes dispositivos.

Pantalla	Prefijo de la clase	Ancho del contenedor
Tamaño extra pequeño Teléfonos (<768px)	.col-xs-	Ninguno (automático)
Tamaño pequeño Tablets (≥768px)	.col-sm-	750px
Tamaño medio Escritorios (≥992px)	.col-md-	970px
Tamaño grande Escritorios (≥1200px)	.col-lg-	1170px

A nivel de código se muestra de la siguiente manera:

```
<!-- En pantallas pequeñas aparecerá una columna que ocupará todo el ancho
y otra que ocupará la mitad de la pantalla -->
<div class="row">
  <div class="col-xs-12 col-md-8">.col-xs-12 .col-md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- En pantallas pantallas se indica que ocupe cada columna la mitad
del ancho disponible -->
<div class="row">
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- Como no se indica el tamaño para pantallas grandes las columnas
siempre ocuparán el 50% -->
<div class="row">
  <div class="col-xs-6">.col-xs-6</div>
  <div class="col-xs-6">.col-xs-6</div>
</div>
```

Y el resultado es el siguiente:

Para pantallas grandes y medianas (escritorio y tablets)

.col-xs-12 .col-md-8		.col-xs-6 .col-md-4	
.col-xs-6 .col-md-4	.col-xs-6 .col-md-4	.col-xs-6 .col-md-4	
.col-xs-6		.col-xs-6	

Para pantallas pequeñas (móviles)

.col-xs-12 .col-md-8	
.col-xs-6 .col-md-4	
.col-xs-6 .col-md-4	.col-xs-6 .col-md-4
.col-xs-6 .col-md-4	
.col-xs-6	.col-xs-6

Esto permite un mayor uso de la aplicación, ya que se puede acceder a ella con un diseño adaptado al dispositivo que se esté utilizando.



Imagen: <http://www.cloudesign.com/images/bootstrap1.png>

- **XAMPP**

Se ha hecho uso de esta herramienta de software libre que permite utilizar un servidor virtual con el fin de poder probar la aplicación web que se está desarrollando. El panel principal de esta herramienta es el siguiente.

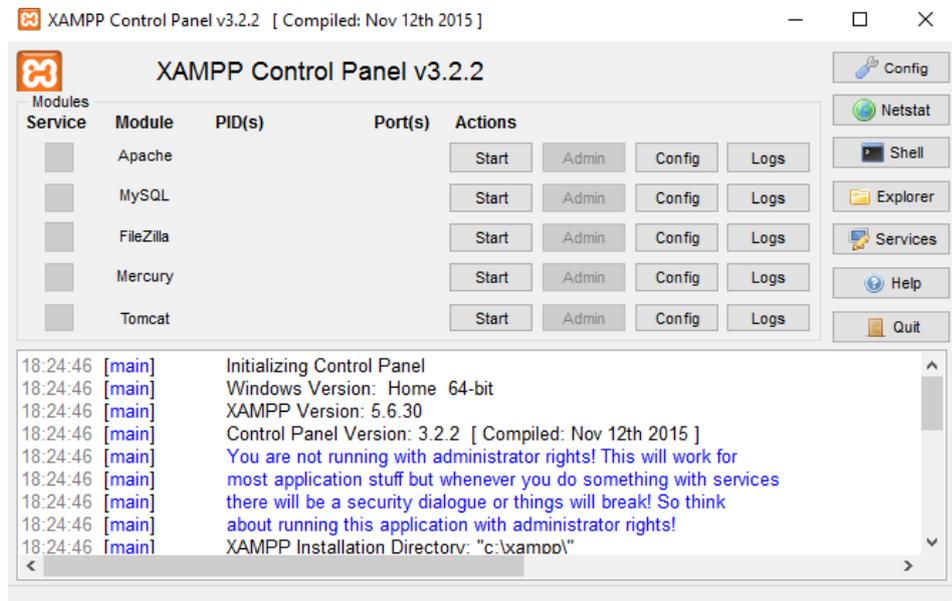


Imagen:

[https://upload.wikimedia.org/wikipedia/commons/thumb/0/03/Xampp\\_logo.svg/1280px-Xampp\\_logo.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/0/03/Xampp_logo.svg/1280px-Xampp_logo.svg.png)

- **CHART.JS:**

Esta es una herramienta de código abierto que está especializada en la generación de gráficos a partir de unos datos concretos recogidos de una base de datos. Está basado en el lenguaje Javascript y ofrece una gran variedad de tipos de gráficas tal y como se puede observar en su página web oficial (<http://www.chartjs.org/>). Además en ella se puede consultar infinidad de información acerca de cómo crear dichas gráficas.

Tiene una gran comunidad que le da soporte y ofrece unos resultados espectaculares.

Gracias a la utilización de Javascript para la generación de las gráficas, éstas están cargadas de efectos que les dan una apariencia muy dinámica.



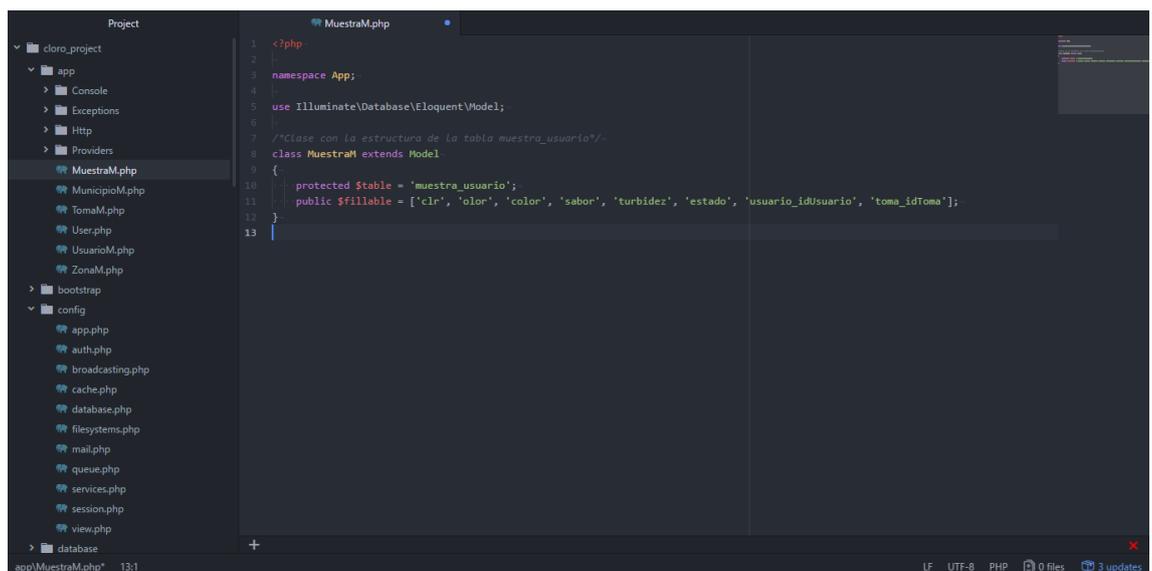
# Chart.js

Simple yet flexible JavaScript charting for designers & developers

Imagen: [www.chartjs.org](http://www.chartjs.org)

- **ATOM**

Editor de código creado por GitHub Inc en el año 2014. Está basado en Node.js y permite añadir plugins para mejorar su funcionalidad. El entorno de trabajo de este editor es el siguiente:



A la izquierda se puede ver la estructura del proyecto con todos los ficheros que lo conforman. A la derecha se visualiza un pequeño resumen a modo de mapa del código escrito (es un plugin para tomar una referencia de la parte de código que se está modificando), y en el centro, la zona dónde se edita el código.

La página oficial de este proyecto es <https://atom.io/> y está diseñado para ejecutarse en cualquier plataforma (Windows, iOS...).



Imagen: <https://atom.io/>

### 3.3 Capas

Este proyecto está basado en una estructura en capas, las cuales trabajan de forma conjunta para la resolución de un problema. En este caso, está formado por tres capas (Presentación, Negocio y Datos) aunque realmente se puede agrupar en dos, ya que las capas de negocio y de datos están más enfocadas a la manipulación de datos, y la capa de presentación está destinada a ofrecer una herramienta visual al usuario para interactuar con las demás capas. Un ejemplo gráfico de la estructura sería el siguiente:

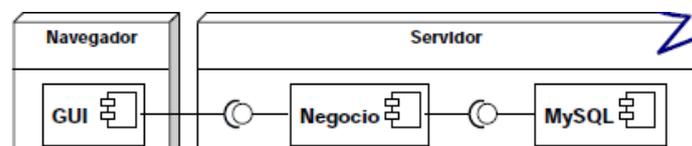


Imagen: Diapositivas Integración Aplicaciones

En la imagen los nombres son distintos pero la relación entre estos nombres y los expuestos en la descripción es la siguiente:

GUI → Presentación

Negocio → Negocio

MySQL → Datos

También se puede observar que la capa de presentación se ejecuta en la parte del cliente (Navegador) y las capas de negocio y datos en la parte del servidor.



Imagen: [https://i.ytimg.com/vi/M\\_M8T8G4Og8/maxresdefault.jpg](https://i.ytimg.com/vi/M_M8T8G4Og8/maxresdefault.jpg)

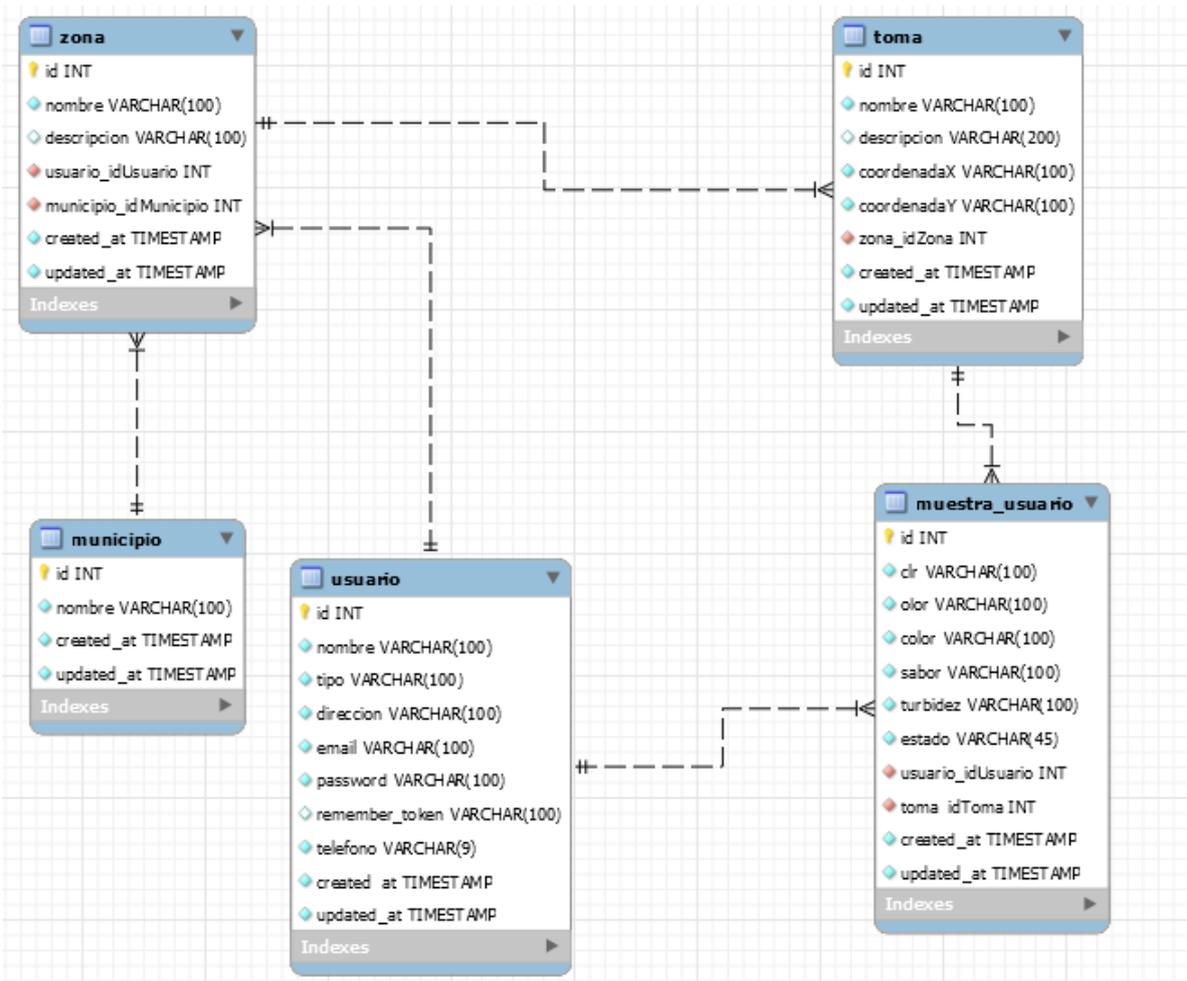
A continuación se van a describir las distintas capas que conforman este proyecto para dar una idea más específica acerca de su funcionamiento.

### 3.3.1 Capa de datos

En esta capa se encuentra la base de datos, donde se almacena la información relevante de la aplicación y una de las partes fundamentales del proyecto. Trabaja conjuntamente con la capa de negocio, que es la encargada de facilitarle los datos a almacenar, y de solicitarle datos que quiere recuperar, por lo tanto la comunicación es bidireccional.

La base de datos ha sido desarrollada con la herramienta Mysql Workbench y tomando como referencia los requisitos de la aplicación que se pretenden cubrir, se ha realizado un análisis de las tablas que son necesarias y las relaciones entre ellas para que se ajuste a los requisitos del proyecto.

Una vez realizado dicho análisis, el diagrama de tablas resultante es el siguiente:



En el diagrama se pueden observar las distintas tablas que se necesitan en la base de datos. También se observa que las tablas están unidas por líneas.

Estas líneas significan que hay una relación existente entre las dos tablas, es decir, que una de las tablas tiene un campo que pertenece a la otra, es lo que se conoce como clave ajena.

Para conocer los tipos de relaciones entre tablas, se facilita la siguiente información:

Tablas	Relación
Municipio – Zona	1:N
Zona – Toma	1:N
Usuario – Zona	1:N
Usuario – Muestra_usuario	1:N
Toma – Muestra_usuario	1:N

Tal como aparece en la tabla, todas las relaciones son 1:N. Para explicar esta relación se toma como referencia la primera fila de la tabla. En este caso, la relación tiene como significado, que un municipio (1) puede tener muchas zonas (N), y que una zona, solo puede pertenecer a un municipio.

Existen más tipos de relaciones (1:1, N:N...) pero en este caso, da la casualidad de que todas las relaciones son iguales.

El primer campo de cada tabla, hace referencia a la clave primaria (PK) que sirve como identificador único.

Los campos precedidos por un rombo de color rojo, son las claves ajenas que hacen referencia otras tablas (FK).

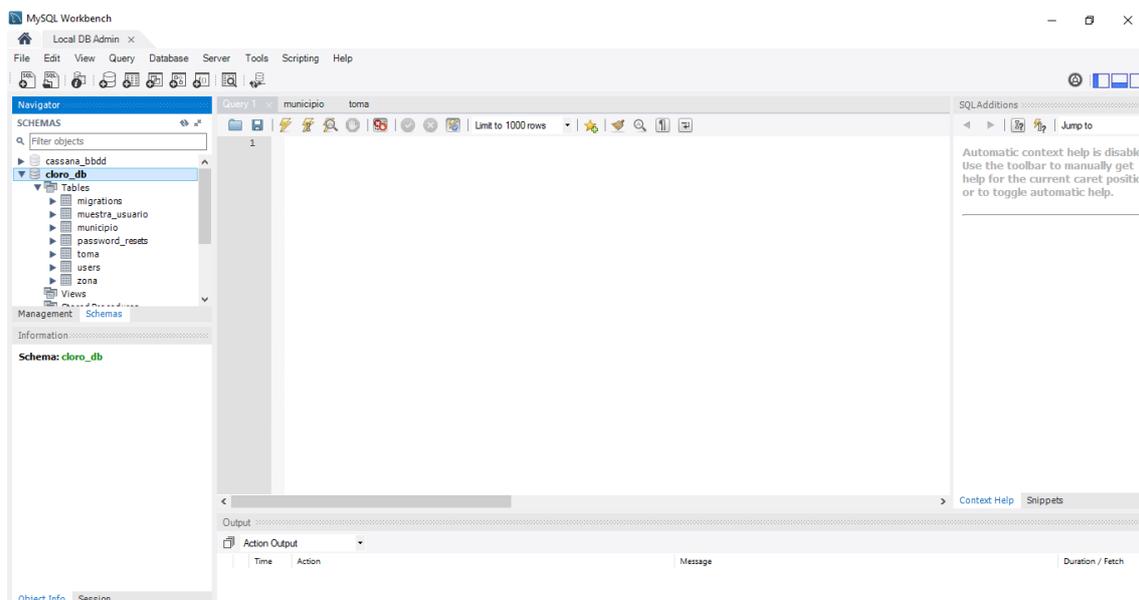
En cada tabla se han añadido dos campos (*created\_at* y *updated\_at*) que permiten almacenar la fecha y hora en que se ha creado y actualizado un registro.

Esto es solo un diagrama, que es muy útil para realizar pruebas de diseño de la base de datos.

Una vez el diagrama de base de datos está claro, se procede a su creación.

Para ello se realiza una exportación de los datos del diagrama a un fichero .sql, en el que se aplican las funciones CREATE DATABASE, CREATE TABLE...

Posteriormente cargando el fichero resultante y ejecutándolo, se crea la base de datos ("cloro\_db") como podemos ver en las imágenes.



Se puede contemplar que la base de datos ha sido creada, con sus tablas y sus campos correspondientes.

Posteriormente, con la creación del login del proyecto con el framework Laravel, se crea una tabla adicional en la base de datos denominada “password\_resets” que sirve para la función de recuperar la contraseña olvidada por el usuario.

En esta capa, también se cuenta el fichero (php) creado en Laravel, para mapear la base de datos. Es conocido como modelo y contiene la estructura de la tabla. De este modo, desde el controlador podremos interactuar con la base de datos basándonos en este modelo. Un ejemplo de fichero de Modelo, se muestra a continuación, el cual está mapeando la tabla Muestra, para que el controlador manipule la base de datos a través de él.

```
MuestraM.php
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;

/*Clase con la estructura de la tabla muestra_usuario*/
class MuestraM extends Model
{
    protected $table = 'muestra_usuario';
    public $fillable = ['clr', 'olor', 'color', 'sabor', 'turbidez', 'estado', 'usuario_idUsuario', 'toma_idToma'];
}
```

La función fundamental de esta capa es la de almacenar los datos que son introducidos por el usuario a través de la capa de presentación (pasando por la capa de negocio) y la de ofrecer los datos que se le soliciten (utilizando para ello consultas).

### 3.3.2 Capa de negocio

También conocida como capa de lógica, está situada entre la capa de presentación y la capa de datos. Sirve como intermediaria entre estas dos capas, de modo que consigue que se comuniquen correctamente entre ellas. Su función dentro del proceso de comunicación entre capas es el siguiente:

- Recoge los datos enviados desde la capa de presentación y ejecutar las funciones correspondientes para interpretar dicha información.
- Una vez la información es interpretada, se comunica con la capa de datos, indicándole qué datos precisa recoger o introducir.
- Recoge los datos facilitados por la capa de datos, los vuelve a tratar y los envía a la capa de presentación con un formato adecuado para que lo visualice el usuario.

Es un fichero php el cual contiene métodos para interactuar con la base de datos. En este caso tiene métodos para insertar, crear, modificar y eliminar

datos de la base de datos. Este controlador se comunica con el modelo mencionado anteriormente para manipular los datos de la base de datos y con las vistas para ofrecerle los datos solicitados por el usuario.

Una vez el controlador, a través del modelo ha conseguido los datos de la base de datos, procede a enviárselos a la capa de presentación (ejecutada en el navegador) en un formato fácil de mostrar. En este caso, le facilita la información en formato JSON, que es un lenguaje para el intercambio de datos y que tiene la siguiente estructura.

```
{ "pagination": { "total": 1, "per_page": 5, "current_page": 1, "last_page": 1, "from": 1, "to": 1 }, "data": { "current_page": 1, "data": [ { "id": 4, "clr": "7", "olor": "Apto", "color": "Apto", "sabor": "No Apto", "turbidez": "No Apto", "usuario_idUsuario": 1, "toma_idToma": 2, "estado": "Realizada", "created_at": "2017-07-25 14:59:01", "updated_at": "2017-07-25 17:26:16" } ], "from": 1, "last_page": 1, "next_page_url": null, "path": "http://localhost:8080/muestras", "per_page": 5, "prev_page_url": null, "to": 1, "total": 1 }
```

En este fragmento de código va contenida toda la información solicitada, solo es necesario que la capa de presentación la trate para que se muestre correctamente.

Como el mismo nombre indica (Controlador), es el encargado de controlar todo el flujo de datos, y de proponer unas reglas que deben ser cumplidas para que las operaciones se ejecuten correctamente.

Para este proyecto se ha creado un controlador específico para cada tabla existente en la base de datos, de este modo se consigue tener las funciones de control separadas en distintos ficheros y es más fácil de localizar.

Además se cuenta con otros ficheros que hacen la función de controlador para toda la función de login, registro, recuperación de contraseña...

En el siguiente apartado se describe la función de la capa de presentación.

### 3.3.3 Capa de presentación

Esta capa se encarga de mostrar al usuario los datos de la aplicación y sirve para que el usuario que utiliza la aplicación pueda interactuar con los datos sin necesidad de conocer la lógica que hay detrás. Simplemente accede a una página donde se muestran datos y puede solicitar/manipular otros. Es la parte más visual del proyecto y se crea con el fin de mostrársela al usuario final.

La capa de presentación está formada por ficheros que están cargados con código html, css, javascript, que son tres lenguajes de programación dedicados a diseño visual (front-end).

Como se ha podido observar anteriormente, la capa de negocio envía los datos en formato JSON que es fácil de transferir e interpretar, pero este formato no es adecuado para que el usuario final lo visualice sin un diseño adecuado. La capa de presentación permite ofrecer dicho diseño atractivo al usuario para que interprete los datos de un modo más cómodo. Los datos en JSON, se convierten en lo siguiente:

Muestras

+ Crear muestra

Pendientes / Realizadas

Cloro	Olor	Color	Sabor	Turbidez	Estado	Usuario	Toma		
7	Apto	Apto	No Apto	No Apto	Realizada	dddddd	2	Editar	Borrar

La capa de presentación es la única zona accesible directamente por el usuario que utiliza la aplicación.

Las funciones principales de la capa de presentación son las siguientes:

- Recoger información del usuario y enviarla al servidor (capa de negocio) para ser procesada.
- Recibir los datos ya procesados de la capa de negocio.
- Visualizar la información recibida con un formato sencillo y apropiado para que el usuario la interprete.

Con la utilización del modelo en capas, se consigue separar el proyecto en diferentes partes según su funcionalidad. Las principales ventajas de utilizar esta arquitectura son:

- Al tener el código separado por funciones, se consigue una mayor escalabilidad.
- Mayor facilidad para encontrar errores.
- Permite reutilizar componentes.
- Mantenimiento del código más sencillo.
- Permite agregar diferentes representaciones de los datos.
- Ofrecen un gran rendimiento.

## 4. Resultado o Tour por la aplicación

Una vez explicados los recursos utilizados para el desarrollo del proyecto, y la función de cada una de las partes del mismo, se procede a visualizar el resultado obtenido a través de las vistas de usuario.

Como se ha comentado al inicio del proyecto, existen dos tipos de usuarios que utilizarán dicha aplicación (admin/inspector y gestor), con lo cual, se han creado vistas que correspondan a cada uno de estos tipos de usuario, ya que un gestor no puede ver las mismas pantallas que un administrador.

Además se ha implementado lo que se conoce como protección de rutas. Este concepto consiste en proteger las vistas para evitar que un usuario no autorizado pueda acceder a zonas que no le corresponden aunque conozca la url de la vista en cuestión.

### 4.1 Login

La primera pantalla que se encuentra el usuario que accede a la aplicación web es el Login.

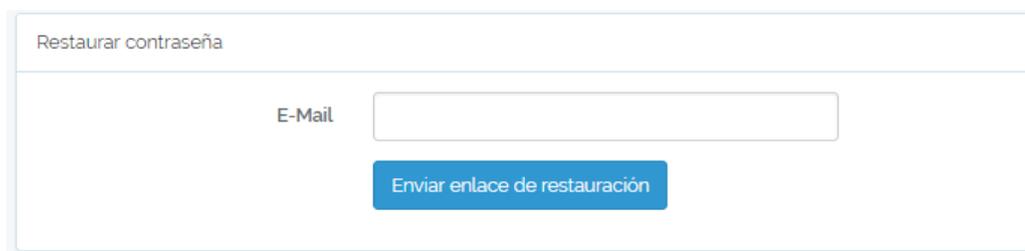


The screenshot shows a login form titled "Login". It contains two input fields: "E-Mail" and "Contraseña". Below the "Contraseña" field is a checkbox labeled "Recuérdame". At the bottom, there is a blue button labeled "Entrar" and a link labeled "He olvidado mi contraseña".

El usuario debe introducir sus credenciales (email y contraseña) para poder acceder al sistema. Para ello, uno de los administradores tendrá que haberlo dado de alta previamente. De este modo se consiguen dos cosas:

1. Evitar que cualquier usuario pueda registrarse en el sistema (esta función la realizan los administradores).
2. Controlar el tipo de usuario que se conecta, para así mostrarle la vista correspondiente.

El usuario también tiene a su disposición la función de recordar contraseña con el fin de ser utilizada en aquellos casos que no se recuerde la contraseña y no pueda acceder al sistema. En este caso, introduciría su email y se le enviaría un correo con un enlace para que la reestablezca.



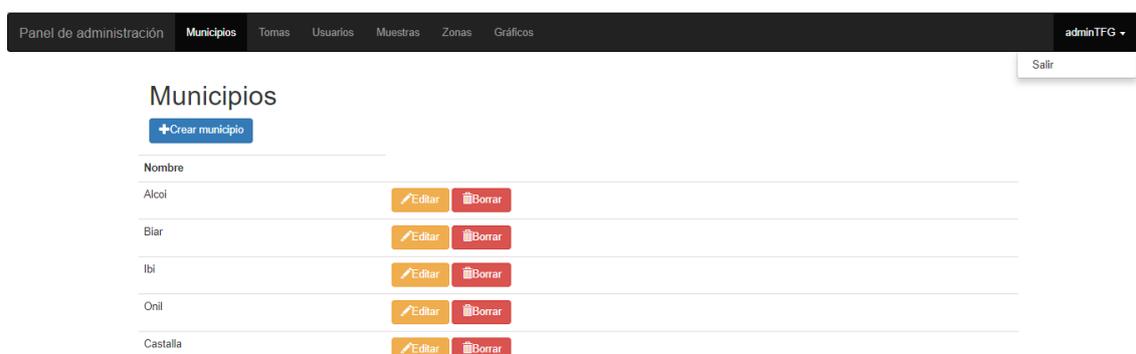
Restaurar contraseña

E-Mail

Enviar enlace de restauración

## 4.2 Panel de administración

Se propone como ejemplo que en primer lugar inicia sesión un administrador/inspector. En este caso se ha creado un usuario de prueba llamado “*adminTFG*” para realizar el tour por la parte de administración. Una vez se comprueba que las credenciales son correctas, accede al panel de administración que tiene el siguiente aspecto



Panel de administración | Municipios | Tomas | Usuarios | Muestras | Zonas | Gráficos | adminTFG

Salir

### Municipios

+ Crear municipio

Nombre	Editar	Borrar
Alcoi	Editar	Borrar
Biar	Editar	Borrar
Ibi	Editar	Borrar
Onil	Editar	Borrar
Castalla	Editar	Borrar

Se observa que el panel de administración está formado por un menú en la parte superior para acceder a la gestión de los distintos elementos, y en la parte derecha del mismo podemos ver el nombre del usuario que se ha registrado, equipado con un botón desplegable para poder salir de la aplicación. Cuando el usuario pulsa sobre el botón “*Salir*” se realiza el “logout” y se redirige a la ventana de “login”.

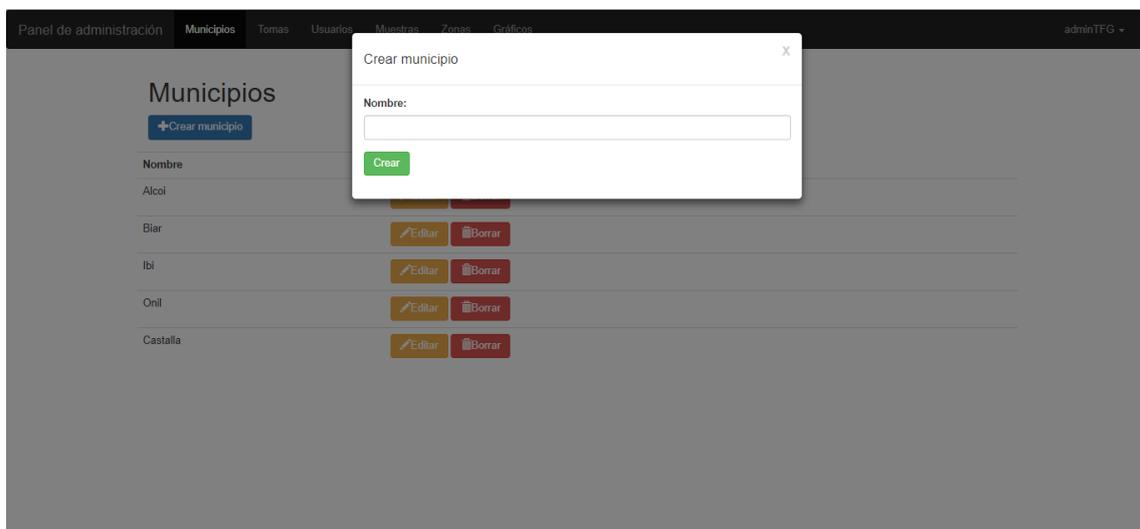
En la parte central del panel, se muestran los datos correspondientes a los municipios, ya que se está visualizando dicha sección, acompañados de tres botones que pueden ser utilizados por el usuario.

El botón azul sirve para crear un nuevo registro y guardarlo en la base de datos. El botón amarillo sirve para editar el registro correspondiente a la fila donde está ubicado, y el botón rojo sirve para eliminar el registro. Cada uno de estos botones llama a la función correspondiente definida en el controlador.

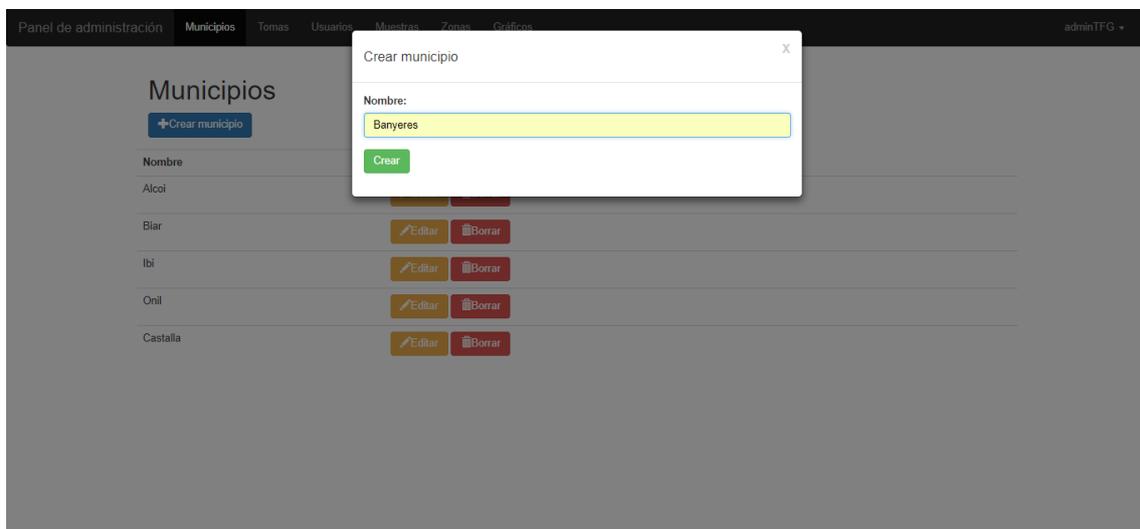
A continuación se muestra el proceso que se sigue a través de las pantallas al pulsar cada uno de los botones.

## **Botón crear**

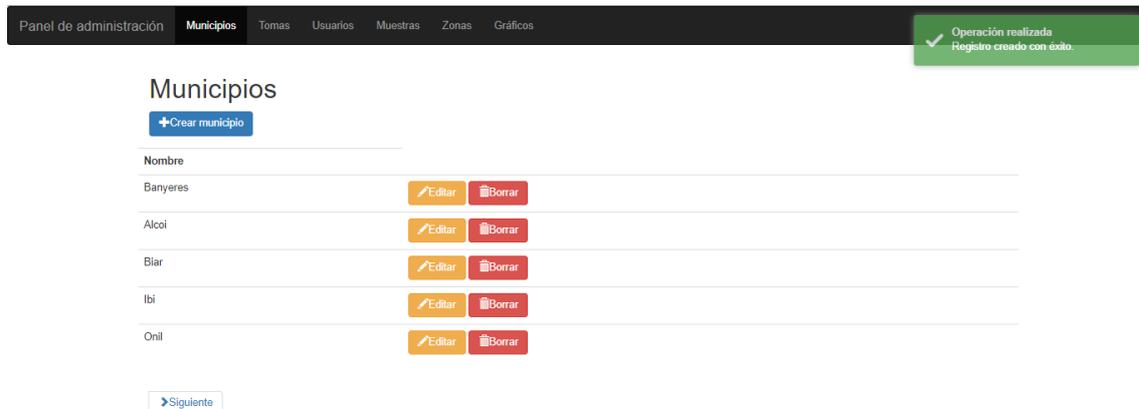
Al pulsar este botón, la aplicación muestra un formulario con un efecto dinámico para introducir los datos del nuevo registro.



Se introduce como prueba el municipio "Banyeres" que todavía no existe en la lista.



Una vez introducido el nombre del municipio y pulsar en el botón de crear, la aplicación vuelve a mostrar la lista de municipios actualizada con el nuevo registro.

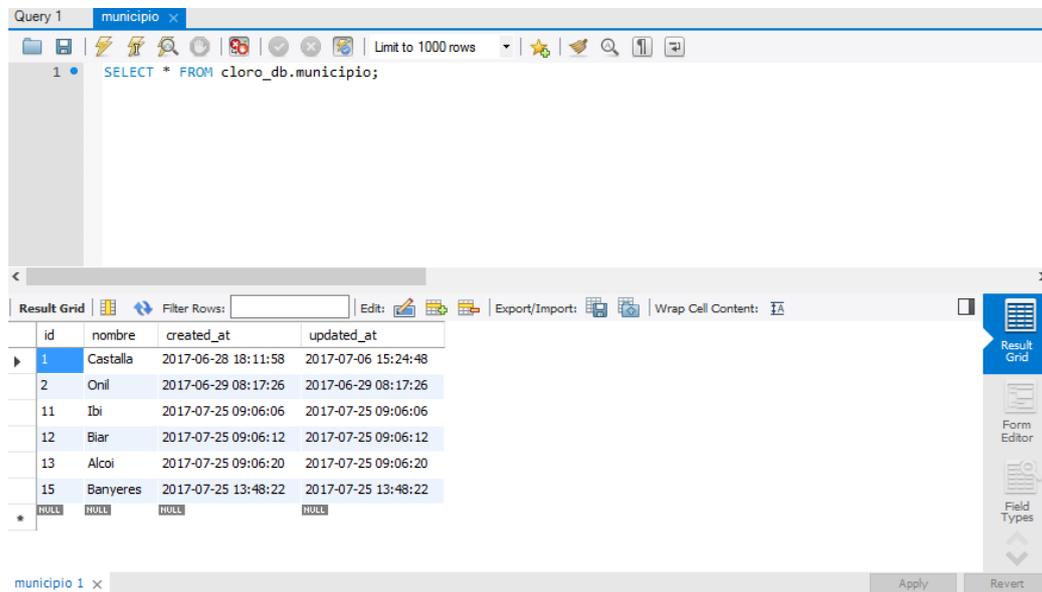


La lista está ordenada en orden ascendente tomando como referencia la fecha de creación del registro, por lo tanto, el registro que se acaba de crear aparece en primer lugar.

También se ha configurado un paginador en la parte inferior de la lista, para evitar que se muestren demasiados registros en una misma página. De este modo solo muestra un máximo de cinco registros por página. En caso de contar con más registros, aparece el botón "siguiente" para acceder a la siguiente página y ver los demás registros. También se puede apreciar que al crear el registro aparece una notificación de color verde en la esquina superior derecha indicando que la operación se ha ejecutado correctamente.



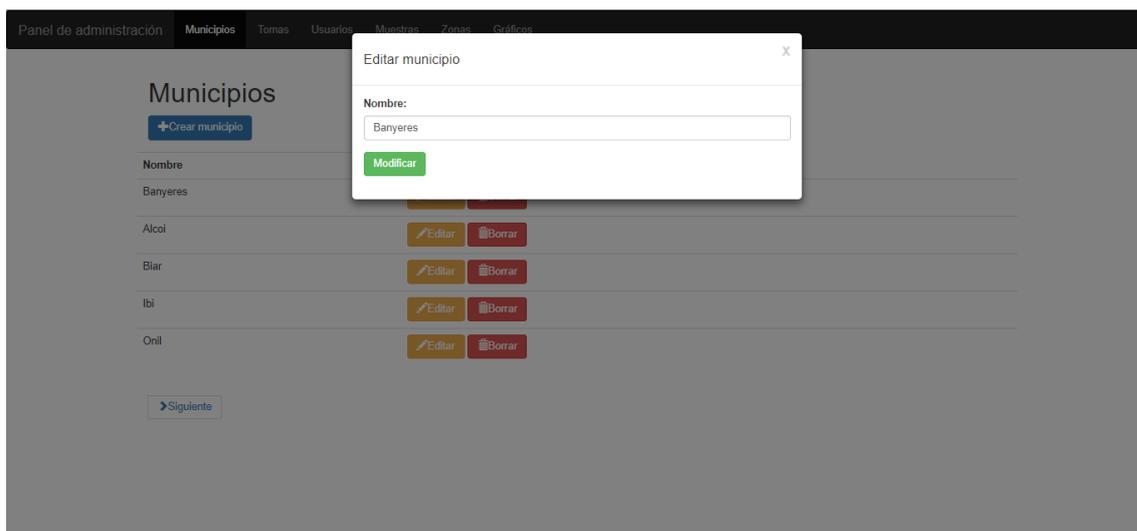
A continuación se accede a la base de datos para comprobar que efectivamente se ha creado el registro con éxito.



### **Botón editar**

Este botón permite modificar un registro ya creado. En este caso se procede a realizar una prueba cambiando el nombre del el registro que se acaba de crear.

Al pulsar sobre el botón aparece el mismo formulario que en el botón de crear, solo que recupera la información ya insertada para ser modificada.



Para la prueba, se modifica el nombre “Banyeres” por “Banyeres de Mariola”. Tras pulsar en el botón modificar se puede comprobar que el registro se ha modificado satisfactoriamente.

Panel de administración **Municipios** Tomas Usuarios Muestras Zonas Gráficos

Operación realizada  
Registro modificado con éxito.

### Municipios

[+Crear municipio](#)

Nombre	Editar	Borrar
Banyeres de Mariola	<a href="#">Editar</a>	<a href="#">Borrar</a>
Alcoi	<a href="#">Editar</a>	<a href="#">Borrar</a>
Biar	<a href="#">Editar</a>	<a href="#">Borrar</a>
Ibi	<a href="#">Editar</a>	<a href="#">Borrar</a>
Onil	<a href="#">Editar</a>	<a href="#">Borrar</a>

[> Siguiente](#)

## Botón borrar

El botón borrar llama a la función “destroy” declarada en el controlador correspondiente y le pasa como parámetro el id del registro para que lo identifique y lo pueda borrar. Este botón no muestra ningún formulario, únicamente muestra un mensaje indicando si la operación se ha realizado correctamente. Se procede a eliminar el registro que se ha creado para la prueba de funcionamiento.

Panel de administración **Municipios** Tomas Usuarios Muestras Zonas Gráficos

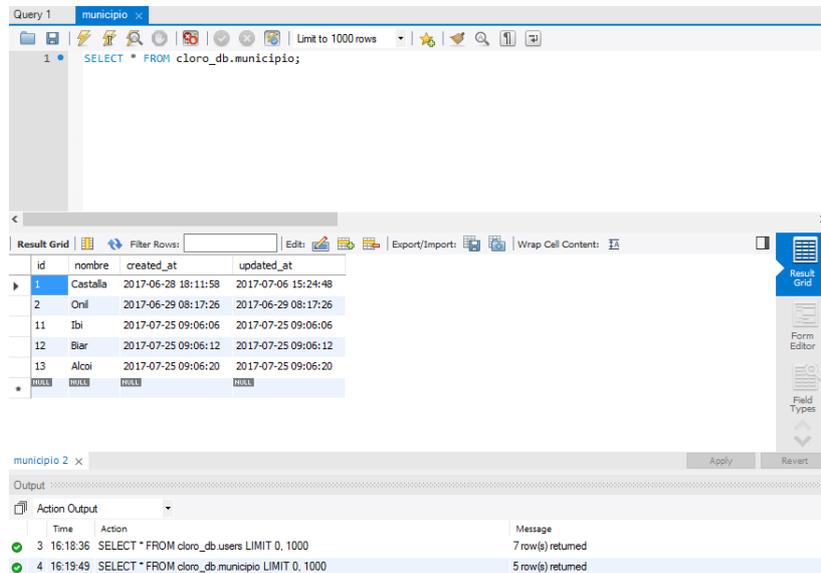
Operación realizada  
Registro borrado con éxito.

### Municipios

[+Crear municipio](#)

Nombre	Editar	Borrar
Alcoi	<a href="#">Editar</a>	<a href="#">Borrar</a>
Biar	<a href="#">Editar</a>	<a href="#">Borrar</a>
Ibi	<a href="#">Editar</a>	<a href="#">Borrar</a>
Onil	<a href="#">Editar</a>	<a href="#">Borrar</a>
Castalla	<a href="#">Editar</a>	<a href="#">Borrar</a>

Se accede a la base de datos para confirmar que el registro se ha eliminado.

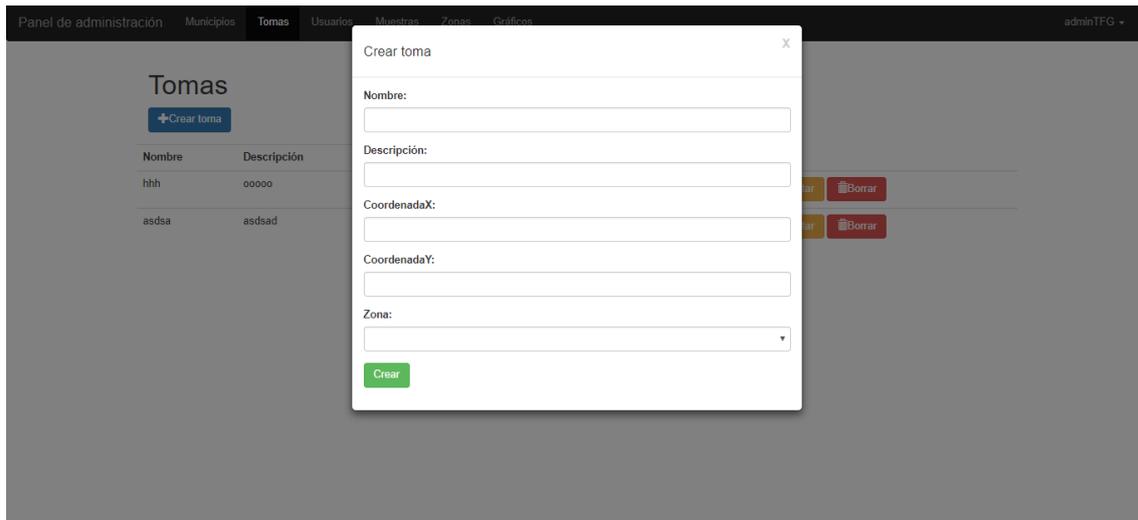


Una vez visto el funcionamiento, el siguiente paso es mostrar las demás vistas de la aplicación para el administrador. El funcionamiento es el mismo en todas ellas.

## 4.3 Vistas

### Vista de tomas





A la hora de crear registros que tienen claves ajenas, es decir, que tienen datos que pertenecen a otras tablas, se ha creado una lista desplegable cargada con los datos de la otra tabla, para que el usuario no tenga que recordar de memoria el dato a introducir.

En este caso, al crear una toma, necesita que se introduzca un campo zona, es decir, a qué zona pertenece. Aquí se puede visualizar el desplegable cargado con datos correspondientes a la tabla "Zona".

Posteriormente en las vistas de zona se podrá comprobar que los datos de la lista desplegable corresponden con los datos de la tabla de zona.

Crear toma
✕

---

**Nombre:**

**Descripción:**

**CoordenadaX:**

**CoordenadaY:**

**Zona:**

- hola
- ff
- prueba

## Vista de usuarios

En esta vista se muestran los usuarios que están dados de alta en el sistema (tanto administradores como gestores). Desde aquí, los administradores pueden dar de alta a los usuarios que quieran para que puedan acceder al sistema.

Panel de administración Municipios Tomas **Usuarios** Muestras Zonas Gráficos adminTFG

### Usuarios

+ Crear usuario

Nombre	Tipo	Dirección	Email	Teléfono		
adminTFG	Admin	adminTFG	adminTFG@adminTFG.com	967888985	Editar	Borrar
mec	Gestor	mec	mec@mec.com	969888888	Editar	Borrar
javi	Gestor	javi	javi@javi.com	968785896	Editar	Borrar
rfff	Gestor	fsdfdf	kk@kk.com	968878548	Editar	Borrar
Ruben	Admin	C/ Eraas	ruben19castalla@gmail.com	965874589	Editar	Borrar

Siguiete

Panel de administración Municipios Tomas **Usuarios** Muestras Zonas Gráficos adminTFG

### Usuarios

+ Crear usuario

Nombre	Tipo	Dirección	E-Mail	Contraseña	Confirmar contraseña	Teléfono		
adminTFG	Admin	adminTFG	adminTFG@adminTFG.com				Editar	Borrar
mec	Gestor	mec	mec@mec.com				Editar	Borrar
javi	Gestor	javi	javi@javi.com				Editar	Borrar
rfff	Gestor	fsdfdf	kk@kk.com				Editar	Borrar
Ruben	Admin	C/ Eraas	ruben19castalla@gmail.com				Editar	Borrar

Siguiete

#### Crear usuario

Nombre

Tipo

Dirección

E-Mail

Contraseña

Confirmar contraseña

Teléfono

Registrar

## Vista de muestras

Esta vista tiene algo diferente a las demás. Las funcionalidades son las mismas (crear, editar, borrar), pero se le añade un filtro en la parte superior de la tabla, para que el usuario pueda buscar los registros por su estado (Realizada o Pendiente).

Primero se muestra una vista general de los registros sin filtrar. Cuando el usuario pulsa sobre una de las dos opciones de filtro, se ejecuta una query(consulta) para que solo muestre un tipo de registro.

Panel de administración Municipios Tomas Usuarios **Muestras** Zonas Gráficos adminTFG

### Muestras

+ Crear muestra

Pendientes / Realizadas

Cloro	Olor	Color	Sabor	Turbidez	Estado	Usuario	Toma		
11	Apto	Apto	No Apto	Apto	Realizada	ddddd	2	Editar	Borrar
324	324	434	234	234	Pendiente	ddddd	1	Editar	Borrar

Windows taskbar: Escribe aquí para buscar, 16:49 25/07/2017

Visualización de muestras “pendientes”.

Panel de administración Municipios Tomas Usuarios **Muestras** Zonas Gráficos adminTFG

### Muestras

+ Crear muestra

Pendientes / Realizadas

Cloro	Olor	Color	Sabor	Turbidez	Estado	Usuario	Toma		
324	324	434	234	234	Pendiente	ddddd	1	Editar	Borrar

## Visualización de muestras realizadas.

Panel de administración | Municipios | Tomas | Usuarios | **Muestras** | Zonas | Gráficos | adminTFG

### Muestras

+ Crear muestra

Pendientes / Realizadas

Cloro	Olor	Color	Sabor	Turbidez	Estado	Usuario	Toma	
11	Apto	Apto	No Apto	Apto	Realizada	ddddd	2	<a href="#">Editar</a> <a href="#">Borrar</a>

Este es el formulario que se muestra para la toma de muestras.

Panel de administración | Municipios | Tomas | Usuarios | **Tomar muestra** | Zonas | Gráficos | adminTFG

### Tomar muestra

Cloro:

Olor:

Color:

Sabor:

Turbidez:

Estado:

Usuario:

Toma:

[Crear](#)

## Vista de zonas

En esta vista se muestran las zonas que pertenecen a cada municipio, y el usuario que se encarga de dicha zona.

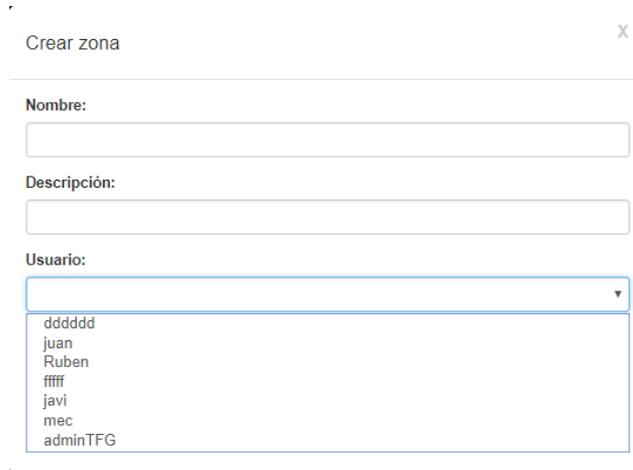
Panel de administración | Municipios | Tomas | Usuarios | Muestras | **Zonas** | Gráficos | adminTFG

### Zonas

+ Crear zona

Nombre	Descripción	Usuario	Municipio	
prueba	tttt	2	2	<a href="#">Editar</a> <a href="#">Borrar</a>
ff	ds	3	2	<a href="#">Editar</a> <a href="#">Borrar</a>
hola	díd	1	1	<a href="#">Editar</a> <a href="#">Borrar</a>

Y aquí se muestra el formulario para introducir registros.



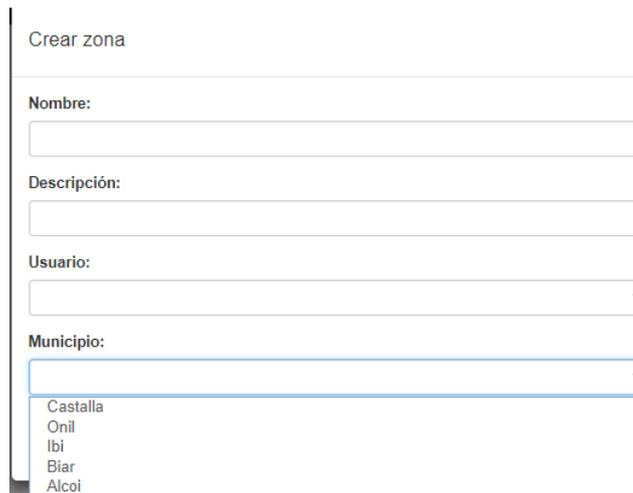
Crear zona

Nombre:

Descripción:

Usuario:

- dddddd
- juan
- Ruben
- ffff
- javi
- mec
- adminTFG



Crear zona

Nombre:

Descripción:

Usuario:

Municipio:

- Castalla
- Onil
- Ibi
- Biar
- Alcoi

### **Vista de gráficos**

En esta vista, el usuario puede consultar los datos de las muestras tomadas de una forma gráfica y cómoda, sin necesidad de acudir a los datos de la tabla.

Está equipado con un desplegable a modo de filtro para escoger la toma que se quiere consultar.

Como los datos se recogen de una query(consulta) a la base de datos cada vez que el desplegable cambia de valor, los datos están completamente actualizados.

### Datos de las muestras

Selecciona una toma de la lista

hhh



## 4.4 Panel de gestor

A continuación se procede a visualizar la parte de la aplicación a la que puede acceder un usuario gestor.

Este usuario obviamente no puede acceder a las vistas de administración, únicamente puede acceder a ver sus muestras, tanto las ya realizadas como las que tiene pendientes de realizar.

Para ver esta parte, se ha creado un usuario de prueba llamado “gestorTFG”.

### Muestras

Pendientes / Realizadas

Cloro	Olor	Color	Sabor	Turbidez	Estado	Usuario	Toma
					Pendiente	1	2

Tomar muestra

Este usuario no puede crear ni eliminar muestras, solo se le permite modificar las muestras que le ha creado el usuario administrador. En este caso, el usuario “gestorTFG” tiene una muestra pendiente de realizar.

Una vez toma los datos de la muestra y los envía, pasa a estado “Realizada”.

Panel de usuario		Muestras						gestorTFG	
		Muestras							
		Pendientes / Realizadas							
Cloro	Olor	Color	Sabor	Turbidez	Estado	Usuario	Toma		
7	Apto	Apto	No Apto	No Apto	Realizada	1	2		

## 4.5 Control de datos

El control de los datos que se introducen es uno de los aspectos fundamentales en una aplicación. Un dato introducido que no tiene el formato correcto puede llevar a un error más grave, por tanto se debe controlar este aspecto.

Para ello se hace uso de lo que se conoce como expresiones regulares. Una expresión regular es un fragmento de código que indica que formato debe tener un dato.

Como ejemplo, se muestran algunas de las expresiones regulares utilizadas en la aplicación para el control de los datos.

- Email → `required|string|email|max:100|unique:users`

Esta expresión pone como requisitos para el campo email:

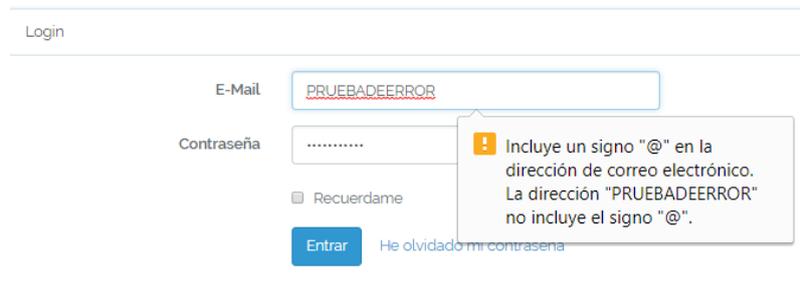
1. Es obligatorio.
2. Debe ser una cadena de caracteres.
3. Debe tener formato de email ([\\*@\\*.\\*](#))
4. Debe tener un máximo de 100 caracteres
5. No debe haber ningún email igual ya introducido.

- Teléfono → `required|numeric|regex:/^[6]{1}([0-9]{8})$`

En este caso, la información introducida en el campo teléfono debe cumplir:

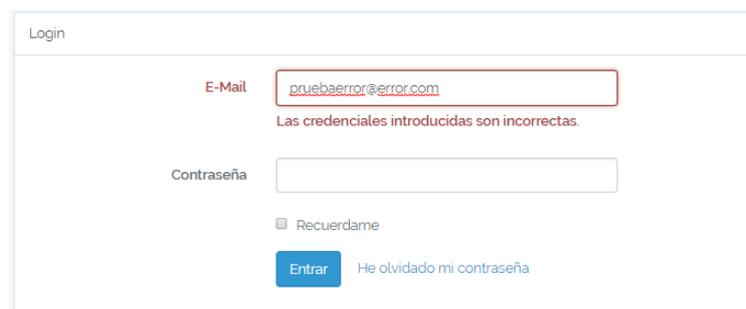
1. Es obligatorio.
2. Debe contener únicamente números
3. Debe empezar por 6 y los siguientes 8 números deben ser dígitos entre 0 y 9.
4. Debe tener una longitud de 9.

A continuación se muestran modo gráfico algunos errores provocados por introducir un formato incorrecto o por no cumplir alguna de las condiciones.



The screenshot shows a login form titled "Login". It contains two input fields: "E-Mail" and "Contraseña". The "E-Mail" field contains the text "PRUEBADEERROR" and is highlighted with a red border. A tooltip message is displayed over the "E-Mail" field, stating: "Incluye un signo '@' en la dirección de correo electrónico. La dirección 'PRUEBADEERROR' no incluye el signo '@'." Below the "E-Mail" field is a "Contraseña" field with a masked password ".....". There is a checkbox labeled "Recuérdame" and a blue "Entrar" button. A link "He olvidado mi contraseña" is located below the "Entrar" button.

Al introducir un formato incorrecto en el campo email, no nos deja acceder al sistema y nos informa de cuál es el error.



The screenshot shows the same login form as above, but with the "E-Mail" field containing the correct format "pruebaerror@error.com". Below the "E-Mail" field, a red error message is displayed: "Las credenciales introducidas son incorrectas." The "Contraseña" field is empty. The "Recuérdame" checkbox and "Entrar" button are still present, along with the "He olvidado mi contraseña" link.

Aquí se visualiza un error de autenticación, ya que el sistema detecta que el email no está dado de alta.

---

Cloro:  
  
El campo clr es requerido.

Olor:

Color:

Sabor:

Turbidez:

Estado:

Usuario:  
  
El campo usuario id usuario es requerido.

Toma:  
  
El campo toma id toma es requerido.

Como último ejemplo, se pueden observar los errores que muestra el sistema cuando se quiere crear una nueva muestra, sin rellenar todos los campos obligatorios.

## 4.6 Diseño

El diseño del proyecto se basa principalmente en dos aspectos esenciales para un buen uso del mismo.

- **Limpieza:** Se busca que el diseño sea limpio, sin demasiados elementos que sobrecarguen la vista de las páginas y que puedan distraer al usuario. La finalidad es mostrar un diseño sencillo y fácil de utilizar, para facilitarle la tarea al usuario.
- **Responsive:** Mediante el uso de Bootstrap, se ha conseguido un diseño responsive, es decir, que se adapta al dispositivo desde el cual se visualiza. Esto es muy importante, ya que ofrece una vista lo más adaptada y limpia posible al usuario, dependiendo del dispositivo que utilice para el uso de la aplicación. De este modo el usuario no está obligado a utilizar un dispositivo en concreto para poder visualizar

correctamente la web, sino que se le ofrece un abanico de posibilidades para ello.

A continuación, una de las vistas de la aplicación con los diferentes diseños, tanto para escritorio como para teléfono móvil.

#### 4.6.1 Vista desde pantalla de escritorio

Nombre	Tipo	Dirección	Email	Teléfono		
gestorTFG	Admin	gestorTFG	gestortfg@gestortfg.com	668585988		
adminTFG	Admin	adminTFG	adminTFG@adminTFG.com	667888985		
mec	Gestor	mec	mec@mec.com	669888888		
javi	Gestor	javi	javi@javi.com	668785896		
ffff	Gestor	fsdfdf	kk@kk.com	668878548		

[→ Siguiente](#)

Esta vista es la que se ha visto a lo largo del proyecto, ya que se está visualizando desde un portátil.

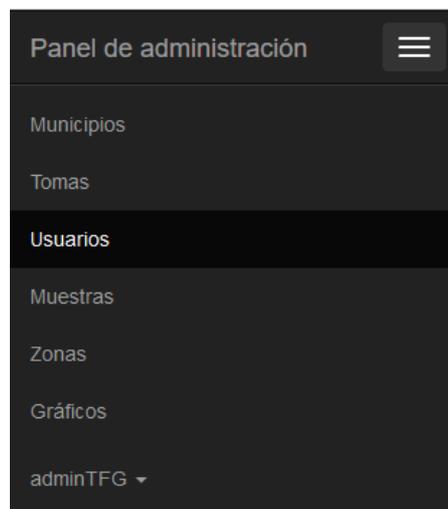
#### 4.6.2 Vista desde teléfono móvil

Nombre	Tipo	Dirección	Email		
gestorTFG	Admin	gestorTFG	gestortfg@g		
adminTFG	Admin	adminTFG	adminTFG@ac		
mec	Gestor	mec	mec@mec.co		
javi	Gestor	javi	javi@javi.cor		
ffff	Gestor	fsdfdf	kk@kk.com		

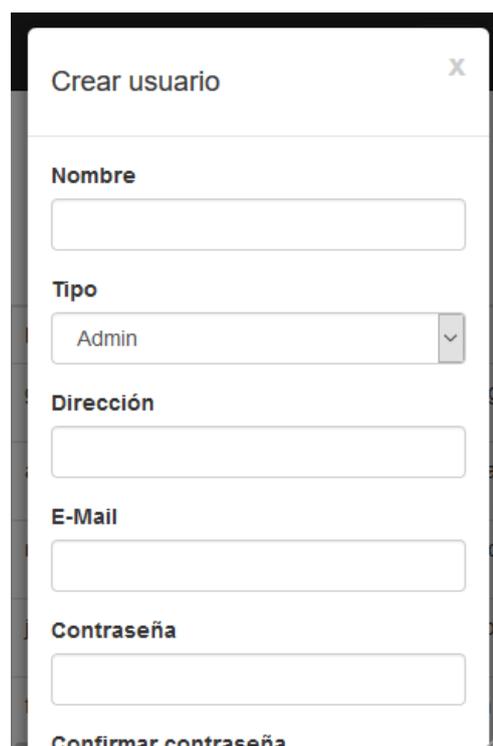
El menú pasa a ser un desplegable, ya que no se puede mostrar todo como en el escritorio. De este modo es más fácil acceder a él por el usuario. Además aparecen barras de desplazamiento para poder visualizar toda la tabla.

No se puede mostrar cada registro completo en la pantalla, ya que se necesitaría ubicarlos en varias líneas, una debajo de otra, y no quedaría clara la información.

El menú se muestra de la siguiente manera.



Y aquí, como se muestra el formulario para introducir un nuevo usuario.

A screenshot of a mobile application's "Crear usuario" form. The form has a title "Crear usuario" and a close button "X" in the top right corner. The form contains several input fields: "Nombre" (text input), "Tipo" (dropdown menu with "Admin" selected), "Dirección" (text input), "E-Mail" (text input), "Contraseña" (text input), and "Confirmar contraseña" (text input).

## 7. Buenas prácticas

A lo largo del grado de ingeniería se ha remarcado en muchas ocasiones la importancia de utilizar buenas prácticas a la hora de desarrollar cualquier proyecto.

La utilización de buenas prácticas facilita la tarea de desarrollo y mantenimiento. A continuación se listan las buenas prácticas utilizadas en este proyecto.

- Código debidamente estructurado y comentado para una lectura fácil y localizar más fácilmente cada parte.
- Uso de un diseño uniforme durante todo el proyecto.
- Funcionalidad sencilla. De este modo al usuario le resulta mucho más cómodo utilizar la aplicación y no se sentirá frustrado por manejar la aplicación.
- Errores fáciles de interpretar. Es algo que se ha remarcado mucho a lo largo del grado. Los errores deben ser visibles, con un color que resalte la importancia del error, y que el mensaje de dicho error sea claro, para que el usuario entienda sin esfuerzo dónde está el fallo.
- Uso de excepciones para controlar y evitar que la aplicación falle y se quede bloqueada.
- Uso de cifrado para almacenar la contraseña. Esto también es algo que ha sido remarcado a lo largo del grado. No se deben almacenar contraseñas en texto plano. Por ello se emplea una función que encripta la contraseña introducida por el usuario para una mayor seguridad.

## 8. Conclusiones y trabajos futuros

Este proyecto se ha realizado con la intención de cubrir una necesidad vigente. A lo largo del mismo he aprendido mucho acerca de diferentes tecnologías, y ha servido sobre todo, para tomar conciencia del trabajo que conlleva desarrollar un proyecto de este tipo.

Evidentemente tiene cosas que mejorar, ya que ha quedado alguna funcionalidad por desarrollar, pero en términos generales creo que es un proyecto que cumple satisfactoriamente con los requisitos solicitados.

Considero que este tipo de trabajos son muy útiles para el alumno que los desarrolla, ya que ayuda a entender problemas reales que surgen en este ámbito y también ayuda a desarrollar la capacidad de resolver problemas por uno mismo.

A continuación se describen las funcionalidades que han quedado pendientes de desarrollar.

1. **Generación de informes en pdf:** Esta funcionalidad consiste en generar un archivo pdf, con una estructura y una información detallada acerca de los datos de las muestras tomadas. Dicha funcionalidad no ha podido ser desarrollada, ya que la estructura y la información que debería ir en el documento, ha sido facilitado por el centro de salud dos días antes de la fecha límite de entrega del trabajo, por lo tanto no tenía tiempo de desarrollarlo.
2. **Envío de notificaciones por e-mail** Esta funcionalidad no ha sido desarrollada, porque he empleado mucho tiempo en el desarrollo del resto del proyecto y no disponía de tiempo suficiente para implementarla.
3. **Detección de incumplimientos de periodicidad:** Esta funcionalidad no se ha implementado por falta de tiempo en el desarrollo y por la dificultad de encontrar una solución de garantías.

## 9. Bibliografía

<https://styde.net/curso-basico-de-laravel-5-0/>

[http://librosweb.es/libro/bootstrap\\_3/](http://librosweb.es/libro/bootstrap_3/)

<http://victorroblesweb.es/2015/06/11/migraciones-en-laravel-5/>

<http://www.chartjs.org/>

<http://rodrigoogr.com/blog/modelo-vista-controlador/>

<https://www.clavei.es/blog/2012/que-aporta-mvc-al-desarrollo-de-aplicaciones-web/>

<https://www.uno-de-piera.com/curso-vue-js-2-laravel-5/>

[https://www.w3schools.com/bootstrap/bootstrap\\_ref\\_comp\\_glyphs.asp](https://www.w3schools.com/bootstrap/bootstrap_ref_comp_glyphs.asp)

<https://www.ohmyroot.com/vue-js-framework-javascript-progresivo/>

<https://laravel.com/>

<https://www.campusmvp.es/recursos/post/los-5-mejores-frameworks-de-javascript-en-2017.aspx>

<http://www.emasagra.es/ESP/191.asp>

<http://www.hidraqua.es/es/noticias/22/la-calidad-del-agua-del-taibilla-a-traves-de-una-aplicacion-informatica>

<https://es.wikipedia.org/wiki/XAMPP>

<http://www.neeonez.com/mejores-frameworks-gratuitos-de-php/>

<http://virtualtec.cl/manual-de-html5-y-css3/>

<https://www.adictosaltrabajo.com/tutoriales/introduccion-a-vue-js/#03>

