



**Desarrollo de una aplicación en red para la configuración del acceso a AEMET  
Opendata desde páginas web**

**Mónica García Villena**

**Tutor: José Ramón Vidal Catalá**

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2017-18

Valencia, 07 de septiembre de 2018



## Resumen

En la actualidad las aplicaciones web se han vuelto muy importantes, convirtiéndose en una herramienta que permite acceder a internet a través de un navegador web, y permitir así el acceso a toda la información desde cualquier parte del mundo.

El objetivo de este proyecto es la creación de una aplicación web que facilite el acceso a la información que proporciona el servicio de datos abiertos de la Agencia Estatal Meteorológica, para que toda persona tenga acceso a ella sin necesidad de ningún conocimiento previo de lenguaje de programación.

Una vez dentro de la aplicación, el cliente creará en primer lugar una consulta seleccionando la provincia, el municipio y los intereses acorde a lo que desea. Una vez creada la consulta el cliente podrá visualizarla en tiempo real, hacer modificaciones en ella, generar el URL si lo desea o incluso eliminarla si ya no le interesa.

## Palabras clave

- Aplicación web
- Datos abiertos
- Consultas
- Visualización de consultas
- Modificación de consultas



## Resum

En l'actualitat les aplicacions web s'han tornat molt importants, convertint-se en una eina que permet accedir a internet a través d'un navegador web, i permetre així l'accés a tota la informació des de qualsevol part del món.

L'objectiu d'aquest projecte és la creació d'una aplicació web que facilite l'accés a la informació que proporciona el servei de dades obert de l'Agència Estatal Meteorològica, perquè tota persona tinga accés a ells sense necessitat de cap coneixement previ de llenguatge de programació.

Una vegada dins de l'aplicació, el client crearà en primer lloc una consulta seleccionant la província, el municipi i els interessos d'acord amb el que desitja. Una vegada creada la consulta el client podrà visualitzar-la en temps real, fer modificacions en ella, generar el URLl si ho desitja o fins i tot eliminar-la si ja no li interessa.

## Paraules clau

- Aplicació web
- Dades obertes
- Consultes
- Visualització de consultes
- Modificació de consultes



## **Abstract**

Currently web applications have become very important, becoming a tool that allows access to the Internet through a web browser, and thus allow access to all information from anywhere in the world.

The objective of this project is the creation of a web application that facilitates access to the information provided by the open data service of the State Meteorological Agency, so that everyone has access to them without needing any prior knowledge of programming language.

Once inside the application, the client will first create a query by selecting the province, the municipality and the interests according to what you want. Once the query is created, the client can view it in real time, make changes to it, generate the URL if desired, or even delete it if it is not interested.

## **Keywords**

- Web Application
- Open data
- Queries
- View queries
- Modification of queries



## Índice

Capítulo 1. Introducción .....	7
1.1 Motivación .....	7
1.2 Objetivos del trabajo .....	7
1.3 Estructura del trabajo .....	9
Capítulo 2. Lenguajes de programación y tecnológicas utilizadas .....	10
2.1 Sistema operativo Windows 8.....	10
2.2 Netbeans.....	10
2.3 Servidor GlassFish .....	10
2.4 HeidiSQL .....	11
2.5 Java.....	11
2.6 Java Servlets.....	12
2.7 JavaScript .....	14
2.8 HTML y CSS .....	14
2.9 Servidores web y JSON.....	15
Capítulo 3. Análisis.....	20
3.1 Requisitos funcionales.....	20
3.2 Requisitos no funcionales.....	20
3.3 Restricciones .....	21
3.4 Diagramas de casos de uso.....	21
3.5 Descripción de los casos de uso .....	22
3.6 Diagramas de secuencia .....	26
Capítulo 4. Diseño .....	34
4.1 Modelo relacional.....	34
4.2 Diccionario de datos.....	35
4.3 Arquitectura del sistema.....	36
Capítulo 5. Pruebas realizadas .....	38
Capítulo 6. Resultados del trabajo .....	39
Capítulo 7. Planificación y Presupuesto.....	50
7.1 Calendario .....	50
7.2 Recursos .....	50
7.2.1 Recursos humanos.....	50
7.2.2 Recursos materiales.....	50
7.2.3 Recursos software .....	50



7.3	Descripción de tareas .....	50
7.3.1	Fase inicial.....	50
7.3.2	Primera iteración, entorno de desarrollo y base de datos .....	51
7.3.3	Segunda iteración .....	51
7.3.4	Tercera iteración.....	51
7.3.5	Fase final .....	51
7.4	Estimación del tiempo.....	51
7.5	Estimación de costes .....	52
7.5.1	Recursos humanos.....	52
7.6	Presupuesto del proyecto.....	52
7.6.1	Recursos humanos.....	52
7.6.2	Recursos hardware .....	53
7.6.3	Recursos de software.....	53
7.6.4	Costes indirectos .....	54
7.6.5	Imprevistos .....	54
7.6.6	Coste total.....	54
Capítulo 8.	Futuras ampliaciones.....	56
Capítulo 9.	Conclusiones .....	57
Capítulo 10.	Bibliografía.....	58



## Capítulo 1. Introducción

De forma general se puede definir una aplicación web como aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. En otras palabras, es un programa que se codifica en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador.

En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del navegador (por ejemplo, un applet de Java) o codificado con algún lenguaje soportado por el navegador (como JavaScript, combinado con HTML); confiándose en el navegador web para que reproduzca (renderice) la aplicación.

Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Es importante mencionar que una Página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

Cada vez es más común que en cualquier tipo de entidad (empresa, asociación, negocio, autónomo, institución...) necesite una solución informática ajustada a sus necesidades. Y cada vez es más habitual que se nombre una “Aplicación Web” como la respuesta a estas necesidades.

### 1.1 Motivación

Actualmente, cualquier persona que requiera acceder a la información de datos abiertos que nos brinda la agencia estatal de meteorología, debe tener unos conocimientos previos de programación para poder trabajar con ellos.

Dado que hoy en día, internet y las aplicaciones web facilitan en gran medida muchas de las tareas cotidianas, cualquier ámbito de trabajo podría favorecerse de estas facilidades. Una herramienta que permita el acceso sencillo y sin ningún tipo de conocimiento de programación a la información que proporciona la agencia estatal de meteorología resultaría muy atractivo para los clientes.

Entre las principales características y propiedades que debería tener una herramienta de este tipo cabría destacar la facilidad de acceso y uso por parte de los usuarios.

Por lo que en este trabajo se intenta desarrollar una aplicación web que facilite el acceso y el uso a la información de los servicios web de Aemet desde páginas web.

### 1.2 Objetivos del trabajo

El objetivo del proyecto es el desarrollo de una aplicación web que permita la configuración de consultas a Aemet Open Data sin ningún conocimiento de programación.



Al implementar esta aplicación como una aplicación web, adquiere una serie de ventajas que la separan del resto de aplicaciones tradicionales. Entre estas ventajas destacamos: compatibilidad multiplataforma, inmediatez de acceso (estas aplicaciones no necesitan ser descargadas, instaladas y configuradas, directamente están listas para ser utilizadas independientemente de la configuración o hardware del equipo), menos requerimiento de memoria (las aplicaciones web tienen una más razonable demanda de memoria RAM que los programas instalados localmente), múltiples usuarios interactúan de forma concurrente, etc.

Tras varias alternativas para el desarrollo de la aplicación, nos hemos decantado por el uso de Java como lenguaje de programación y Java Servlets como una clase de Java, ya que permite ampliar las capacidades de un servidor. Un estudio más detallado lo podremos encontrar en el capítulo 3.

Existen varios objetivos secundarios que se deben conseguir:

- La aplicación deberá permitir crear una consulta.
- La aplicación deberá permitir visualizar la consulta creada seleccionada.
- La aplicación deberá permitir generar el URL correspondiente a la consulta seleccionada.
- La aplicación deberá permitir modificar/eliminar la consulta seleccionada.

Además, la aplicación debe incluir la definición de servicios web REST para la visualización de la información obtenida de la Agencia Estatal de Meteorología.

Se define un servicio web REST como un servicio que no tiene estado, lo que significa que, entre dos llamadas, el servicio pierde sus datos. No se puede llamar a un servicio REST y pasarle unos datos y esperar que “nos recuerde” en la siguiente petición. Si quiero que un servicio REST me recuerde, debo pasarle quien soy en cada llamada. Y lo mismo aplica para el resto de información.

Los sistemas RESTful tienen las siguientes restricciones:

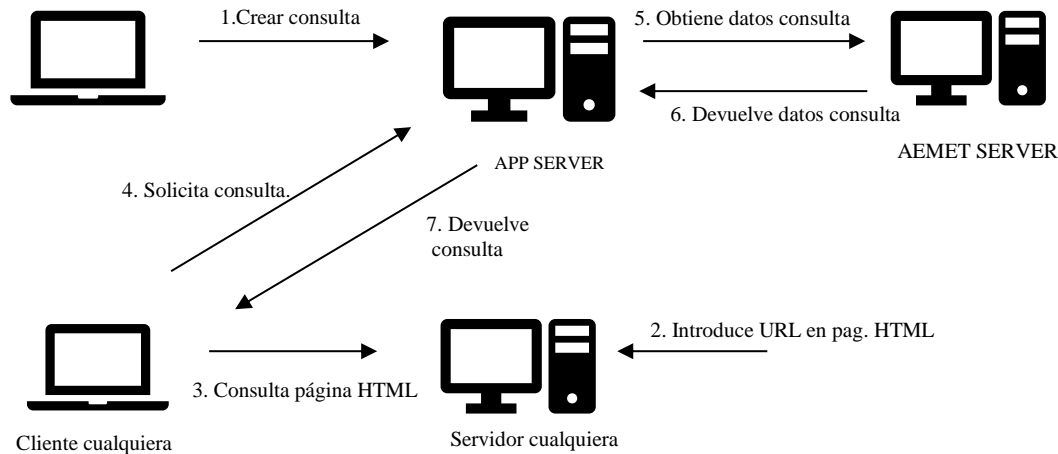
- **Cliente-servidor:** el cliente va separado del servidor. Ambas partes son completamente independientes, y pueden ser reemplazadas de una forma transparente para la otra parte.
- **Sin estado:** cada petición realizada que recibe el servidor debe contener toda la información necesaria para poder resolverla y además debe ser independiente.
- **Cacheable:** cada petición tiene que indicar si el resultado es cacheable o no.
- **Interfaz uniforme:** la cual representa una forma única de recursos, por lo que un servicio Rest debe tener una única dirección URI.

Algunas características importantes de estos servicios son :

- Utiliza explícitamente los metodos HTTP. Son los métodos indicados en la cabecera HTTP por parte del cliente y estos son cuatro: GET recoge información, POST crea un nuevo recurso en el server, PUT modifica o actualiza el estado y DELETE elimina un recurso del server.
- Las respuestas a las peticiones se hacen en XML o JSON.



El diagrama de nuestra aplicación será el siguiente:



### 1.3 Estructura del trabajo

La memoria del presente trabajo de fin de grado se divide en un listado de capítulos que se detallan a continuación.

En el segundo capítulo se detallan las horas de trabajo y las fases en las que consiste el proyecto.

El tercer capítulo profundiza en las principales herramientas y tecnologías utilizadas para el desarrollo del proyecto.

El cuarto capítulo es un análisis de los requisitos que ha de cumplir la aplicación a desarrollar y las especificaciones de dicha aplicación.

En el quinto capítulo se definen los diagramas y la arquitectura de la aplicación web.

En el sexto capítulo se comprueba el correcto funcionamiento de la aplicación.

El séptimo capítulo recoge un ejemplo de uso de la aplicación.

El octavo capítulo muestra una planificación inicial del trabajo y el presupuesto del proyecto.

El capítulo noveno presenta las conclusiones sobre la realización del trabajo, comentando en qué grado se han conseguido los objetivos iniciales.

Por último, en la bibliografía se citan las fuentes consultadas durante la realización de este trabajo.



## Capítulo 2. Lenguajes de programación y tecnológicas utilizadas

La aplicación ha sido desarrollada utilizando las siguientes tecnológicas:

- Windows, es el sistema operativo utilizado
- NetBeans, es el entorno de desarrollo
- GlassFish, es el servidor de aplicaciones utilizado
- Tomcat, es el contenedor de Servlets
- HeidiSQL, es el programa de base de datos utilizado
- Java, lenguaje de programación
- Java Servlets, es una clase en el lenguaje de programación Java
- HTML y CSS, es el lenguaje de la web
- Servidores web y JSON, es el texto ligero para el intercambio de datos

Este conjunto de tecnológicas nos permite desarrollar programas y aplicaciones, por lo que no debe confundirse con un lenguaje de programación en sí mismo, sino que debe definirse como un conjunto de herramientas tecnológicas que funcionan de manera coordinada.

### 2.1 Sistema operativo Windows 8

He elegido Windows como sistema operativo debido a que es el sistema que lleva instalado por defecto el portátil que se ha utilizado para desarrollar el proyecto, aunque se sabe que los servidores Windows son los menos utilizados debido a que los servidores Linux ofrecen un amplio abanico de ventajas. Es un sistema operativo mucho más estable y es capaz de manejar grandes cantidades de procesos mejor que Windows.

Aun sabiendo esto, no se ha tenido ningún problema en utilizar el sistema operativo Windows en el desarrollo del proyecto.

### 2.2 Netbeans

Se ha escogido Netbeans porque es un software de desarrollo libre. Además, es el software que se utilizó en la asignatura de Ingeniería de sistemas telemáticos del grado.

Ofrece una gran lista de ventajas:

- Es un buen editor de código, multilenguaje, con sugerencias de código, acceso a clases y un largo etcétera.
- El debugger que incluye el IDE es muy útil para encontrar fallos.
- Dispone en una buena optimización del código.
- El propio Netbeans se puede conectar a distintos gestores de bases de datos.

### 2.3 Servidor GlassFish

He utilizado como servidor de aplicaciones GlassFish porque es gratuito, de código libre, y permite ejecutar aplicaciones y además viene instalado por defecto en el software de Netbeans.

## 2.4 HeidiSQL

He elegido HeidiSQL porque es un software de código abierto y libre y sirve para conectarse a servidores MySQL (y sus derivaciones como MariaDB y Percona Server) y se utilizó en la asignatura de sistemas telemáticos para la gestión de información del grado.

En el proyecto he puesto solo los municipios de la Comunidad Valenciana, ampliable a todos los municipios de España.

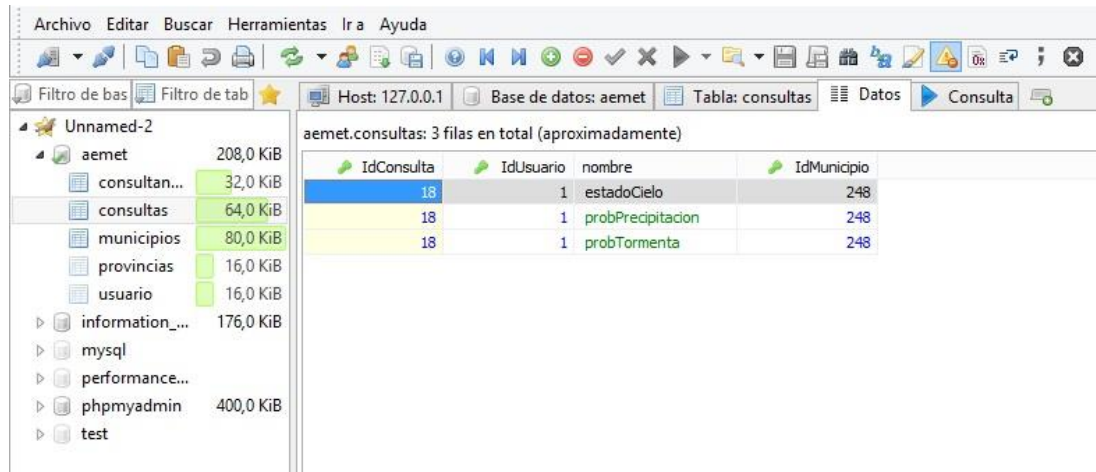


Ilustración 1. Bases de datos HeidiSQL

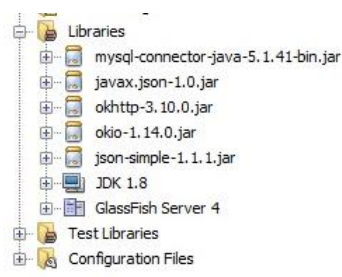
## 2.5 Java

He utilizado Java como el lenguaje de programación para realizar todas las operaciones del lado del servidor, incluida la gestión de la base de datos, y porque es el lenguaje que se ha estudiado durante el grado.

Algunas de las ventajas más importantes por las que programar en java son:

- Java es una plataforma (multiplataforma) independiente.
- El lenguaje Java es orientado a objetos.
- Java no tiene problemas de liberación de memoria en el sistema.
- El lenguaje Java es relativamente fácil de aprender.
- Dispone de una serie de librerías estándar.

En este proyecto he utilizado las siguientes librerías:



## Ilustración 2. Librerías

```
45
46 //CONSULTA Y SABER DONDE ESTAMOS
47 SQL="SELECT * FROM usuario WHERE email='"+email+"'";
48 rs= st.executeQuery(SQL);
49 if(rs.next()){
50     out.println("<script type='text/javascript' language='\"javascript\"> alert(\"Atencion, ya existe el m");
51     out.println("window.location.href=\"index.html\";");
52     out.println("</script>");
53 }
54 }else{
55     if(pass!=rpass){
56         System.out.println(pass);
57         System.out.println(rpass);
58         out.println("<script type='text/javascript' language='\"javascript\"> alert(\"Las contraseñas son inco");
59         out.println("window.location.href=\"index.html\";");
60         out.println("</script>");
61     }else{
62         SQL="INSERT INTO usuario (IdUsuario,nombre,Contraseña,email) VALUES ('"+null+"','"+realname+"','"+pass+"','"+";
63         st.executeUpdate(SQL);
64         out.println("<script type='text/javascript' language='\"javascript\"> alert(\"Usuario registrado con e");
65         out.println("window.location.href=\"index.html\";");
66         out.println("</script>");
67     }
68 }
69 //CERRAMOS CONEXION
70 rs.close();
71 out.close();
72 con.close();
73 }catch (IOException | SQLException e){
74     System.out.println("Error en Registro: "+e.toString());
75     return;
76 }
77 }
```

Ilustración 3. Ejemplo código Java en NetBeans

## 2.6 Java Servlets

Ya que he escogido Java como lenguaje de programación, he utilizado los servlets como clase de este para ampliar las capacidades de un servidor, ya que tienen una gran lista de características y además se han estudiado en la asignatura de sistemas telemáticos para la gestión de información del grado.

Las principales características de los servlets son las siguientes:

- Son independientes del servidor utilizado y de su sistema operativo.
- Los servlets pueden llamar a otros servlets.
- Obtienen fácilmente la información.
- Permiten además la utilización de cookies y sesiones.
- Pueden actuar como enlace
- Permiten la generación dinámica de código HTML

```

probPrecipitacion = req.getParameter("probPrecipitacion");
probTormenta = req.getParameter("probTormenta");
nieve = req.getParameter("nieve");
probNieve = req.getParameter("probNieve");
vientoAndRachaMax = req.getParameter("vientoAndRachaMax");
temperatura = req.getParameter("temperatura");
sensTermica = req.getParameter("sensTermica");
humedadRelativa = req.getParameter("humedadRelativa");

if(humedadRelativa ==null && sensTermica==null && temperatura==null && vientoAndRachaMax==null && probNieve==null && niev
System.out.println("ENIRA AQUI");
sesion.setAttribute("nombreProv1", provincia);
System.out.println(provincia);
out.println("<script type='text/javascript' language='javascript'> alert(\"Debes seleccionar minimo un campo\");");
out.println("window.location.href=\"crearConsulta\";");
out.println("</script>");
}else{
SQL1 = "INSERT INTO consultaNueva(IdConsulta, IdUsuario) VALUES (NULL, "+IdUsuario+")";
st.executeUpdate(SQL1);
String SQL5="SELECT consultanueva.IdConsulta FROM consultanueva WHERE consultanueva.IdUsuario="+IdUsuario+" ORDER BY cons
rs=st.executeQuery(SQL5);
if(rs.next()){
int IdConsulta=rs.getInt("IdConsulta");
//sesion.setAttribute("IdConsulta", IdConsulta);
System.out.println(IdConsulta);
estadoCielo="estadoCielo";
String SQL7 = "INSERT INTO consultas(IdConsulta,IdUsuario,nombre,IdMunicipio) VALUES (" +IdConsulta+", "+IdUsuario+", " +
st.executeUpdate(SQL7);
if(precipitacion!=null){
precipitacion="precipitacion";
String SQL8 = "INSERT INTO consultas(IdConsulta,IdUsuario,nombre,IdMunicipio) VALUES (" +IdConsulta+", "+IdUsuario+", " +

```

Ilustración 4. Ejemplo de Servlet en NetBeans

```

@WebServlet(urlPatterns = {"/seleccionado"})
public class seleccionado extends HttpServlet {
    @Override
    protected void doPost (HttpServletRequest req, HttpServletResponse res) throws IOException {
        try {
            Class.forName("com.mysql.jdbc.Driver");
        }
        catch(ClassNotFoundException e){
            System.out.println("Error abriendo el driver: "+e.toString());
            return;
        }
        System.out.println("ESTAMOS EN SELECCIONADO");
        try{

            Connection con;
            Statement st;
            ResultSet rs;
            PrintWriter out;
            String username, authST, IdUsuarioST, nombreProv;
            String SQL,SQL1, provincias, nombre, estadoCielo, precipitacion,probPrecipitacion, probTormenta, nieve, probNieve;
            String vientoAndRachaMax,sensTermica, temperatura, humedadRelativa, provincia;

            con =DriverManager.getConnection("jdbc:mysql://localhost/aemet","root","");
            st = con.createStatement();
            out = res.getWriter();
            res.setContentType("text/html");

            HttpSession sesion =req.getSession(true);
            Object usernameaux=sesion.getAttribute("username");
            Object IdUsuarioaux=sesion.getAttribute("IdUsuario");

```

Ilustración 5. Ejemplo de Servlet en java en NetBeans

En este proyecto he utilizados los siguientes servlets:

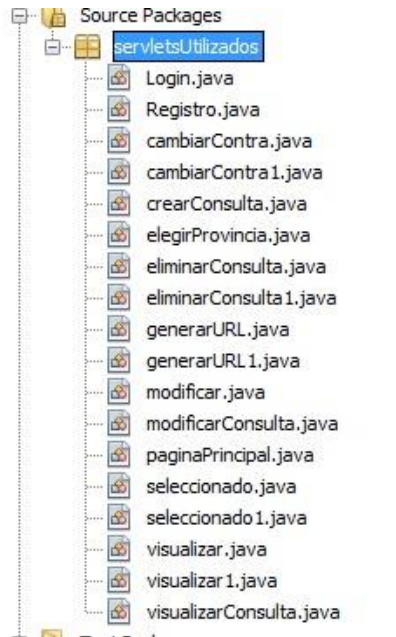


Ilustración 6. Servlets utilizados

## 2.7 JavaScript

He utilizado JavaScript como lenguaje de programación interpretado porque cuenta con múltiples opciones de efectos visuales y porque es una gran solución para la validación de datos en un formulario.

Con JavaScript he creado todas las alertas.

```
if(rs.next()==false){
    if(username.isEmpty()){
        System.out.println("CAMPO VACIO");
        out.println("<script type='text/javascript' language='javascript'> alert(\"Campo vacio\");");
        out.println("window.location.href=\"index.html\";");
        out.println("</script>");
    }else{
        out.println("<script type='text/javascript' language='javascript'> alert(\"Correo incorrecto\");");
        out.println("window.location.href=\"index.html\";");
        out.println("</script>");
    }
}
}else{
    email=rs.getString("usuario.email");
```

Ilustración 7. Ejemplo de alertas con JavaScript

## 2.8 HTML y CSS

He utilizado HTML para el diseño de la web debido a que es un código sencillo y simplificado. Y CSS para el diseño gráfico de la aplicación .

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="iso-8859-1">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="Responsive sidebar template with sliding effect and dropdown menu based on bootstrap">
  <title>Consultas AEMET</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-+
  crossorigin="anonymous">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.0.13/css/all.css" integrity="sha384-DNOH268U8h2fK
  crossorigin="anonymous">
  <link rel="stylesheet" href="//maliyu.github.io/custom-scrollbar/jquery.mCustomScrollbar.min.css">
  <link rel="stylesheet" href="assets/css/custom.css">
  <link rel="stylesheet" href="assets/css/custom-themes.css">
  <link rel="shortcut icon" type="image/png" href="assets/img/favicon.png" />
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
</head>

<body>
<body>\n" +
<div class="page-wrapper chiller-theme sidebar-bg bgl toggled" overflow: hidden;>
<a id="show-sidebar" class="btn btn-sm btn-dark" href="#">
  <i class="fas fa-bars"></i>
</a>
<nav id="sidebar" class="sidebar-wrapper">
<div class="sidebar-content">
<div class="sidebar-header">
<div class="sidebar-brand">
```

Ilustración 8. Ejemplo código CSS

## 2.9 Servidores web y JSON

He utilizado el servicio OpenData de AEMET, de la Agencia Estatal de Meteorología. Para poder trabajar con él, es necesario trabajar con certificados digitales. El protocolo de acceso es obligatoriamente https, el cual emplea TLS (evolución de SSL). La conexión https con AEMET requiere que el servidor de AEMET envíe un certificado o cadena de certificados al navegador cliente para que este compruebe que la información no ha sido alterada. La cadena de certificados debe acabar en un certificado autofirmado calificado de certificado raíz de confianza que el navegador ya debería tener instalado previamente en cierto almacén de certificados o habrá que instalarlo explícitamente para que pueda funcionar correctamente y de forma segura. En caso contrario, el navegador avisa de alguna manera que la conexión https no es segura. En este caso, el certificado raíz de confianza se denomina ACCVRAIZ1 emitido por la organización ACCV (Agencia de Tecnología y Certificación Electrónica, <http://www.accv.es>).

Esta serie de acciones llevadas a cabo por el navegador se trasladan al contexto de Java. Del mismo modo que con el navegador, hay que instalar el certificado raíz de la cadena de certificados enviados por el sitio web de AEMET como certificado de confianza, en algún sitio. Para conseguir este certificado, lo descargamos desde la página web de ACCV.

Quiénes somos | Ciudadanos | Empresas | Administración Pública | Sala de prensa | Ayuda

Estás en Inicio

Acceso a Área personal GVA

### Tipos de certificados

	CIUDADANOS	EMPRESAS	ADM. PÚBL.
De Ciudadano en Soporte Software	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
De Ciudadano en Tarjeta Criptográfica	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
De Representante de Entidad en software	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
De Representante de Entidad en tarjeta	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
De Pertenencia a Empresa	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
De Empleado Público	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
De Sede Electrónica Administrativa	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
De Sello Electrónico de Órgano	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
De Servidor con soporte SSL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
De Servidor con soporte VPN	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
De Firma de Código	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
De Aplicación	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Encuentra tu Punto de Registro más cercano:  
Valencia Ademuz

Contacta con nosotros  
o llámanos al 902 482 481

Generar CERTIFICADOS

Descargar CERTIFICADOS ACCV

NUESTROS PRODUCTOS Y SERVICIOS

/ayuda/descargar-certificados-digitales/

Ilustración 9. Página web de la agencia ACCV

Comprobación de la Firma Electrónica >

Estado de los Certificados >

Para poder utilizar sus certificados digitales debe registrar en el navegador web **las claves públicas** de los certificados digitales de la Agencia de Tecnología y Certificación Electrónica (ACCV). Son necesarias para verificar que el certificado digital que se va a utilizar ha sido emitido por una Autoridad de Certificación en la que se confía.

A continuación puede descargar dichas claves públicas y las guías de instalación en los diferentes navegadores web.

- Cert Autoridad Certificación Raíz: ACCV Raíz 1 (CRT 4KB) - Vigente hasta 31/12/2030**
- Cert Autoridad Certificación Raíz: ACCV Raíz 2 (SHA-256) - Universidad Valenciana (CRT 3KB) - Vigente hasta 01/07/2021
- Cert Autoridad Certificación para personas físicas (EJBCA): ACCV-CA2 (CRT 3KB) - Vigente hasta el 01/05/2016
- Cert Autoridad Certificación para personas jurídicas: ACCV-CA1 (CRT 3KB) - Vigente hasta el 01/05/2016
- Cert Autoridad Certificación para personas físicas (Nueva Jerarquía): ACCVCA-120
- Cert Autoridad Certificación para personas físicas (SHA-256): ACCVCA-120
- Cert Autoridad Certificación para entidades (Nueva jerarquía): ACCVCA-110
- Cert Autoridad Certificación para entidades (SHA-256): ACCVCA-110
- Cert Autoridad Certificación para personas físicas y logon en Windows (Nueva jerarquía): ACCVCA-130
- Cert Autoridad Certificación para personas físicas y logon en Windows (SHA-256): ACCVCA-130
- Cert Autoridad Certificación para personas físicas: CAGVA (CRT 2 KB) - CADUCADO (ver más información al final de esta página)
- Certificado de la Autoridad de Sellado de Tiempo (CER 2KB) - Vigente hasta 18/11/2016

Valencia Ademuz

Localizar por mapa

Contacta con nosotros  
o llámanos al 902 482 481

Ilustración 10. Descarga del certificado ACCRAIZ1 de ACCV

Java posee un almacén donde se pueden guardar los certificados de manera centralizada, pero también pueden emplearse almacenes particulares para tal efecto. La ejecución del siguiente comando en línea de comandos

```
> keytool -import -file ACCVRAIZ1.crt -storepass ESCRIBID_UNA_CLAVE_DE_ACCESO -keystore ALMACEN -alias ALIAS_DEL_CERTIFICADO
```

consigue guardar el certificado digital contenido en el fichero ACCVRAIZ1.crt en un almacén (fichero) cuyo nombre es el que acompaña al parámetro -keystore,



donde:

- **ESCRIBID UNA CLAVE DE ACCESO:** puede ser cualquier clave (no es necesaria en Java si solo leemos certificados)
- **ALMACEN:** nombre del fichero que almacenará claves y certificados.
- **ALIAS DEL CERTIFICADO:** forma de identificarlo fácilmente dentro del almacén.

El comando keytool viene con la distribución del JDK que esté descargada en el ordenador, por lo que tiene las mismas consideraciones de ejecución que, por ejemplo, javac. Tenemos que indicar a la máquina virtual donde puede encontrar el certificado que le hará falta para validar la conexión https con la siguiente instrucción en nuestro código Java antes de proceder a realizar cualquier petición HTTPS hacia el servidor:

ALMACEN denota el nombre del fichero que contiene el almacén de certificados generado anteriormente mediante la herramienta keytool, añadiendo el path oportuno.

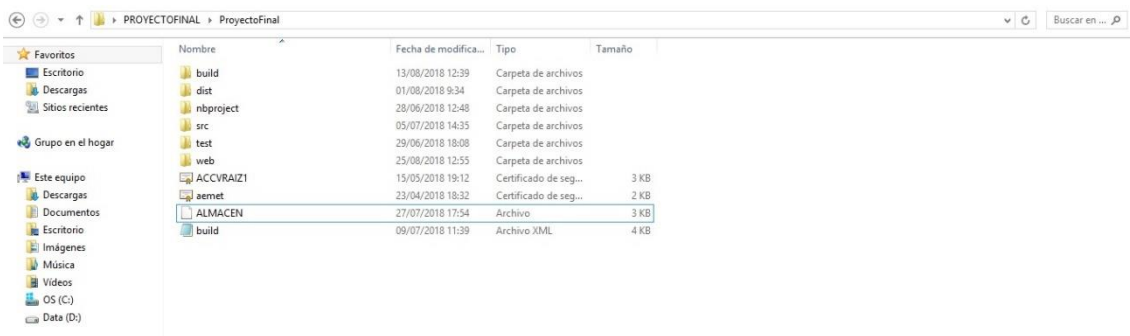


Ilustración 11. Ubicación del ALMCEN

```
String IdConsulta = req.getParameter("IdConsulta");  
sesion.setAttribute("IdConsulta", IdConsulta);  
System.setProperty("javax.net.ssl.trustStore", "C:\\Users\\Monica Garcia\\Desktop\\PROYECTOFINAL\\ProyectoFinal\\ALMACEN");  
System.out.println("CONSULTA EN VISUALIZAR: "+IdConsulta);
```

Ilustración 12. Indicación a la máquina virtual de donde está el certificado

Una vez instalado el certificado raíz, se solicita la obtención de la API Key de AEMET. Introduciendo una dirección de correo electrónico, se recibe un correo en el que se debe confirmar la generación de la clave (pulsando en el enlace Confirmar generación API Key dentro del correo recibido), tras lo cual se recibe de nuevo otro correo electrónico en el que se aprecia claramente la clave necesaria. Esta clave es efímera, por lo que requeriría su renovación transcurridos 90 días.



Ilustración 13. Clave API key

Como ejemplo de funcionamiento de la página de datos abiertos de Aemet, entramos en la página de open data Aemet, seleccionamos Acceso Desarrolladores, Documentación AEMET Open data, predicciones específicas y /api/prediccion/especifica/municipio/horaria/{municipio}. Rellanos el campo del municipio con el código 46015 y obtenemos:

GET /api/prediccion/especifica/municipio/diaria/{municipio} Predicción por municipios diaria. Tiempo actual.

GET /api/prediccion/especifica/municipio/horaria/{municipio} Predicción por municipios horaria. Tiempo actual.

**Implementation Notes**  
Predicción horaria para el municipio que se pasa como parámetro (municipio). Presenta la información de hora en hora hasta 48 horas.

**Response Class (Status 200)**  
respuesta con éxito

Model | Example Value

```
{
  "descripcion": "Éxito",
  "estado": 200,
  "datos": "string",
  "metadatos": "string"
}
```

Response Content Type: application/json

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
municipio	46015	Código de municipio <a href="http://www.ine.es/daco/daco42/codmun/codmunmapa.htm">http://www.ine.es/daco/daco42/codmun/codmunmapa.htm</a>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
------------------	--------	----------------	---------

Ilustración 14. Ejemplo de datos abiertos de AEMET

Request URL

https://opendata.aemet.es/opendata/api/prediccion/especifica/municipio/horaria/46015

Response Body

```
{
  "descripcion": "Éxito",
  "estado": 200,
  "datos": "https://opendata.aemet.es/opendata/sh/4b0a1ad3",
  "metadatos": "https://opendata.aemet.es/opendata/sh/93a7c63d"
}
```

Response Code

200

Response Headers

```
{
  "connection": "close",
  "content-length": "175",
  "content-type": "application/json;charset=ISO-8859-15",
  "date": "Mon, 27 Aug 2018 09:31:19 GMT",
  "remaining-request-count": "149",
  "server": "Apache/2.2.15 (CentOS)"
}
```

Ilustración 15. Ejemplo de datos abiertos de AEMET



## Capítulo 3. Análisis

En este apartado, nos concentramos en el análisis de los requisitos que cumple la aplicación a desarrollar y las especificaciones de dicha aplicación.

El resultado de este capítulo nos ayuda a conseguir los objetivos planteados de la aplicación con los requisitos identificados y las funcionalidades explicadas detalladamente.

### 3.1 Requisitos funcionales

Los requisitos funcionales definen una función del sistema de software o sus componentes.

#### Usuario no registrado:

- Número 01: El sistema muestra un formulario para registrarse.
- Número 02: El sistema comprueba si los datos introducidos en el registro son correctos o no.

#### Usuario registrado:

- Número 03: El sistema muestra un formulario de introducción datos de login (email y contraseña).
- Número 04: El sistema comprueba los datos introducidos para ver si son correctos o no.
- Número 05: El sistema muestra un formulario para solicitar una nueva contraseña.
- Número 06: El sistema permite al usuario poder crear una consulta.
- Número 07: El sistema muestra un listado con las consultas creadas por el usuario.
- Número 08: El sistema permite al usuario poder visualizar la consulta seleccionada.
- Número 09: El sistema permite al usuario poder modificar la consulta seleccionada.
- Número 10: El sistema permite al usuario poder generar el URL la consulta seleccionada.
- Número 11: El sistema permite al usuario poder eliminar la consulta seleccionada.
- Número 12: El sistema permite al usuario cerrar la sesión.

### 3.2 Requisitos no funcionales

Se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema, sino a las propiedades del sistema.

Hay diferentes tipos de requisitos no funcionales.

- Disponibilidad
  - Número 1: El sistema debe estar disponible las 24 horas del día, los 7 días a la semana.
- Accesibilidad
  - Número 02: Al sistema se debe acceder a través de un navegador web.
- Interfaz
  - Número 03: El sistema tiene una interfaz de usuario sencilla.
- Compatibilidad
  - Número 04: La aplicación web debe ser compatible con todos los navegadores web.

- Seguridad
  - Número 05: Los usuarios debe identificarse a través de su correo electrónico y su contraseña que previamente han registrado.

### 3.3 Restricciones

- Número 01: No pueden existir dos usuarios con el mismo correo.
- Número 02: Sólo se puede acceder al sistema si estas dado de alta en él.
- Número 03: Sólo pueden interactuar con las consultas los usuarios que estén logueados.

### 3.4 Diagramas de casos de uso

Un caso de uso es una secuencia de acciones realizadas por el sistema. Los diagramas de caso de uso muestran una relación entre los actores y los casos de uso de un sistema. Una relación es una conexión entre los elementos.

A continuación, se pueden observar los diferentes diagramas de casos de uso y su correspondiente explicación.

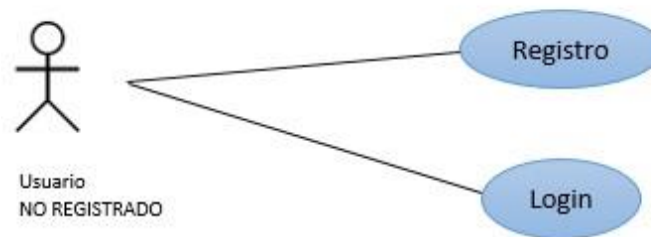


Ilustración 18. Diagrama de caso de uso de un usuario no registrado.

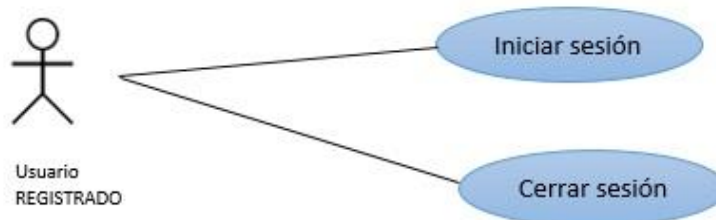


Ilustración 19. Diagrama de caso de uso de un usuario si registrado.

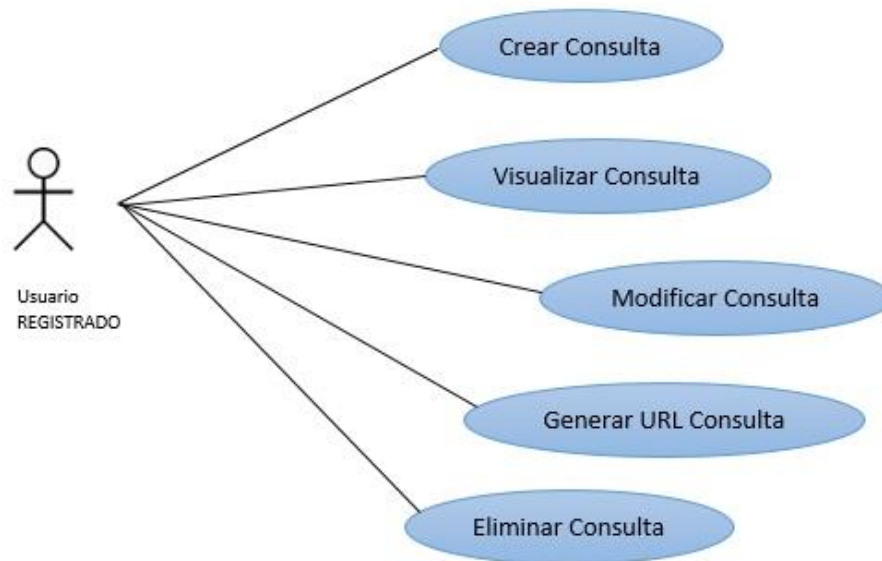


Ilustración 20. Diagrama de caso de uso de un usuario registrado.

### 3.5 Descripción de los casos de uso

#### - Registro

<b>Caso de uso:</b> Registro	
<b>Actor principal:</b> Usuario no registrado	
<b>Precondición:</b> -	
<b>Disparador:</b> El usuario no registrado quiere registrarse para acceder a la aplicación	
<b>Escenario principal de éxito:</b>	
<b>Actor</b>	<b>Sistema</b>
	1. El sistema muestra un formulario para introducir nombre, email y password.
2. El usuario completa el formulario y pulsa el botón “Registrarse”:	
	3. El sistema valida los datos introducidos
	4. El sistema da de alta al nuevo usuario registrando los datos introducidos e indica al usuario que se ha dado de alta.
<b>Extensiones:</b> -	

Tabla 2. Registro

- **Login**

<b>Caso de uso:</b> Login	
<b>Actor principal:</b> Usuario registrado	
<b>Precondición:</b> -	
<b>Disparador:</b> El usuario quiere acceder a la aplicación	
<b>Escenario principal de éxito:</b>	
<b>Actor</b>	<b>Sistema</b>
	1. El sistema muestra un formulario para introducir username y password.
2. El usuario introduce los datos de acceso y pulsa el botón “Aceptar”.	
	3. El sistema verifica los datos introducidos
4. El sistema verifica los datos introducidos.	
<b>Extensiones:</b>	
4a. El sistema detecta que algún campo está vacío.	
<ul style="list-style-type: none"> <li>- 4. a1) El sistema muestra un aviso indicando que hay un campo que no está rellenado.</li> <li>- 4. a2) Vuelve al punto 2.</li> </ul>	
4b. El sistema detecta que los datos introducidos son incorrectos.	
<ul style="list-style-type: none"> <li>- 4. b1) El sistema muestra un aviso indicando que los datos son incorrectos.</li> <li>- 4. b2) Vuelve al punto2.</li> </ul>	

Tabla 3. Login

- **Cerrar sesión**

<b>Caso de uso:</b> Cerrar sesión	
<b>Actor principal:</b> Usuario registrado	
<b>Precondición:</b> El usuario ha iniciado la sesión	
<b>Disparador:</b> El usuario quiere cerrar su sesión	
<b>Escenario principal de éxito:</b>	
<b>Actor</b>	<b>Sistema</b>
1. El usuario pulsa el botón “Cerrar sesión”.	



	2. El sistema cierra la sesión actual del usuario.
<b>Extensiones:</b> -	

**Tabla 4. Cerrar sesión**

- **Crear consulta**

<b>Caso de uso:</b> Crear Consulta	
<b>Actor principal:</b> Usuario registrado	
<b>Precondición:</b> El usuario ha iniciado la sesión	
<b>Disparador:</b> El usuario quiere crear una consulta	
<b>Escenario principal de éxito:</b>	
<b>Actor</b>	<b>Sistema</b>
1. El usuario selecciona la opción de crear consulta.	
	2. El sistema muestra un desplegable con un listado de todas las provincias de España.
3. El usuario selecciona la provincia que desea.	
	4. El sistema muestra un listado con las posibles opciones de consulta y un desplegable con todos los municipios de la provincia escogida previamente.
5. El usuario selecciona las opciones y el municipio que desea y pulsa el botón de “Crear consulta”.	
<b>Extensiones:</b>	
4a. El sistema detecta que algún campo está vacío.	
<ul style="list-style-type: none"> <li>- 4. a1) El sistema muestra un aviso indicando que hay un campo que no está rellenado.</li> <li>- 4. a2) Vuelve al punto 4.</li> </ul>	

**Tabla 5. Crear Consulta**



- **Modificar consulta**

<b>Caso de uso:</b> Modificar Consulta	
<b>Actor principal:</b> Usuario registrado	
<b>Precondición:</b> El usuario ha iniciado la sesión	
<b>Disparador:</b> El usuario quiere modificar una consulta	
<b>Escenario principal de éxito:</b>	
Actor	Sistema
	1. El sistema muestra las consultas creadas y un desplegable con cada una de ellas.
2. El usuario selecciona la consulta que desea modificar mediante en desplegable y pulsa el botón de “Modificar”:	
	3. El sistema muestra un listado con las posibles opciones de consulta y un desplegable con todos los municipios de la provincia seleccionada una vez creada la consulta.
4. El usuario selecciona las opciones y el municipio que desea y pulsa el botón de “Modificar consulta”.	
	5. El sistema actualiza los cambios y regresa a la pantalla principal donde se pueden apreciar dichos cambios.
<b>Extensiones:</b>	
4a. El sistema detecta que algún campo está vacío.	
<ul style="list-style-type: none"> <li>- 4. a1) El sistema muestra un aviso indicando que hay un campo que no está relleno.</li> <li>- 4. a2) Vuelve al punto 2.</li> </ul>	

**Tabla 6. Modificar Consulta**

- **Generar URL consulta**

<b>Caso de uso:</b> Generar URL Consulta	
<b>Actor principal:</b> Usuario registrado	
<b>Precondición:</b> El usuario ha iniciado la sesión	
<b>Disparador:</b> El usuario quiere generar URL de la consulta	
<b>Escenario principal de éxito:</b>	
Actor	Sistema

	1. El sistema muestra las consultas creadas y un desplegable con cada una de ellas.
2. El usuario selecciona la consulta de la quiere saber su URL y pulsa el botón de “Generar URL”:	
	3. El sistema muestra URL de la consulta.
<b>Extensiones: -</b>	

Tabla 7. Generar URL Consulta

- **Eliminar consulta**

<b>Caso de uso:</b> Eliminar Consulta	
<b>Actor principal:</b> Usuario registrado	
<b>Precondición:</b> El usuario ha iniciado la sesión	
<b>Disparador:</b> El usuario quiere eliminar una consulta	
<b>Escenario principal de éxito:</b>	
<b>Actor</b>	<b>Sistema</b>
	1. El sistema muestra las consultas creadas y un desplegable con cada una de ellas.
2. El usuario selecciona la consulta que desea eliminar mediante en desplegable y pulsa el botón de “Eliminar”:	
	3. El sistema actualiza los cambios.
<b>Extensiones: -</b>	

Tabla 8. Eliminar Consulta

### 3.6 Diagramas de secuencia

Un diagrama de secuencia es un tipo de diagrama de interacción que describe el comportamiento dinámico de un sistema de información haciendo hincapié en la secuencia de los mensajes intercambiados por los objetos.

Los diagramas de secuencia realizados son los siguientes:

### Inicio de sesión

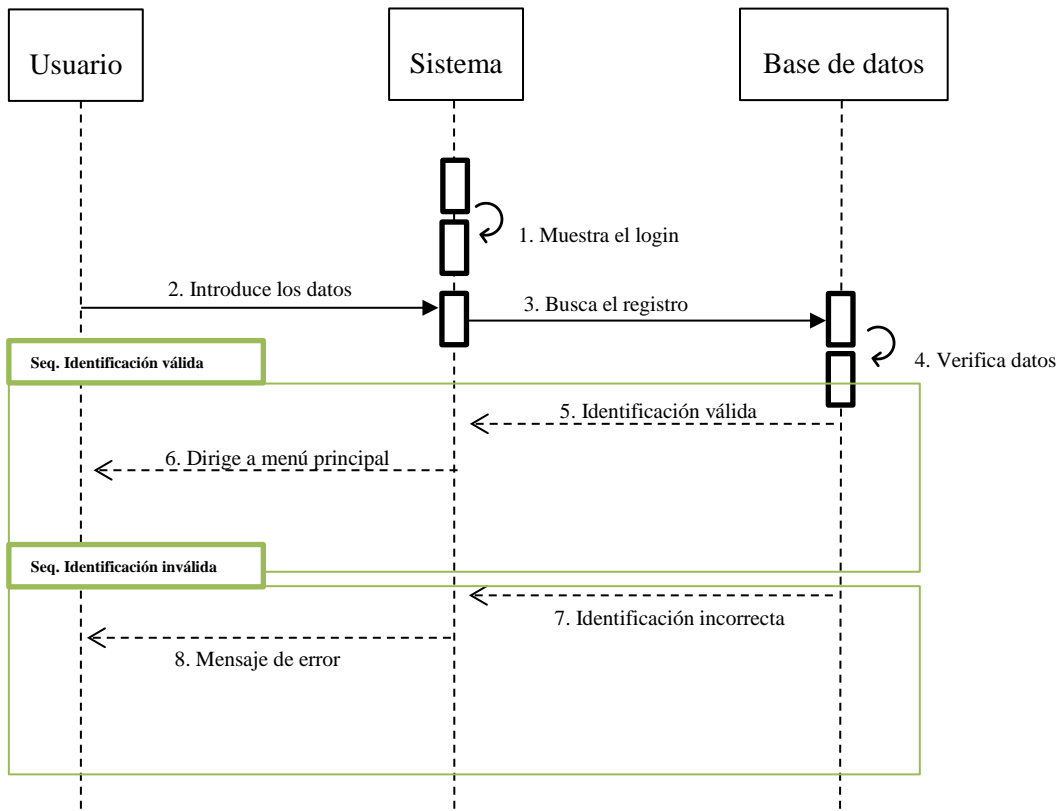


Ilustración 21. Diagrama de secuencia de inicio de sesión

## Registrar cliente

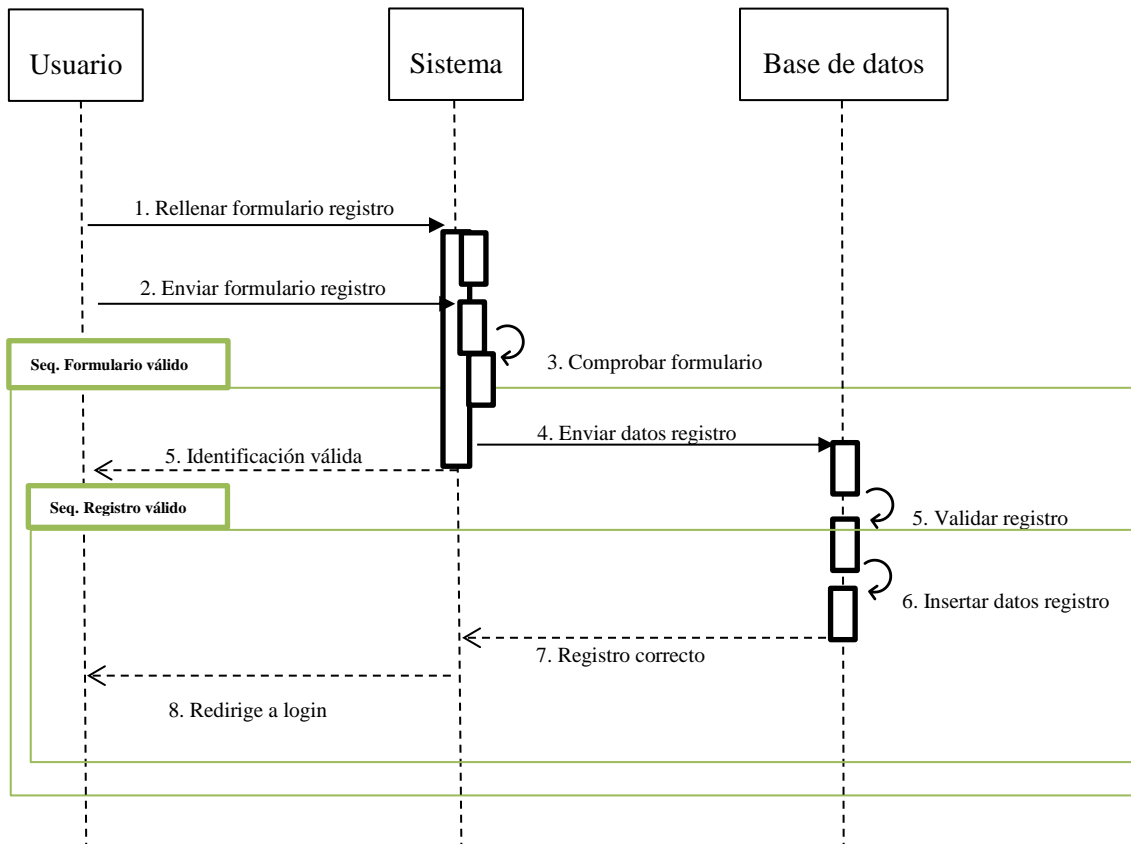


Ilustración 22. Diagrama de secuencia de registrar cliente

### Crear Consulta

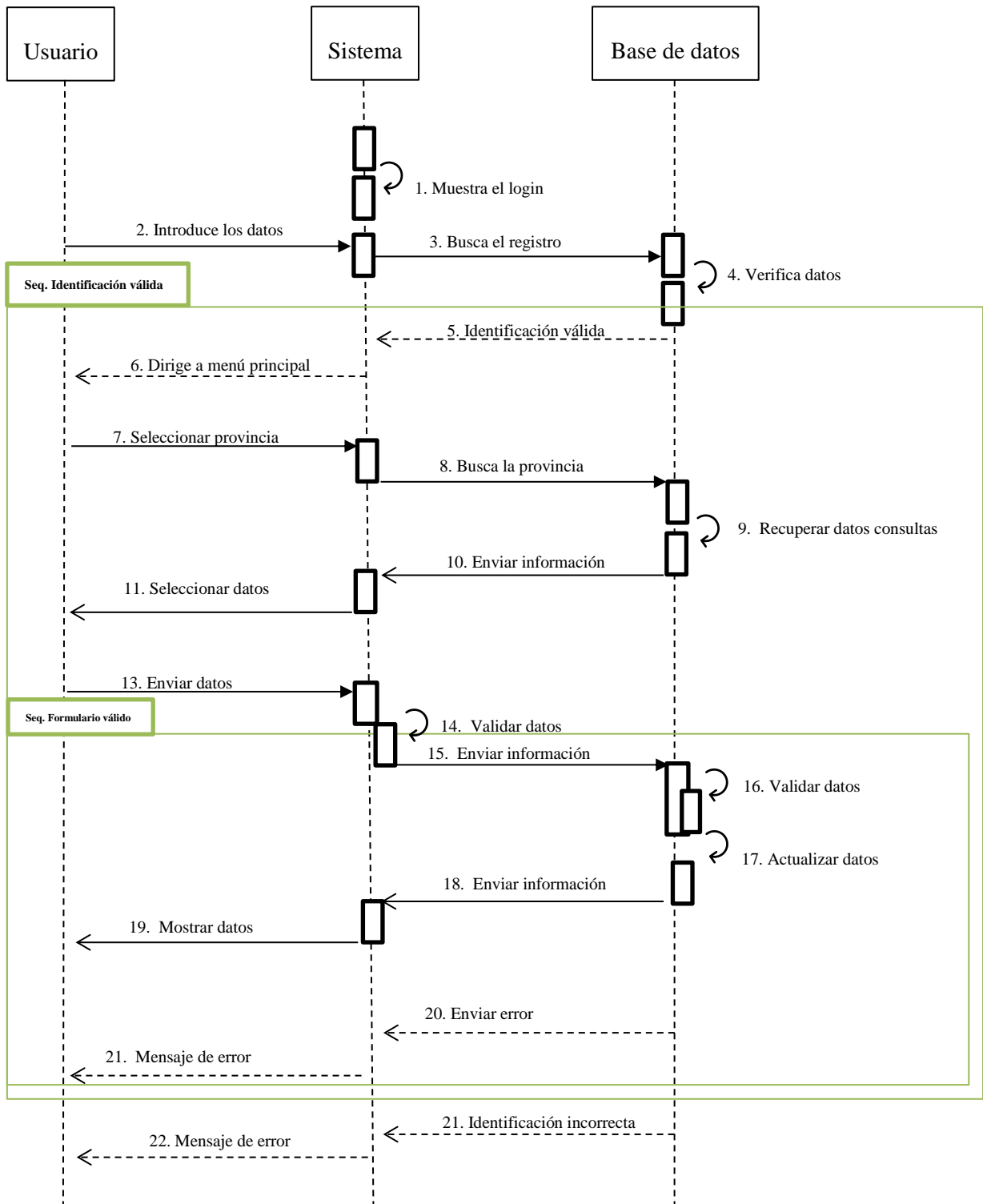


Ilustración 23. Diagrama de secuencia de crear consulta

### Visualizar Consulta

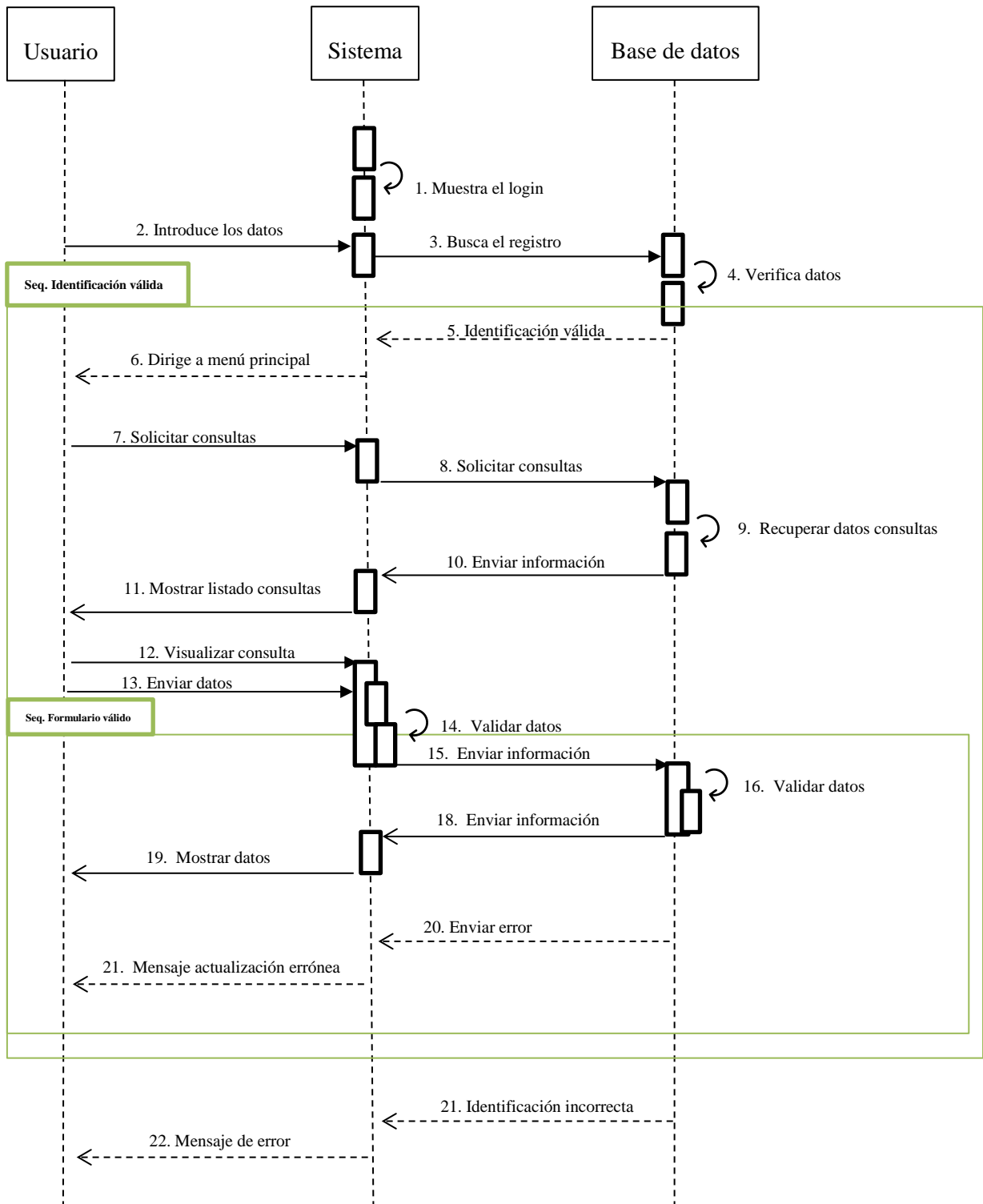


Ilustración 24. Diagrama de secuencia de visualizar consulta

### Modificar Consulta

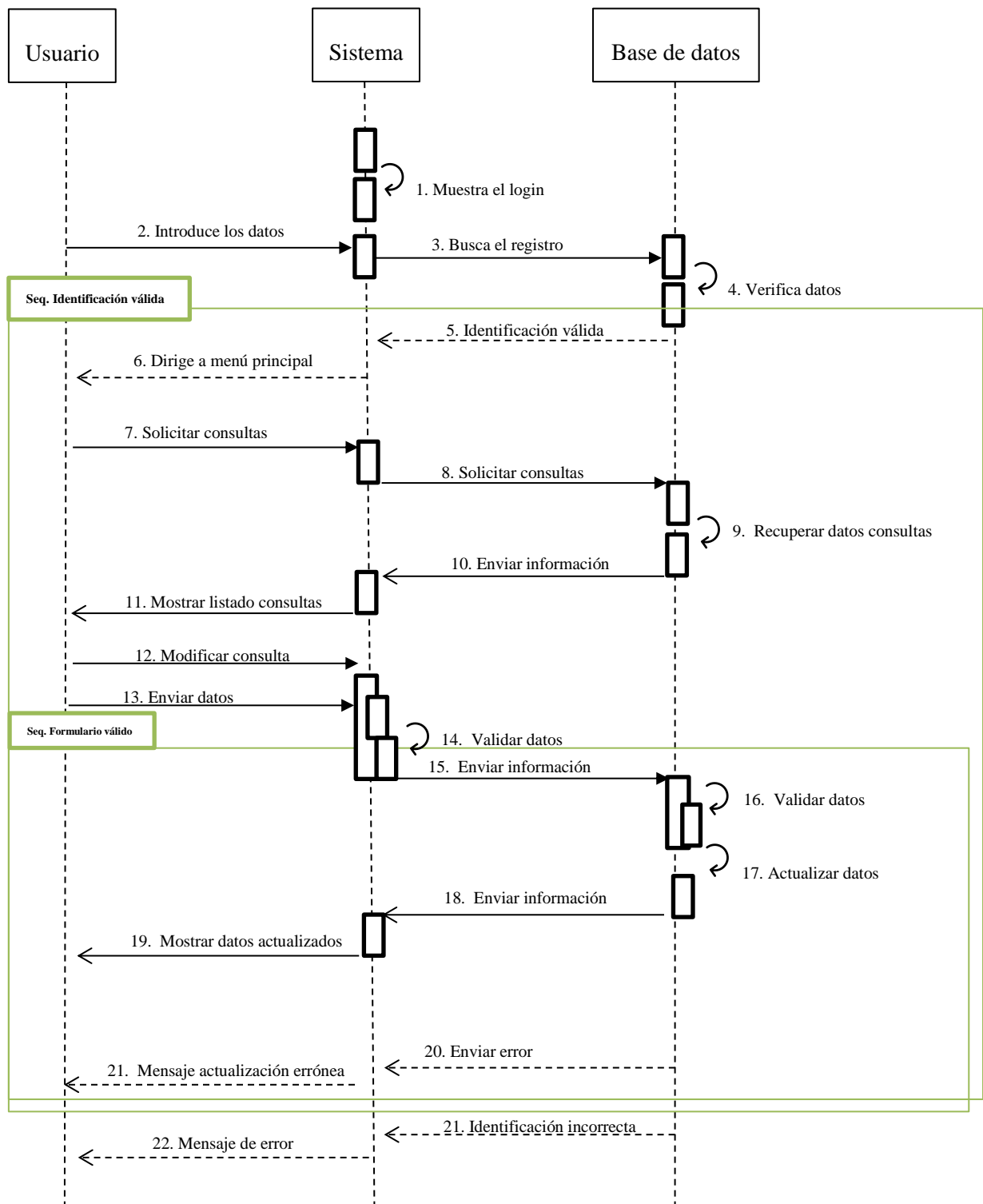


Ilustración 25. Diagrama de secuencia de modificar consulta

### Generar URL Consulta

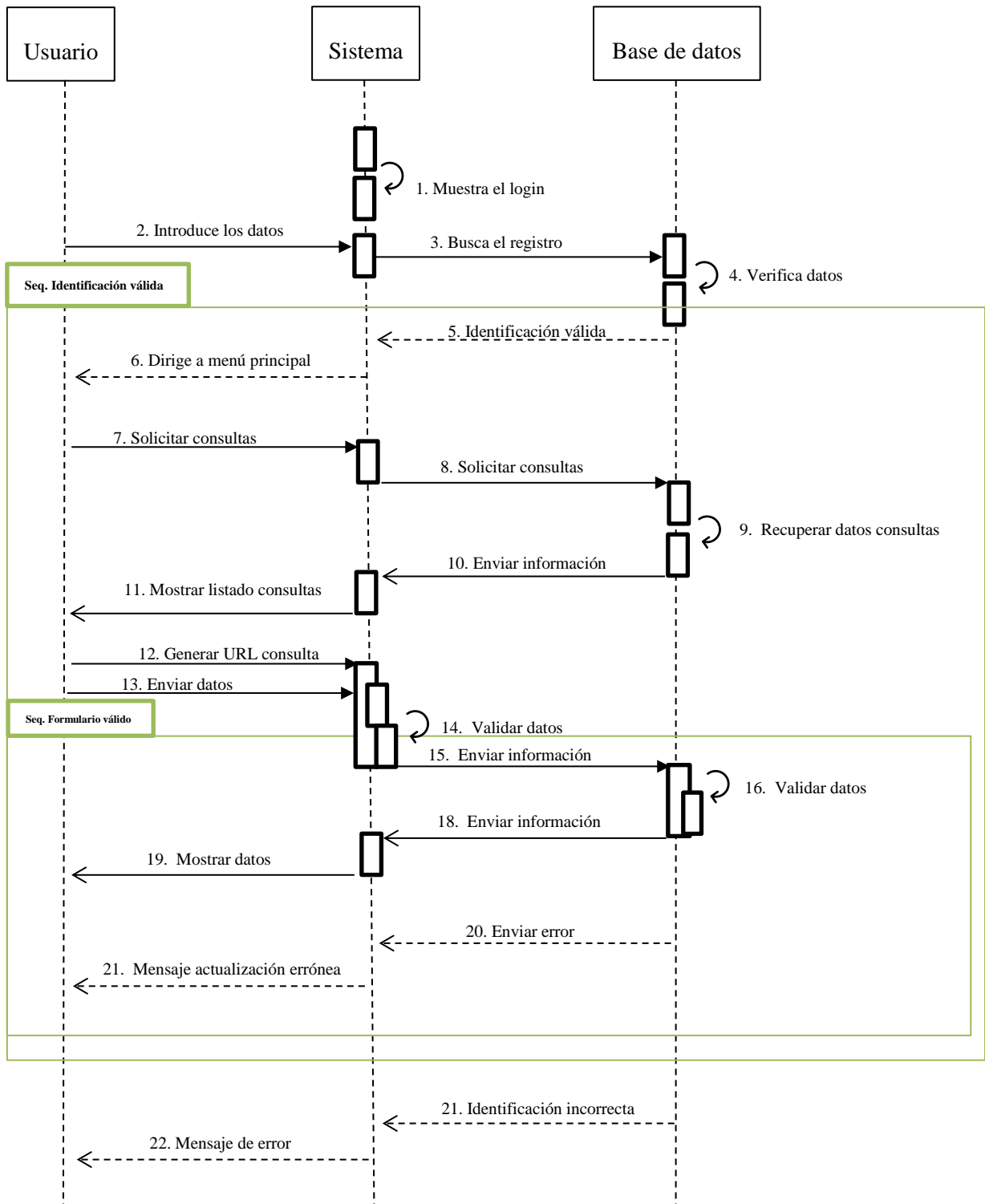


Ilustración 26. Diagrama de secuencia de generar URL consulta



## Eliminar Consulta

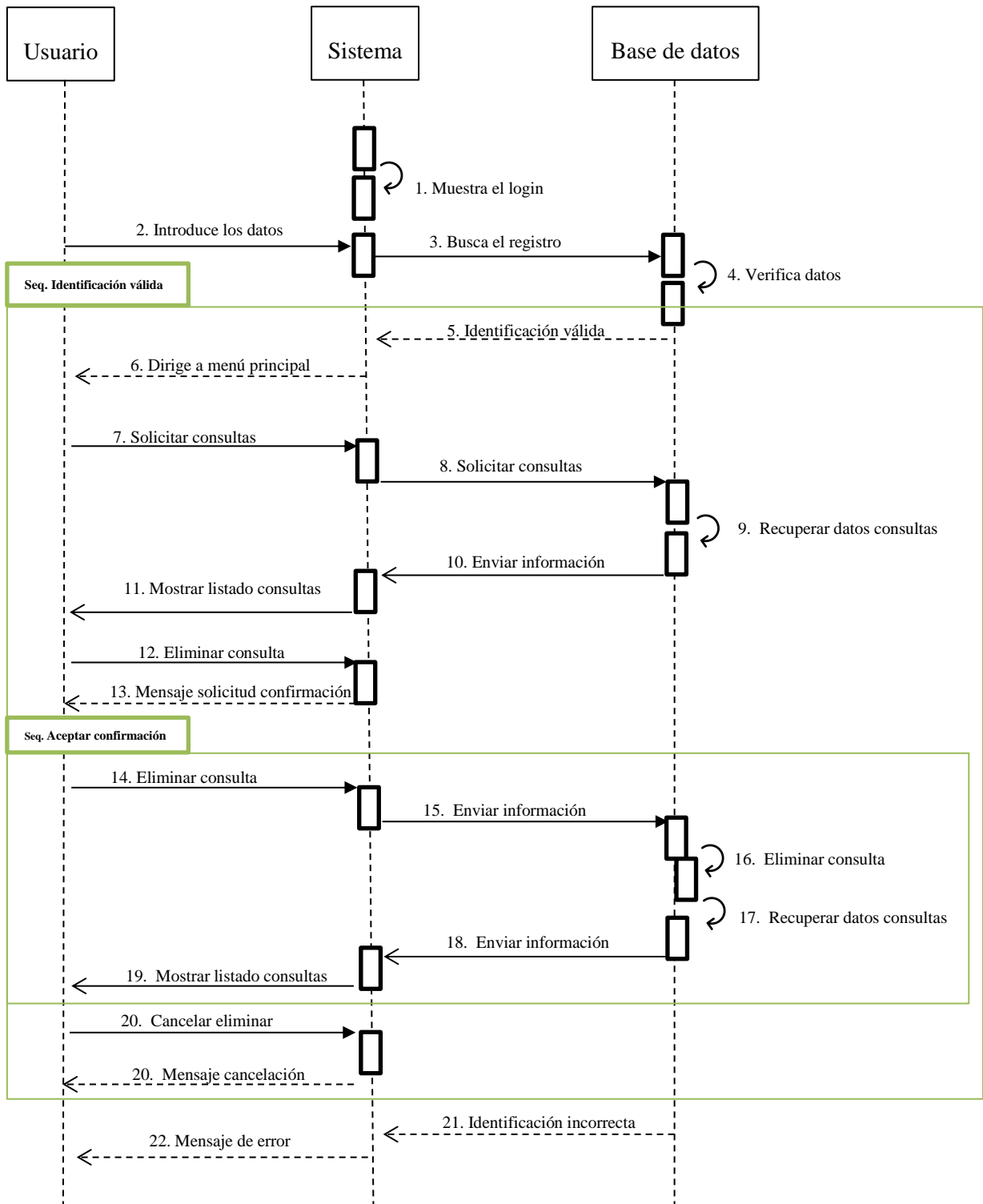


Ilustración 27. Diagrama de secuencia de eliminar consulta

## Capítulo 4. Diseño

### 4.1 Modelo relacional

El modelo relacional es modelo de datos que se basa en la lógica de predicados y en la teoría de conjuntos. La idea principal es la del uso de relaciones., es decir una tabla bidimensional constituida por líneas (tuplas) y columnas (atributos).

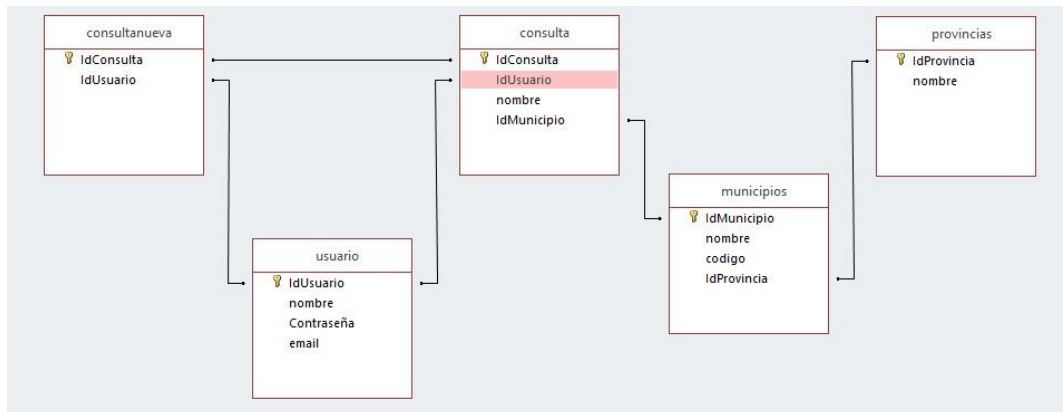


Ilustración 28. Modelo relacional

Las relaciones de los atributos entre tablas son las siguientes:

- ❖ **usuario**: clave primaria IdUsuario y tiene relación con las tablas:
  - consultanueva
  - consultasiendo la clave foránea en cada una de ellas.
- ❖ **consultanueva**: clave primaria IdConsulta y tiene relación con las tablas:
  - consultasiendo la clave foránea en cada una de ellas.
- ❖ **consultas**: no tiene clave primaria.
- ❖ **municipios**: clave primaria IdMunicipio y tiene relación con la tabla:
  - consultasiendo la clave foránea en cada una de ellas.
- ❖ **provincias**: clave primaria IdProvincia y tiene relación con la tabla:
  - municipiossiendo la clave foránea en cada una de ellas.

## 4.2 Diccionario de datos

Los diccionarios de datos son un conjunto de archivos que guarda información sobre los datos que pueden ser almacenados en la base de datos.

Nombre del campo	Atributos	Tipo de datos	Tamaño	Descripción	Restricción
consultanueva	IdConsulta	Enter	11	Identificativo consulta	Primary key
	IdUsuario	Enter	11	Identificativo usuario	Secondary Key

Tabla 9. Diccionario de datos consultanueva

Nombre del campo	Atributos	Tipo de datos	Tamaño	Descripción	Restricción
consultas	IdConsulta	Enter	11	Identificador de la consulta	Secondary key
	IdUsuario	Enter	11	Identificador del usuario	Secondary key
	Nombre	Varchar	50	intereses	No null
	IdMunicipio	Enter	11	Identificador de municipio	Secondary key

Tabla 10. Diccionario de datos consultas

Nombre del campo	Atributos	Tipo de datos	Tamaño	Descripción	Restricción
municipios	IdMunicipio	Enter	11	Identificador del municipio	Clave Primaria
	nombre	Varchar	50	Nombre del municipio	Not null
	codigo	Varchar	50	Código del municipio	Not null
	IdProvincia	Enter	11	Identificador de la provincia	Clave secundaria

Tabla 11. Diccionario de datos municipios

Nombre del campo	Atributos	Tipo de datos	Tamaño	Descripción	Restricción
provincias	IdProvincia	Enter	11	Identificativo provincia	Primary key
	nombre	Varchar	50	Nombre provincia	Not null

Tabla 12. Diccionario de datos provincias

Nombre del campo	Atributos	Tipo de datos	Tamaño	Descripción	Restricción
usuario	IdUsuario	Enter	11	Identificativo usuario	Primary Key
	nombre	Varchar	50	Nombre del usuario	Not null
	Contraseña	Varchar	50	Contraseña usuario	Not null
	email	Varchar	50	Email usuario	Not null

Tabla 12. Diccionario de datos usuario

### 4.3 Arquitectura del sistema

La aplicación utiliza una arquitectura cliente-servidor. Esta arquitectura es un modelo de aplicación que consiente en que un cliente realiza peticiones a un servidor que le da respuestas.

Toda acción de visitar un portal web requiere esta arquitectura, ya que es el servidor web s quien sirve las paginas web al navegador.

Para entender un mejor funcionamiento de la arquitectura tenemos las siguientes funciones que desempeñan tanto el cliente como el servidor:

El cliente:

- Es quien inicia solicitudes o peticiones, tiene por tanto un papel activo en la comunicación.
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa con los usuarios a través de la interfaz gráfica de usuario.

El servidor:

- Al iniciar espera las solicitudes del cliente.

- Tras la recepción de una solicitud, la procesa y envía la respuesta al cliente.
- Acepta un gran número de conexiones de clientes.

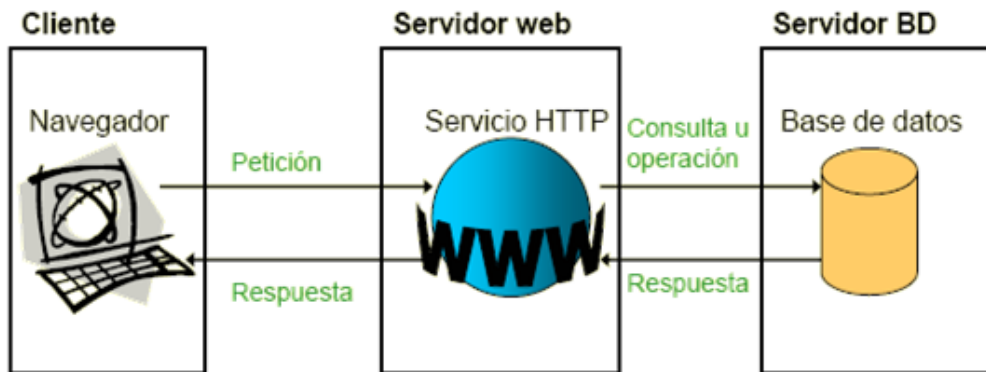


Ilustración 29. Arquitectura cliente-servidor

La división de aplicaciones está formada por las siguientes capas:

- Capa de cliente
- Capa de presentación
- Capa de servicios de negocios
- Capa de datos

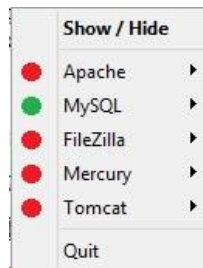
Explicadas cada una de ellas a detalles:

- **Capa de cliente.** La capa de cliente formada por la lógica de la aplicación a la que el usuario final accede directamente mediante una interfaz de usuario.
- **Capa de presentación.** La capa de presentación formada por la lógica de aplicación, que prepara los datos para su envío a la capa de cliente y procesa las solicitudes desde la capa de cliente para su envío a la lógica de servicios de negocios del servidor.
- **Capa de servicios de negocios.** La capa de servicios de negocio consiste en procesamiento de datos, implementación de funciones de negocios, coordinación de varios usuarios y administración de recursos externos.
- **Capa de datos.** La capa de datos formada por los servicios que proporcionan los datos persistentes utilizados por la lógica de negocios.

## Capítulo 5. Pruebas realizadas

Algunos detalles a tener en cuenta son:

- Para que la aplicación se ejecute correctamente lo primero que tenemos que hacer es iniciar MySQL de Xampp. Para ello hay que abrir Xampp y en la opción de MySQL pulsar el botón de start. Una vez pulsado el botón, aparecerá un led verde como se puede apreciar en la foto, que indica que MySQL esta activo.



- Tanto la aplicación web como la base de datos tienen un cotejamiento de caracteres iso-8859-1.
- El usuario de la base de datos debe ser root y la contraseña debe dejarse en blanco. Por seguridad sería recomendable tener una contraseña que proteja la base de datos.
- Todos los formularios tienen que estar rellenos con los datos adecuados. De lo contrario aparecerán mensajes de error indicando la causa del error.
- A la hora de eliminar consultas, modificar consultas, registrar o hacer login a un usuario, se hacen comprobaciones para ver la existencia de los mismos, en caso de que haya errores, aparecerán unos mensajes de error.

## Capítulo 6. Resultados del trabajo

### Inicio de sesión

Para comenzar hay que escribir en el navegador de Internet elegido: localhost/ProyectoFinal/. La aplicación solo es visible desde localhost porque no se ha subido a ningún servidor.

Para poder acceder a la aplicación debemos estar registrados.

En la página principal, disponemos de dos formularios. El primero es para usuarios ya registrados, en los que para acceder hay que utilizar el email y el password utilizados en el registro previo. El segundo es para usuarios que todavía no están registrados en el sistema. Para ello tienen que registrarse poniendo un nombre, una dirección de correo electrónico, un password y una verificación de password.

The screenshot shows the main page of the application. On the left is a vertical banner with the 'AEMet' logo and the text 'Área Especial de Metodos'. On the right, there are two forms. The first is the 'Login' form, which has an 'Email' field containing 'moni94g' and a 'Pass' field with four dots. Below these fields is a blue 'Aceptar' button. The second form is the 'Registro' (Registration) form, which has four input fields: 'Ingresar tu nombre', 'Ingresar tu email', 'Ingresar tu Password', and 'Repite tu contraseña'. Below these fields is a blue 'Registrarse' button.

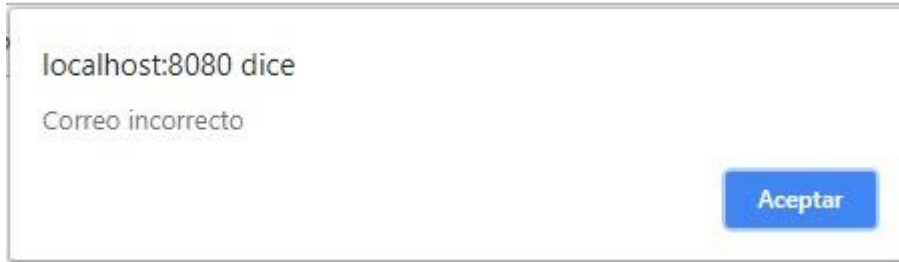
Ilustración 30. Página principal

Tanto en el registro como en el login todos los campos de los formularios deben estar rellenos, de no ser así, aparecerá un mensaje de error.

En caso de error en login porque la contraseña o el correo no es el correcto se deniega el acceso.

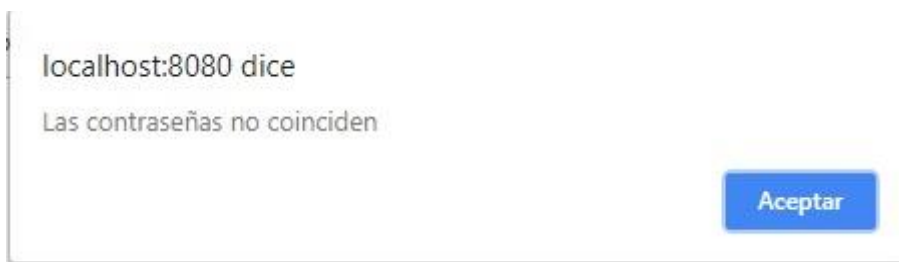
The screenshot shows a white error message dialog box with a thin border. The text inside reads 'localhost:8080 dice' followed by 'Contraseña incorrecta' on the next line. In the bottom right corner of the dialog box is a blue button with the text 'Aceptar'.

Ilustración 31. Mensaje de error



**Ilustración 32. Mensaje de error**

En caso de error en Registro porque las contraseñas no coinciden se deniega el acceso.



**Ilustración 33. Mensaje de error**

Si los datos introducidos ya existen se deniega el acceso mediante un mensaje de error y se redirige a la pantalla de registro.



**Ilustración 34. Mensaje de error**

Si el usuario no existe, el usuario se introduce en la base de datos y se mostrará un mensaje en el que informa de que el registro se ha realizado correctamente y se redirige a la pantalla de inicio.



**Ilustración 35. Mensaje usuario registrado**



## Menú

### Crear consulta

En el menú de la izquierda podemos ver nuestro perfil, un menú con las diferentes opciones a seleccionar y un botón para cerrar nuestra sesión. La imagen de usuario no se puede cambiar, es una imagen puesta por defecto.

Dentro de las diferentes opciones que ofrece el menú se pueden crear consultas, visualizarlas, modificarlas, eliminarlas y generar el URL de cada una de ellas, además de cambiar la contraseña.

Si el usuario todavía no ha creado ninguna consulta, ya que es un nuevo usuario en la aplicación o bien porque previamente ha borrado las consultas creadas, la aplicación le pedirá la creación de una consulta. Lo primero que deberá hacer es elegir la provincia sobre la cual quiere crear la consulta. Las provincias que elegir son las provincias de la comunidad valenciana.

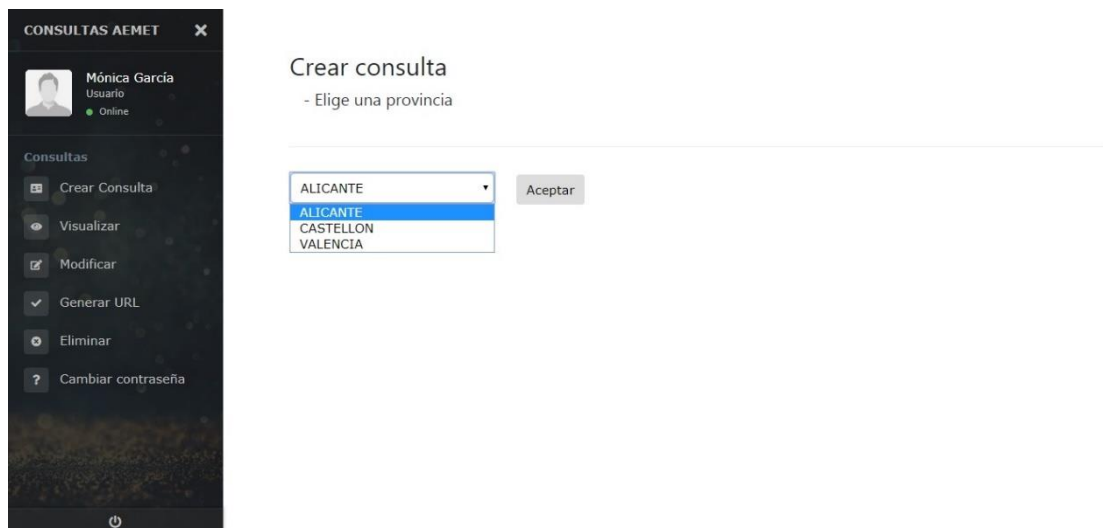


Ilustración 36. Crear consulta

Una vez seleccionada la provincia, la aplicación nos redirige a la página de crear consultas en la que hay que seleccionar los campos de interés y el municipio sobre los que deseamos hacer la consulta.

CONSULTAS AEMET

Mónica García  
Usuario  
Online

Consultas

- Crear Consulta
- Visualizar
- Modificar
- Generar URL
- Eliminar
- Cambiar contraseña

Crear Consulta

- Debes seleccionar mínimo un campo

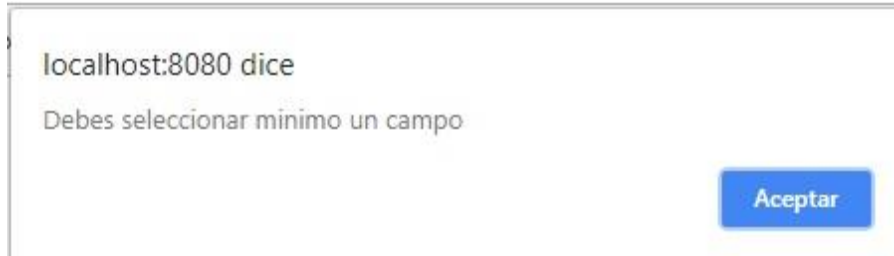
- Precipitación en mm
- Prob. de precipitación en %
- Prob. de tormenta en %
- Nieve en mm
- Prob. de nieve en %
- Dirección y velocidad del viento
- Sensación térmica en °
- Temperatura en °C
- Humedad relativa en %

València

Crear consulta

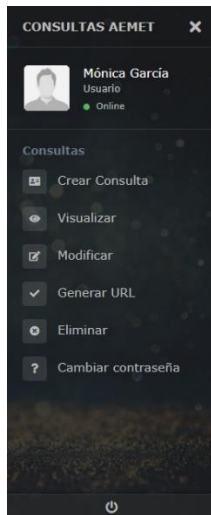
**Ilustración 37. Crear consulta**

Si por error no se selecciona ningún campo de interés, la aplicación muestra un mensaje de error en el que nos dirá que mínimo debemos seleccionar un campo.



**Ilustración 38. Mensaje de error**

Una vez seleccionado el botón de “Crear consulta”, la aplicación nos mostrará la página principal en la que podremos ver el listado de las consultas creadas por nosotros.



## Listado de consultas

Consultas

### Consulta 1

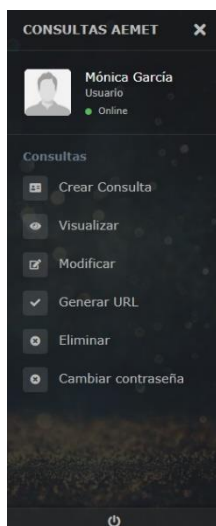
València, VALENCIA

- Precipitación en mm
- Temperatura en °C
- Sensación térmica en °

Ilustración 38. Listado de consultas

## Visualizar

En este apartado podremos visualizar la consulta creada previamente. Basta seleccionar la consulta que deseamos visualizar y hacer clic en el botón “Visualizar consulta” y así poder ver a detalle la consulta como se puede observar en la fotografía.



## Listado de consultas

Visualizar consulta

### Consulta 1

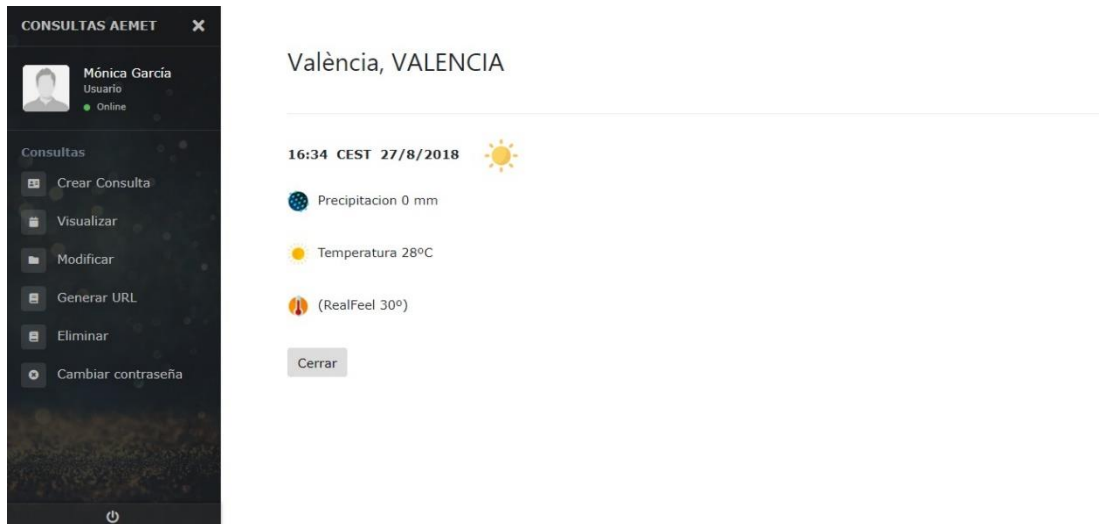
València, VALENCIA

- Precipitación en mm
- Temperatura en °C
- Sensación térmica en °

Consulta 1

Visualizar consulta

Ilustración 39. Visualizar consulta



**Ilustración 40. Visualizar consulta**

## Modificar consulta

Para modificar una consulta se elige la consulta que queremos modificar. Una vez elegida, se deben seleccionar los nuevos campos de interés que se desean respecto al municipio y la provincia ya elegidos previamente en la creación de la consulta.



**Ilustración 41. Modificar consulta**



**Ilustración 42. Modificar consulta**

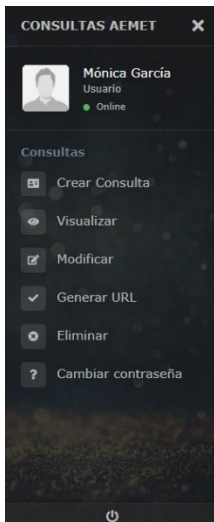
Al hacer click en el botón de “Modificar Consulta”, se lanza un mensaje para verificar que queremos modificar la consulta o no.



**Ilustración 43. Mensaje de confirmación en modificar consulta**

## Generar URL

Para generar un URL de una consulta se elige la consulta de la que se quiere obtener. Una vez seleccionada se mostrará una página con el URL de esta.



URL

<http://localhost:8080/ProyectoFinal/visualizar1?idConsulta=21>

Cerrar

**Ilustración 44. Generar URL**

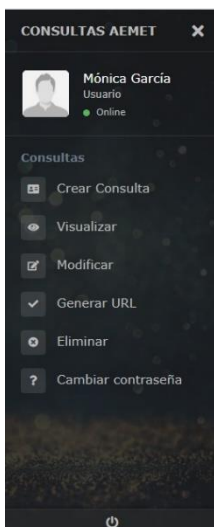
Si copiamos el URL generado en cualquier navegador podemos observar el resultado en la Ilustración 41.



**Ilustración 45. URL en el navegador**

## Eliminar consulta

Para eliminar una consulta hay que elegir la consulta que queremos eliminar.



Listado de consultas

Eliminar consulta

**Consulta 1**

València, VALENCIA

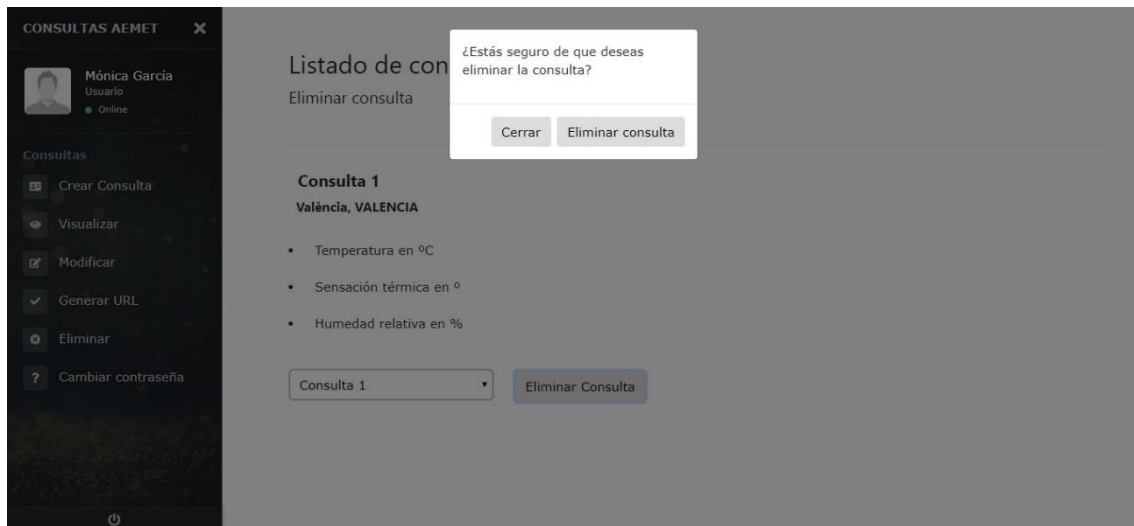
- Temperatura en °C
- Sensación térmica en °
- Humedad relativa en %

Consulta 1

Eliminar Consulta

**Ilustración 46. Eliminar consulta**

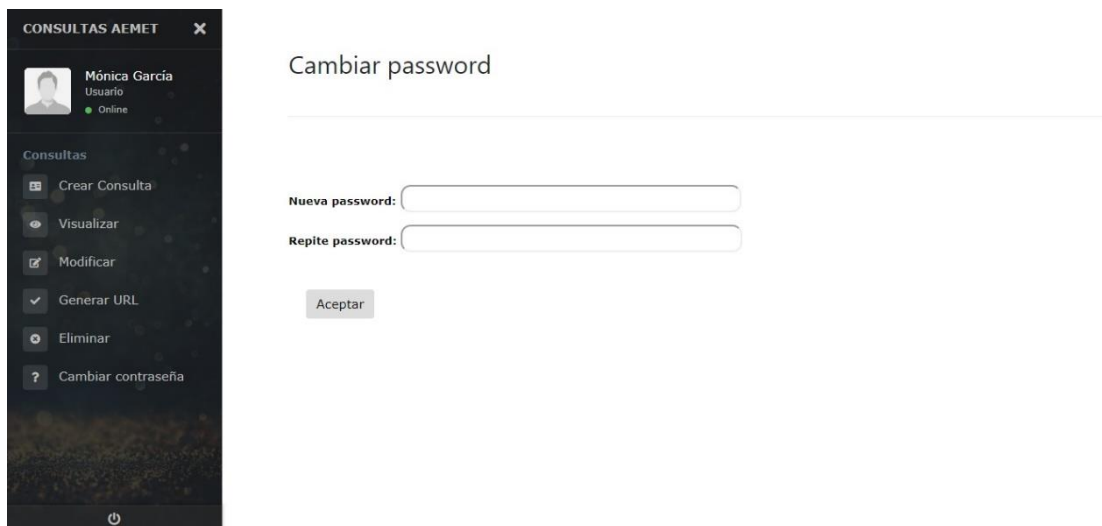
Al hacer click en el botón de “Eliminar Consulta”, se lanza un mensaje para verificar si queremos eliminar la consulta o no.



**Ilustración 47. Mensaje de confirmación eliminar consulta**

## Cambiar contraseña

Para cambiar la contraseña se tiene que introducir la nueva contraseña dos veces.



**Ilustración 48. Cambiar contraseña**

Al pulsar el botón de “Aceptar”, se lanza un mensaje para verificar si queremos cambiar la contraseña no.

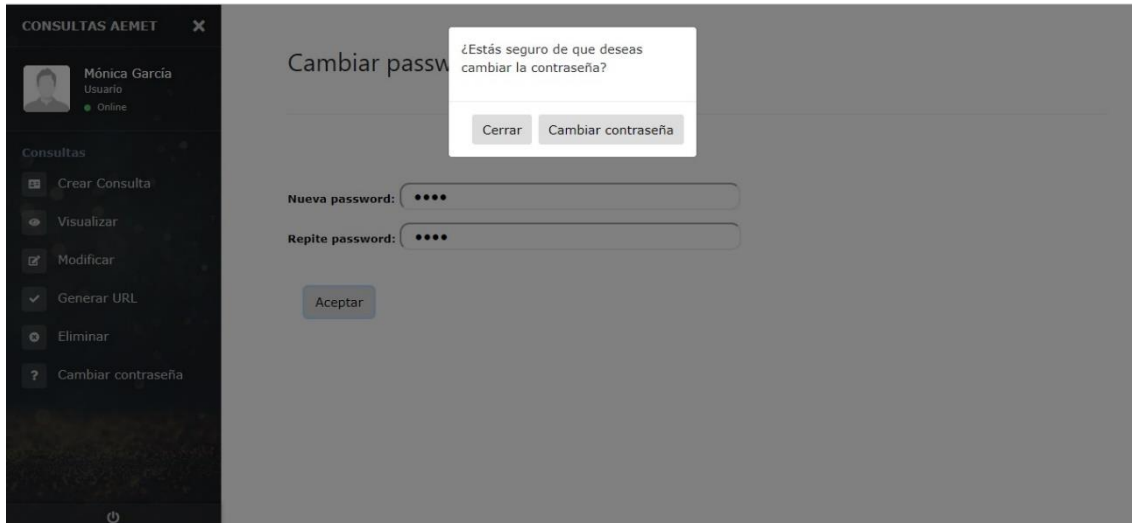


Ilustración 49. Mensaje de confirmación eliminar consulta

## URL

Si copiamos el URL que nos ha generado la consulta seleccionada en la aplicación y lo pegamos en una página web cualquiera obtenemos los siguientes resultados:

```
index.html
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163

<!-- Icon Cards -->
<div class="row">
  <div class="col-xl-3 col-sm-6 mb-3">
    <div class="card text-white bg-primary o-hidden h-100">
      <div class="card-body">
        <div class="card-body-icon">
          <i class="fas fa-fw fa-list"></i>
        </div>
        <iframe name="" frameborder="0" width="" height="" src="http://localhost:8080/ProyectoFinal/visualizar1?IdConsulta=21"></iframe>
      </div>
    </div>
  </div>
  <div class="col-xl-3 col-sm-6 mb-3">
    <div class="card text-white bg-warning o-hidden h-100">
      <div class="card-body">
        <div class="card-body-icon">
          <i class="fas fa-fw fa-comments"></i>
        </div>
        <div class="mr-5">11 New Tasks!</div>
      </div>
      <div class="card-footer text-white clearfix small z-1" href="#">
        <span class="float-left">View Details</span>
        <span class="float-right">
          <i class="fas fa-angle-right"></i>
        </span>
      </div>
    </div>
  </div>
</div>
```

Ilustración 50. Ejemplo utilización URL



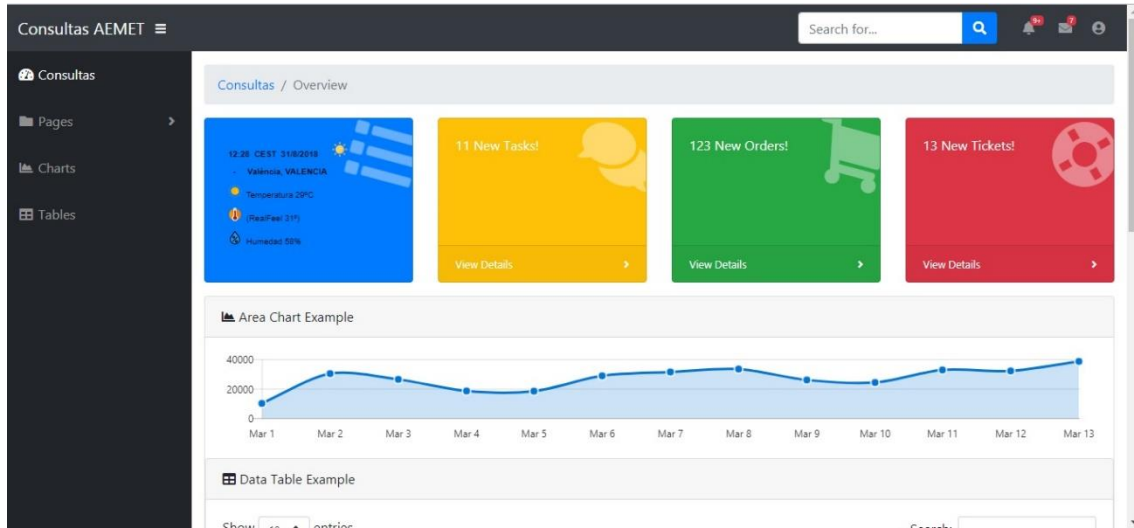


Ilustración 51. Resultados de URL generado



Ilustración 51. Resultados de URL generado



## Capítulo 7. Planificación y Presupuesto

### 7.1 Calendario

El proyecto se planificó para una duración estimada de aproximadamente tres meses. Los cuales comenzaron el 1 de junio de 2018 y acaban a principios de septiembre de 2018, fecha que corresponde con la defensa oral. No obstante, en la primera planificación inicial se consideró terminar el proyecto con un par de semanas de antelación, para así disponer tiempo suficiente para cubrir posibles contratiempos.

### 7.2 Recursos

Para poder llevar a cabo el proyecto, han sido necesarios tres tipos de recursos: personal, material y software.

#### 7.2.1 Recursos humanos

En este caso, es una persona la autora del proyecto, con una dedicación de 25/30 horas semanales y asumiendo todos los roles (analista, arquitecto, programador...) necesarios para la realización del proyecto.

#### 7.2.2 Recursos materiales

Un ordenador portátil Asus con sistema operativo Windows 8 y un lugar de trabajo donde poder realizar el proyecto con los recursos indirectos como la conexión a internet, electricidad, etc.

#### 7.2.3 Recursos software

NetBeans 8.0.2, HeidiSQL y Xampp.

Descritas con profundidad en el capítulo 4.

### 7.3 Descripción de tareas

Hay que tener en cuenta que este proyecto consta de una única desarrolladora y que las tareas han sido realizadas por una sola persona con un orden secuencial.

Las diferentes fases del proyecto son las siguientes:

#### 7.3.1 Fase inicial

Esta fase es una de las fases más importantes, ya que en ella se determinó lo que se iba a realizar en el proyecto y de qué manera se iba a llevar a cabo.

En esta fase inicial se realizó un estudio global sobre los diferentes aspectos del proyecto, como el contexto, los requisitos, los objetivos y el abasto, que permitió hacer una primera planificación y un coste estimado.

Esta fase tuvo una duración estimada de tres semanas y no tuvo dependencia de procedencia debido a que es la primera se realizada.

### 7.3.2 *Primera iteración, entorno de desarrollo y base de datos*

Una vez se tenía claro el modelo conceptual del proyecto, se configuró el entorno de desarrollo, instalando todas las aplicaciones que eran necesarias para el proyecto y se creó la base de datos que necesitábamos.

Esta fase tuvo la dependencia de procedencia de la fase inicial.

### 7.3.3 *Segunda iteración*

Esta segunda iteración tuvo como objetivo añadir las funcionalidades básicas a la aplicación. Estas funcionalidades eran: registrarse y acceder a la cuenta de usuario.

Esta fase tenía la dependencia de procedencia la iteración anterior ya que se necesitaba la base de datos para poder desarrollar la aplicación.

### 7.3.4 *Tercera iteración*

En la última iteración de desarrollo de la aplicación web se incorporaron las funcionalidades de poder crear, visualizar, modificar, generar URL y eliminar consultas. También se definió en esta iteración los servicios web REST que se implementan para obtener la información de las consultas realizadas explicados en el capítulo 1 y 3.

Esta fase tuvo como dependencia de procedencia todas las iteraciones anteriores.

### 7.3.5 *Fase final*

Durante esta fase, se terminó de elaborar la memoria del proyecto, así como su presentación.

## 7.4 **Estimación del tiempo**

Fase	Dedicación en horas
Fase inicial	40
Primera iteración	30
Segunda iteración	80
Tercera iteración	115
Fase final	90
<b>Total</b>	<b>355</b>

**Tabla 13. Dedicación proyecto**

Una vez se planificó el proyecto fue necesario realizar un estudio económico para determinar la viabilidad del proyecto. Para ello se tuvo que identificar y estimar los diversos costes que fueron necesarios para la realización de este proyecto.

## 7.5 Estimación de costes

Los costes se descomponen en, costes de recursos humanos, costes de recursos hardware, costes de recursos software y costes indirectos. Además, hay que tener en cuenta los posibles costes de imprevisto.

### 7.5.1 Recursos humanos

En este apartado se analizaron los costes de recursos humanos.

Este proyecto se ha llevado a cabo por una sola persona, así pues, esta persona actúa como jefe de proyecto, analista, arquitecto, programador y tester. En la siguiente tabla se muestra la remuneración asociada a cada role basada en el estudio de PagePersonnel del año 2017.

Rol	Remuneración en euros/hora
Jefe de proyecto	20
Analista	18
Arquitecto	18
Programador	12
Tester	15

Tabla 14. Remuneración en euros/hora

## 7.6 Presupuesto del proyecto

### 7.6.1 Recursos humanos

Utilizando la planificación del proyecto, se pudo obtener el coste de recursos humanos. Podemos observar en la siguiente tabla para cada tarea, la persona que la realiza y las horas que dedica.

Actividad	Horas	Recurso	Coste estimado
Fase inicial	40		
Contextualización y alcance	40	Jefe de proyecto	800 €
Planificación temporal			
Gestión económica y sostenibilidad			
<b>Primera iteración</b>	<b>30</b>		

Configuración del entorno de desarrollo	10	Programador	120 €
Creación de base de datos	20	Programador	200 €
<b>Iteración 2-3</b>	<b>195</b>		
Análisis y especificación de requisitos	20	Analista	360 €
Diseño	25	Arquitecto	450 €
Implementación	130	Programador	1560 €
Testing	20	Tester	300 €
<b>Fase final</b>	<b>90</b>		
Documentar y finalizar la memoria y preparación de la defensa oral	90	Jefe de proyecto	1800 €
<b>TOTAL</b>			<b>5590 €</b>

Tabla 15. Coste de recursos humanos

### 7.6.2 Recursos hardware

Para el desarrollo de la aplicación se utilizó un portátil F522LDV. Se hizo la suposición de que el portátil tiene una vida útil de 5 años, que un año tiene unos 250 días hábiles y que en cada día hábil se trabajan 8 horas.

Así, se calculó la amortización del portátil con la fórmula: amortización = precio total / vida útil \* tiempo de uso. Por tanto, el coste de amortización por hora es de  $650 \text{ €} / (4 \text{ años} * 250 \text{ días/años} * 8 \text{ h/día}) = 0.08125 \text{ €/h}$ .

Para las 355 horas de duración del proyecto, el coste estimado del portátil fue de: 28.8 €

Amortización por fase	Unidad	Precio	Coste
Fase inicial	40 h	0.08125 €/h	3.3 €
Primera iteración	30 h		2.4 €
Segunda iteración	80 h		6.5 €
Tercera iteración	115 h		9.3 €
Fase final	90 h		7.3 €
<b>Total</b>	<b>355 h</b>		<b>28.8 €</b>

Tabla 16. Coste de recursos hardware

### 7.6.3 Recursos de software

Todos los recursos software que se utilizaron en el desarrollo del proyecto son de uso gratuito.

#### 7.6.4 Costes indirectos

Los costes indirectos a tener en cuenta fueron:

- Acceso a Internet. Se utilizó una conexión de 30 Mb/s de subida y 27 Mb de bajada con 30 euros de cuota mensual.
- Consumo energético. El portátil que se utilizó tiene un consumo de 120W, por lo que supuso un precio de 0.15 €/kWh.
- Alquiler de oficina. El trabajo se realiza desde casa, pero se ha tenido en cuenta el arrendamiento para simular el coste del alquiler (500 € al mes).

Recurso	Unidad	Precio	Coste
Internet	3 meses	30 €/ mes	90 €
Consumo portátil	120 W *355 h	0.15 / kWh	6.39 €
Alquiler	3 meses	500 €/ mes	1500 €
<b>Total</b>			<b>1596.39 €</b>

Tabla 17. Estimación de costes indirectos

#### 7.6.5 Imprevistos

Los costes para imprevistos que se provinieron que pueden surgir durante el desarrollo del proyecto fueron los siguientes:

- **Avería del portátil:** En caso de posibles fallos de hardware se asumió que el coste previsto de reparación era de 100 € con una probabilidad del 5% de que ocurriese. Por lo tanto, había que añadir 5 € al presupuesto.
- **Retraso del proyecto:** Había una probabilidad del 25% que necesitase usar un plazo de dos semanas más. En este caso, se consideró unos 10 días laborales extras que corresponden a 40 horas de trabajo, y la remuneración media de los roles es 16.6€/h. Por lo tanto, el coste por retraso sería:  $0.25 * [16.6 \text{ (recursos humanos)} + 0.08125 \text{ (amortización portátil)} + 0.08125 * 0.15 \text{ (consumo portátil)} * 40 + 15 \text{ (Internet)} + 250 \text{ (alquiler)}] = 72.06 \text{ €}$

#### 7.6.6 Coste total

Concepto	Coste (€)
Recursos humanos	5590 €
Recursos hardware	28.8 €
Recursos software	0 €
Costes indirectos	1596.39 €
Imprevistos	72.06 €
<b>Coste total</b>	<b>7287.25 €</b>

Tabla 18. Estimación coste total



Por tanto, el presupuesto para el proyecto de 3 meses de duración y una sola persona fue de 8015.97 euros.

Se ha de tener en cuenta que los resultados obtenidos serían más correctos si el proyecto se encargase de manera profesional a una empresa para un desarrollo completo, incluyendo las posibles ampliaciones que pudiera tener.



## Capítulo 8. Futuras ampliaciones

Como ya se ha mencionado anteriormente, el proyecto que he desarrollado ha alcanzado los objetivos planteados inicialmente. Sin embargo, hay aspectos que pueden ser mejorados y existen funcionalidades que podrían añadirse a la aplicación ya que, debido a la limitación del tiempo, solo se han implementado las funcionalidades básicas.

Por un lado, se podría ampliar el listado de provincias y municipios para que todos los usuarios registrados tengan una más amplia selección para la creación de sus consultas.

Se podría añadir la opción de búsqueda por estaciones de todas las provincias de España e incluso se podría añadir cualquier dato meteorológico y climatológico de la Agencia, como por ejemplo la predicción específica de playas, predicción de índice de radiación UV máximo, la información nivológica para la zona montañosa, valores climatológicos de todas las estaciones, etc.

Por otro lado, puede darse el caso en el que se necesite restaurar la base de datos a una versión anterior. La aplicación podría contar con un formulario donde se pudiese elegir la copia de seguridad antigua deseada con formato .sql y volviera así a la versión elegida sin dificultad alguna.

Finalmente, en lo que respecta a la interfaz del sistema, sería útil la participación de un diseñador web para conseguir una visualización más atractiva, dado que la visualización de la aplicación es un poco pobre debido a que es el apartado en el que se ha dispuesto de menos tiempo.





## Capítulo 9. Conclusiones

El objetivo principal de este Trabajo de Fin de Grado se ha cumplido, se ha logrado crear una aplicación web que ofrece a un gran número de usuarios una nueva forma intuitiva y sencilla de acceso a los datos de Aemet Open Data sin necesidad de ningún conocimiento de programación.

Han sido unos meses de duro trabajo, pero ha merecido la pena. Ha supuesto un reto realizar esta aplicación ya que he aprendido a desarrollar paso a paso esta aplicación analizando los requisitos y sus funcionalidades, y además me ha servido para obtener unos mayores conocimientos de la programación en Java.

Personalmente, el proyecto me ha ayudado a poner en práctica todos los conocimientos obtenidos en las asignaturas del grado, especialmente la de ingeniería del software y la de sistemas telemáticos para la gestión de la información. También mencionar los nuevos conocimientos adquiridos como es de las tecnologías de los servicios web REST y del formato de texto ligero para intercambio de datos JSON.

Destacar la facilidad con la que hoy en día se encuentra información en internet y en libros de texto, ya que en ningún momento te puedes ver con falta de ayuda.

En definitiva, la aplicación resultante del proyecto es una herramienta para facilitar el acceso de los usuarios el acceso a Aemet OpenData, mediante una página web. Todos los objetivos establecidos inicialmente se han alcanzado con éxito y existe la posibilidad de extender las funcionalidades de la aplicación fácilmente.



## Capítulo 10. Bibliografía

[1] Aplicaciones Web

<https://www.internetya.co/ventajas-y-beneficios-de-las-aplicaciones-web/> [Online].

[https://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web) [Online].

<http://intranet.cip.org.pe/imagenes/temp/tesis/44554727.pdf> [Online].

<https://proyectodeluisnavarro.blogspot.com/> [Online].

[2] Netbeans, “NetBeans”

<https://www.genbeta.com/desarrollo/netbeans-1> [Online].

[3] GlassFish, “GlassFish”

<https://es.wikipedia.org/wiki/GlassFish> [Online].

[4] HeidiSQL

<https://es.wikipedia.org/wiki/HeidiSQL> [Online].

[5] Java, “Ventajas y desventajas de programar en JAVA”

<http://adictoalcodigo.blogspot.com/2016/07/ventajas-y-desventajas-de-programar-en.html> [Online].

[6] Servlet, “Ventajas de los Servlet”

<https://blogdeaitor.wordpress.com/2008/10/20/servlet-frente-a-jsp/> [Online].

[7] Servidores web, JSON

Practica 4 asignatura Aplicaciones telemáticas

[8] HTML, “The language for building web pages,”

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

[9] Servlets Java NetBeans, “Introducción a Java Servlets con NetBeans”,

<https://luauf.com/2008/05/21/introduccion-a-java-servlets-con-netbeans/> [Online].

[10] JSON en java, “Jugando con JSON en Java y la librería Gson”,

<https://www.adictosaltrabajo.com/tutoriales/gson-java-json/> [Online].

[11] “Servicios web RESTful con HTTP. Parte I: Introducción y bases teóricas”,



<http://www.adwe.es/general/colaboraciones/servicios-web-restful-con-http-parte-i-introduccion-y-bases-teoricas> [Online].

[12] Requisitos no funcionales, “Requisitos no funcionales”

[https://www.ecured.cu/Requisitos\\_no\\_funcionales](https://www.ecured.cu/Requisitos_no_funcionales) [Online].

[13] Diagrama de caso de uso, “Diagrama de caso de uso”

[https://es.wikipedia.org/wiki/Diagrama\\_de\\_casos\\_de\\_uso](https://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso) [Online].

[14] Diagrama de secuencia, “Diagrama de secuencia”

[https://es.wikipedia.org/wiki/Diagrama\\_de\\_secuencia](https://es.wikipedia.org/wiki/Diagrama_de_secuencia) [Online].

[15] Diagrama entidad-relación, “Diagrama entidad-relación”

[https://es.wikipedia.org/wiki/Modelo\\_entidad-relaci%C3%B3n](https://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n) [Online].

[16] Diccionario de datos, “Diccionario de datos”

<https://ingenieriadesoftwaretda.weebly.com/diccionario-de-datos.html> [Online].

[17] Arquitectura cliente-servidor, “Arquitectura cliente-servidor”

<https://es.wikipedia.org/wiki/Cliente-servidor> [Online].

<https://doc.oracle.com/> [Online].

[18] División de aplicación por capas, “División de aplicación por capas”

<https://docs.oracle.com/cd/E19528-01/820-0888/aaubb/index.html> [Online].

[19] Ejemplo de TFG

<http://docplayer.es/>

<https://uvadoc.uva.es/> [Online]