



Gestor de clientes y citas

Rocio Lozano de Sosa Palma  
ETIG / ETIS

Nombre Consultor

12/01/2015

## *Resumen*

---

El proyecto “Gestor de clientes y citas” engloba las funcionalidades básicas para determinados negocios que precisen de un repositorio con datos de sus clientes y un programa de citas.

Números negocios se adaptan al perfil de usuario de este proyecto, podría ser una peluquería, clínica de estética, masajes, consulta médica etc....

Para este proyecto en concreto se ha enfocado la aplicación hacia un gestor de clientes para una clínica de fisioterapia.

El acceso a la aplicación es a través de un navegador de Internet permitiendo el acceso desde cualquier dispositivo con internet ya sea una tablet, móvil, PC etc. Además se ha eliminado la dependencia de tener aplicaciones instaladas en los pc de los usuarios.

En el proyecto se contempla la ley Orgánica de protección de datos para aquella información perteneciente a los clientes. Los datos clínicos de los clientes se almacenan en tablas diferentes a los de cliente.

El objetivo de este proyecto es facilitar a las clínicas de fisioterapia la gestión de información de los clientes, poder asignar o eliminar citas a los clientes, así como disponer de un calendario con las citas programadas.

## Índice de contenidos

---

Resumen.....	2
Índice de contenidos.....	3
CAPITULO 1: Introducción.....	4
Justificación del TFC .....	4
Objetivo del TFC .....	4
Enfoque y método a seguir.....	4
Planificación del proyecto.....	5
Producto obtenido .....	5
Descripción del resto de capítulos .....	6
Descripción funcional.....	7
<b>Cientes</b> .....	<b>7</b>
<b>Usuarios</b> .....	<b>7</b>
<b>Citas</b> .....	<b>7</b>
Casos de usos .....	8
<b>Caso de uso: Ver listado de clientes</b> .....	<b>8</b>
<b>Caso de uso: Ver detalle de clientes</b> .....	<b>9</b>
<b>Caso de uso: Alta cliente</b> .....	<b>10</b>
<b>Caso de uso: Baja cliente</b> .....	<b>12</b>
<b>Caso de uso: Insertar documentos a un cliente</b> .....	<b>14</b>
<b>Caso de uso: Visualizar documentos a un cliente.</b> .....	<b>15</b>
<b>Caso de uso: Ver calendario</b> .....	<b>15</b>
<b>Caso de uso: Alta cita.</b> .....	<b>18</b>
<b>Caso de uso: Alta usuario</b> .....	<b>20</b>
<b>Caso de uso: Listado usuario</b> .....	<b>21</b>
<b>Caso de uso: Baja usuario.</b> .....	<b>22</b>
Funcionalidades futuras.....	23
Conclusiones.....	24
Glosario .....	25
Bibliografía .....	26
Anexo 1 .....	26
Manual de Instalación: .....	26
<b>1. Instalación de Java SE 6.0:</b> .....	<b>26</b>
<b>2. Instalación de Apache Tomcat:</b> .....	<b>27</b>
<b>3. Instalación de MySQL</b> .....	<b>27</b>
<b>4. Preparación de la base de datos</b> .....	<b>27</b>
<b>5. Copiamos el war en la ruta de instalación de Tomcat en el directorio "webapp"</b> .....	<b>29</b>
<b>6. Accedemos a <a href="http://localhost:8080/TFC">http://localhost:8080/TFC</a></b> .....	<b>29</b>
Anexo 2: Sentencias DDL.....	30

## *CAPITULO 1: Introducción*

---

### **Justificación del TFC**

Actualmente muchos negocios mantienen un sistema rudimentario para la gestión de citas de clientes. Solo las grandes empresas y organizaciones públicas han empezado a implementar sistemas en los cuales los propios usuarios son los que pueden pedir y anular citas por Internet.

Las pequeñas empresas suelen utilizar agendas físicas para gestionar las citas de los clientes y muchas de estas empresas ni siquiera disponen de un registro de clientes.

Se han identificado los siguientes sectores como usuarios potenciales de este sistema.

- Peluquerías
- Centros de belleza y estética
- Centros de fisioterapia
- Clínicas veterinarias
- Salones de celebraciones (boda, bautizos, comuniones)
- Oficios: Autónomos dedicados a servicios. (fontaneros, pintores...)

### **Objetivo del TFC**

El objetivo de este proyecto es dotar a una consulta de quiromasajes un sistema que permita tener un registro de los clientes y disponer de un calendario con las citas agendadas.

El listado de clientes permitirá a los usuarios tener registrado los datos del cliente para poder ofrecerle promociones, ya sea vía mail o vía teléfono. Además de tener todas las citas del paciente. De esta manera se podrá saber cuántas veces ha ido a la consulta un paciente y la evolución que ha tenido.

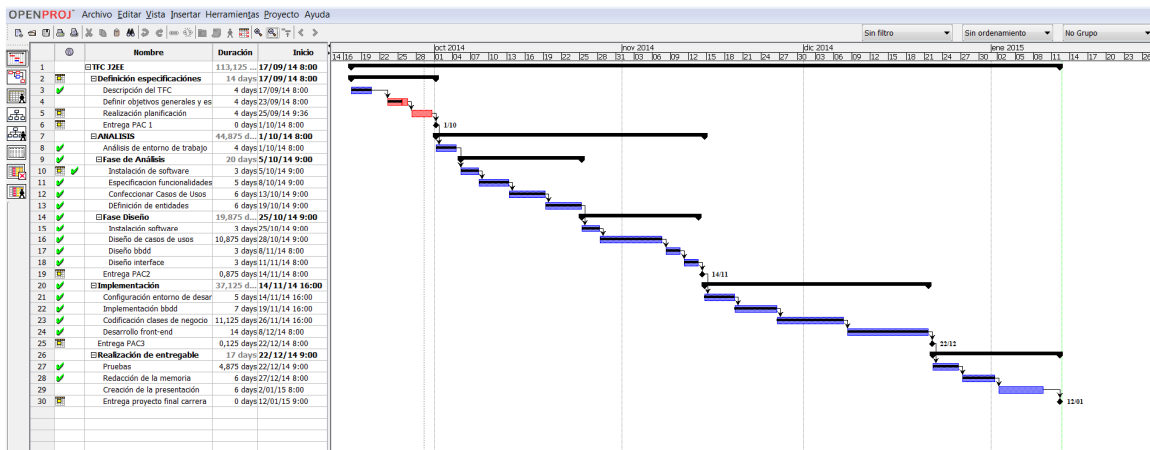
### **Enfoque y método a seguir**

La metodología a utilizar en el proyecto ha sido realizar los siguientes pasos básicos:

- ❖ Step 1:
  - Análisis de los requisitos pedidos
  - Análisis de los objetivos
  - Propuesta del proyecto y su planificación
- ❖ Step 2:
  - Desarrollo de la aplicación
- ❖ Step 3:
  - Pruebas de la aplicación y corrección de incidencias
- ❖ Step 4:
  - Finalización del proyecto
  - Realización de la memoria
  - Presentación del proyecto

## Planificación del proyecto

Considerando las tareas a realizar en las diferentes entregas (PAC) se ha trabajado en el proyecto con el siguiente calendario.



## Producto obtenido

El producto obtenido es un archivo, "GestorClientesyCitas.zip" que contiene la estructura del programa y el contenido requerido para su utilización, además de la información necesaria para entender su funcionamiento.

A continuación se detalla el contenido de los archivos más relevantes:

ARCHIVO	UBICACIÓN	INFORMACIÓN
web.xml	\\WebContent\\WEB-INF	Es el descriptor de despliegue de la aplicación y está contenido en el directorio WEB-INF. Define varios parámetros que son usados cuando se despliega la aplicación en el contenedor de Servlets/JSP Tomcat.
	\\WebContent\\WEB-INF\\lib	Librerías utilizadas por la aplicación
.classpath .project	Raíz	Archivos autogenerados Procedentes del IDE Eclipse.
*.class	build\\classes\\com\\rlozano\\tfc\\actions	Archivos java compilados

Por último también se incluye un ppt con la presentación del proyecto y un documento Word con la explicación del software necesario y los pasos a seguir para la instalación en local del servidor y la base de datos.

### Descripción del resto de capítulos

A continuación se detalla cada uno de los capítulos siguientes:

Capítulo 2: Se describe la funcionalidad de la aplicación

Capítulo 3: Se detallan los casos de uso del proyecto

Conclusiones: que recogen las reflexiones finales del autor con respecto al TFC

Glosario, con un índice de los términos utilizados en la memoria

Bibliografía: Apartado en el que se listan consultados para la realización del TFC.  
Anexos, con el código fuente más relevante de la aplicación.

## *Descripción funcional*

---

Este proyecto consiste en una aplicación web para una consulta de fisioterapia.

La funcionalidad de la aplicación se divide en tres grandes bloques: Clientes, Usuarios y Citas

### **Clientes:**

El módulo de clientes permitirá dar de alta nuevos clientes almacenando en una base de datos la siguiente información:

- Nombre
- Apellidos
- Dirección
- Email
- Teléfono
- Patología
- Evolución

Además se podrá realizar una gestión de estos clientes pudiendo modificarse de forma posterior a la creación así como la eliminación del cliente en la base de datos.

### **Usuarios:**

Solo para los administradores, se habilitará un módulo donde se podrá gestionar los usuarios de acceso a la aplicación. En principio solo se contemplan dos tipos de usuarios: Administradores y usuarios generales de la aplicación. En un futuro se contempla la posibilidad de habilitar una zona para los clientes los cuales puedan acceder y visualizar contenido, ofertas, etc...

### **Citas:**

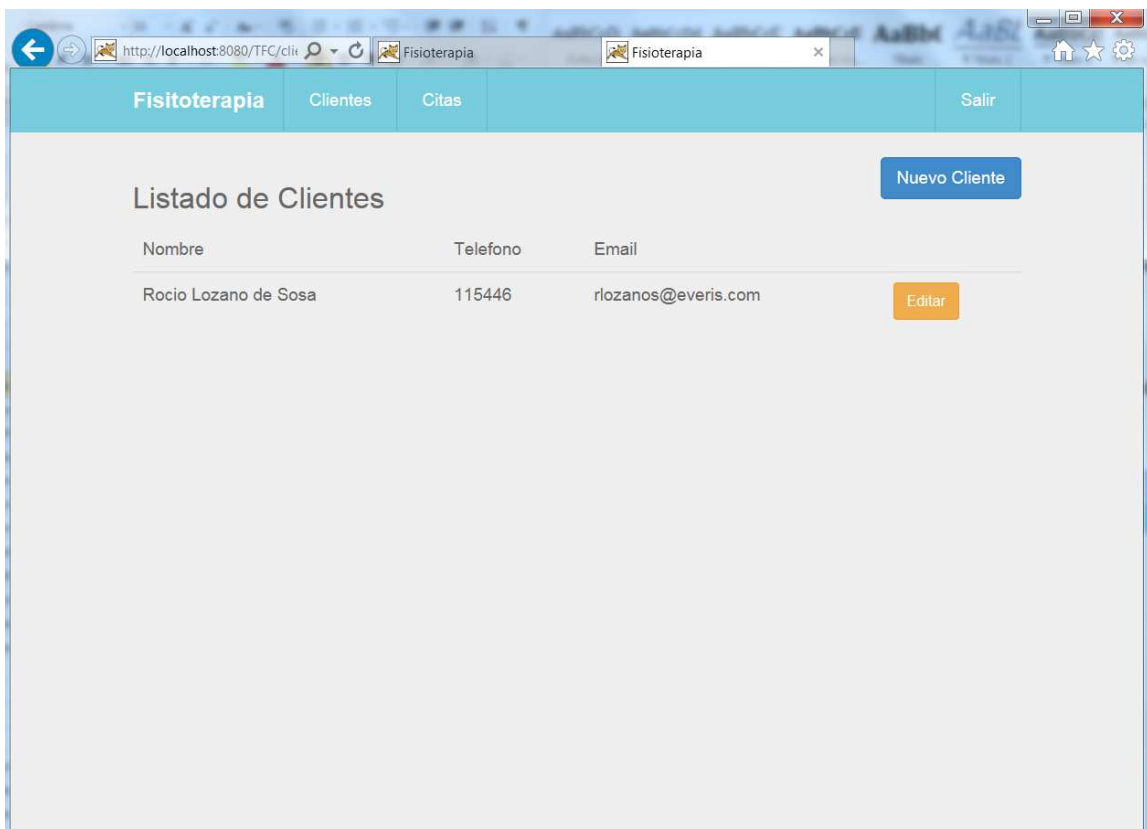
Cuando se visualice el detalle de cada cliente por la se visualizará también se podrán consultar las citas que tiene programadas o que ha tenido el cliente.

Desde el módulo de citas se podrá consultar el calendario de citas de forma mensual, diario, o semanal.

## Casos de usos

### Caso de uso: Ver listado de clientes

- 1.1. Actor principal: Administrador
- 1.2. Precondición: Estar logado
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Alta cliente, baja cliente y ver detalle de cliente
2. Escenario principal:
  - 2.1. El usuario se conecta a la web y se loga.
  - 2.2. El sistema muestra el listado de clientes dados de alta en él sistema.



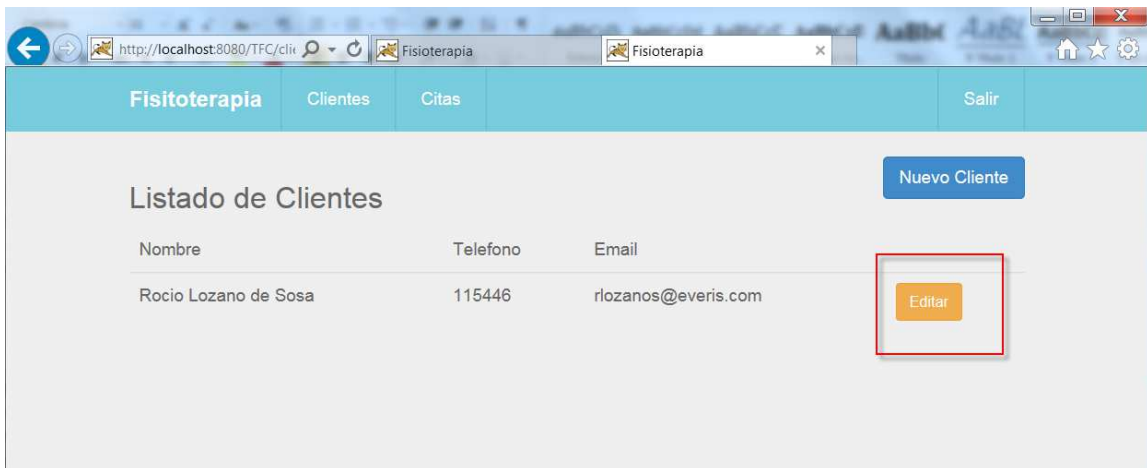


**Caso de uso:** Ver detalle de clientes

- 1.1. Actor principal: Administrador
- 1.2. Precondición: Estar logado
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Alta cliente, baja cliente

2. Escenario principal:

- 2.1. El usuario se conecta a la web y se loga.
- 2.2. El sistema muestra el listado de clientes dados de alta en el sistema
- 2.3 El usuario selecciona un cliente y se le muestra el detalle del cliente.



**Fisioterapia**   Cientes   Citas   Salir

**Editar Cliente**   Atras   Guardar Cliente   Eliminar Cliente

Nombre:    Apellidos:

Email:    Telefono:

Direccion:

Municipio:    Codigo Postal:    Provincia:

Patologia:

Evolucion:

**Citas:**

Inicio	Fin	Descripción
--------	-----	-------------

### Caso de uso: Alta cliente

- 1.1. Actor principal: Administrador
- 1.2. Precondición: Estar logado
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Listado cliente, baja cliente

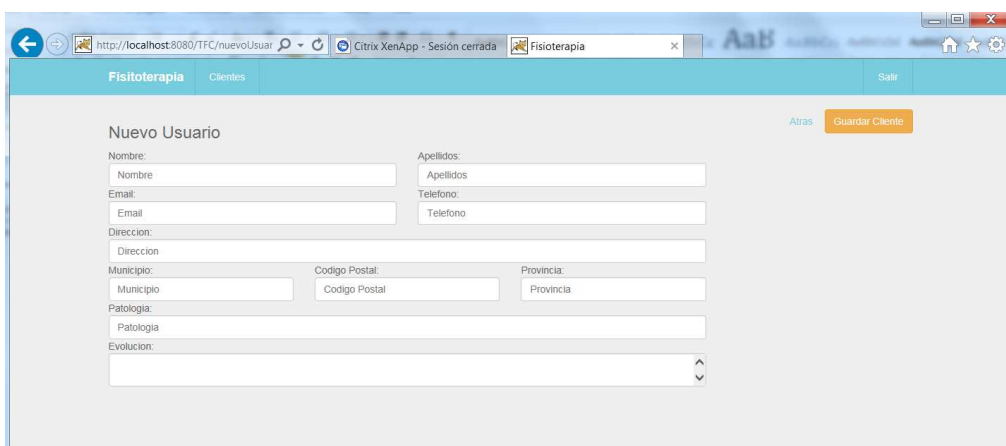
#### 2. Escenario principal:

- 2.1. El usuario se conecta a la web y se loga.
- 2.2. Desde la pantalla inicial el usuario podrá seleccionar con botón el alta de cliente
- 2.3. Se muestra una plantilla a rellenar por el usuario
- 2.4. Usuario informa los datos del cliente y acepta
- 2.5. El nuevo cliente se muestra por pantalla en la lista de clientes de la página principal.

A continuación se muestra la prueba de dicho caso.

Desde la pantalla de listado cliente se accede a la pantalla de alta usuario.

## Memoria.TFC-J2EE.UOC. 014/15 Gestor de clientes y citas (GCC)



The screenshot shows a web browser window with the URL `http://localhost:8080/TFC/nuevoUsuar`. The page title is "Fisioterapia" and the navigation menu includes "Fisioterapia", "Clientes", and "Salir". The main content area is titled "Nuevo Usuario" and contains the following form fields:

- Nombre:
- Apellidos:
- Email:
- Telefono:
- Direccion:
- Municipio:
- Codigo Postal:
- Provincia:
- Patologia:
- Evolucion:

Buttons: "Atras" and "Guardar Cliente".

Insertamos los datos del cliente:

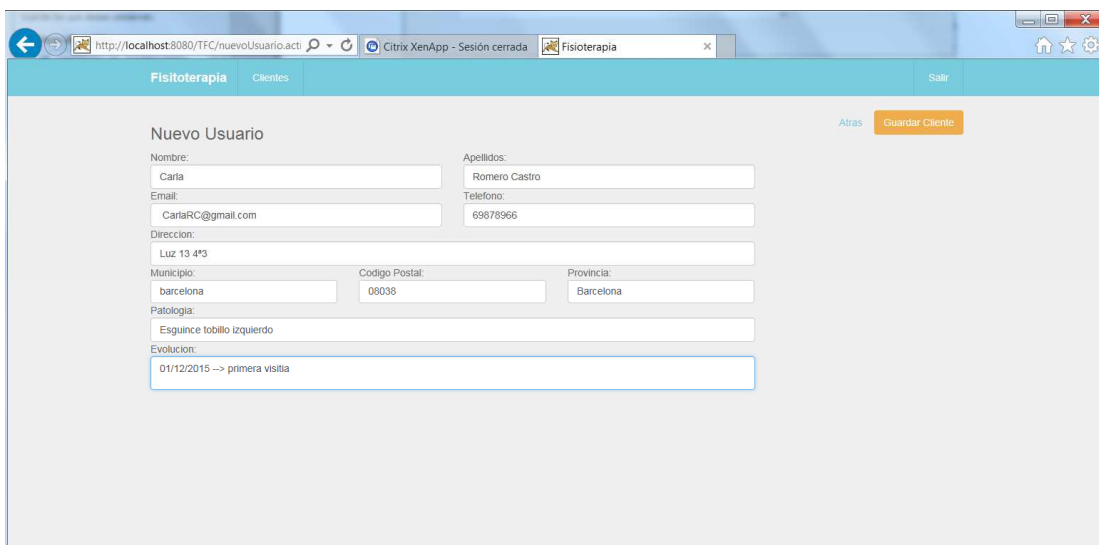
Nombre: Carla

Apellidos: Romero Castro

Email: [CarlaRC@gmail.com](mailto:CarlaRC@gmail.com)

Tlf: 69878966

....

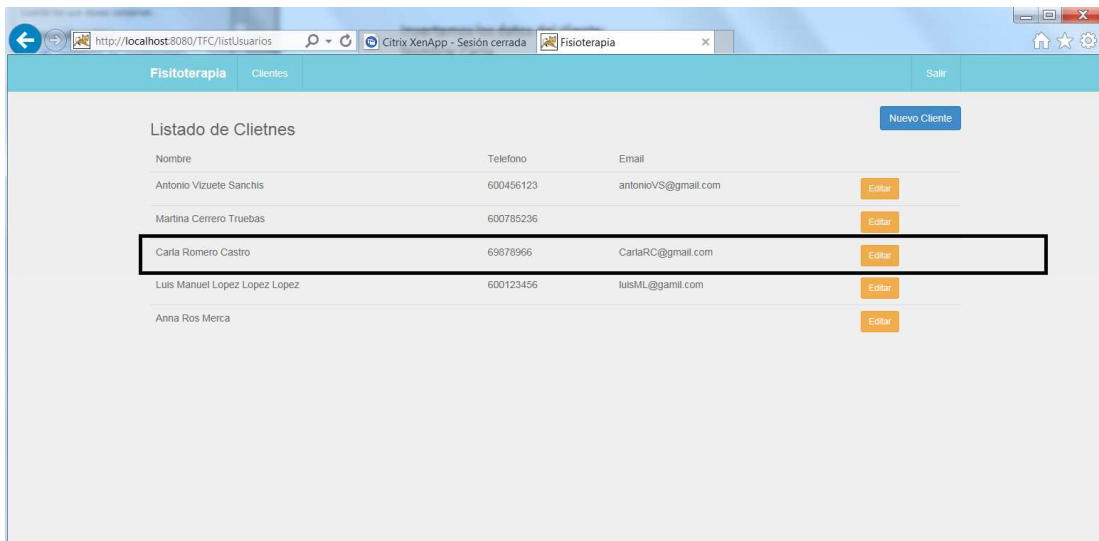


The screenshot shows the same web browser window, but the form is now filled with the client's data:

- Nombre:
- Apellidos:
- Email:
- Telefono:
- Direccion:
- Municipio:
- Codigo Postal:
- Provincia:
- Patologia:
- Evolucion:

Buttons: "Atras" and "Guardar Cliente".

Al seleccionar el botón de "Guardar Cliente" se puede comprobar en el listado de clientes que aparece un nuevo registro.



### Caso de uso: Baja cliente

- 1.1. Actor principal: Administrador
- 1.2. Precondición: Estar logado
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Listado cliente, modificar cliente

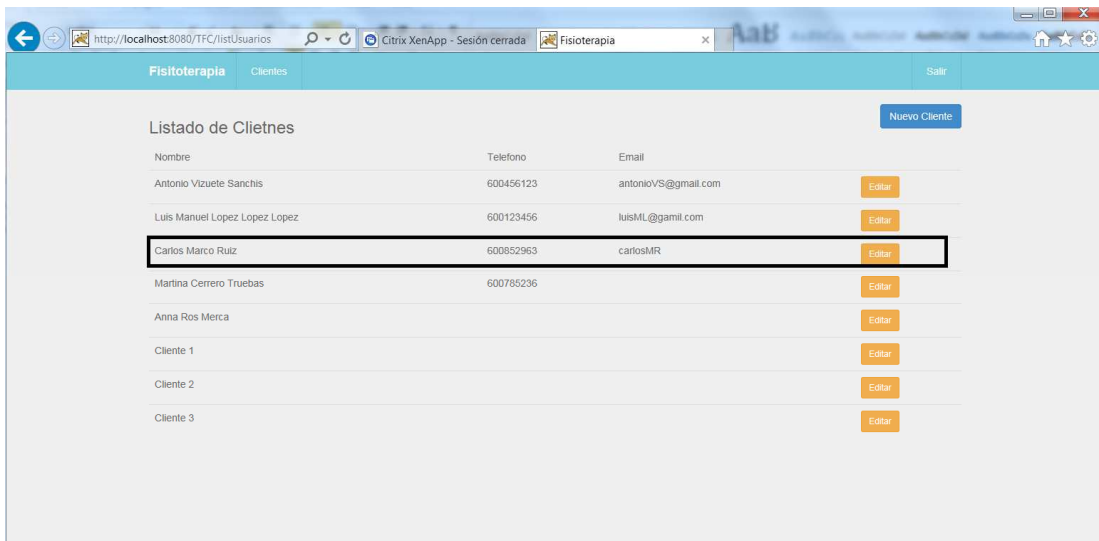
#### 2. Escenario principal:

- 2.1. El usuario se conecta a la web y se loga.
- 2.2. Desde la pantalla inicial el usuario podrá seleccionar editar un usuario
- 2.3. Se muestra el detalle del cliente por pantalla y se muestra un botón de "eliminar"
- 2.4. AL seleccionar el botón de eliminar el sistema solicita confirmación.
- 2.5. Al aceptar el usuario se elimina del listado de cliente

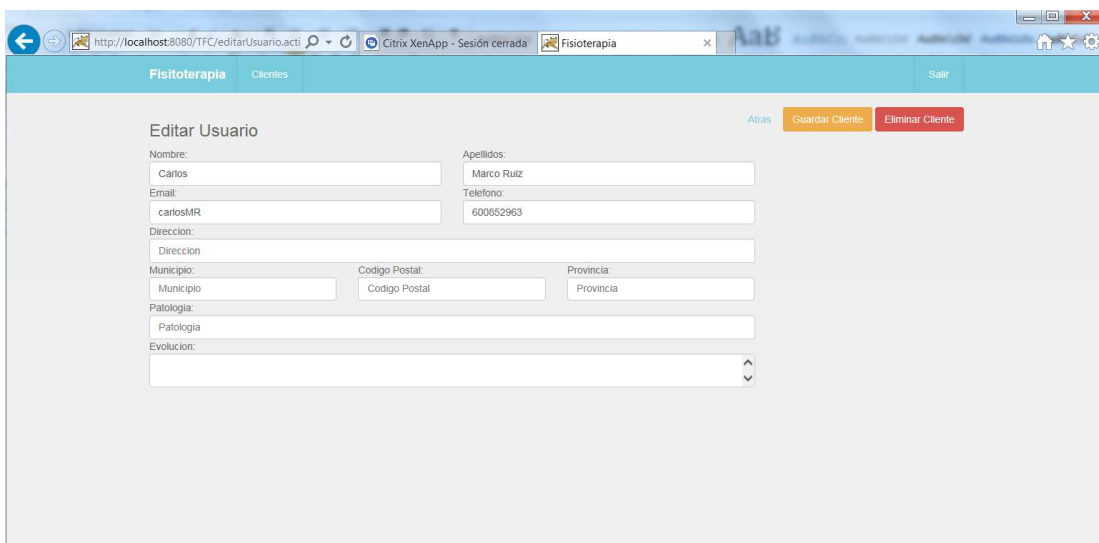
La prueba consiste en eliminar el cliente "Carlos Marzo Ruiz".

Paso 2.2 . Seleccionamos el usuario a través del botón editar

## Memoria.TFC-J2EE.UOC. 014/15 Gestor de clientes y citas (GCC)

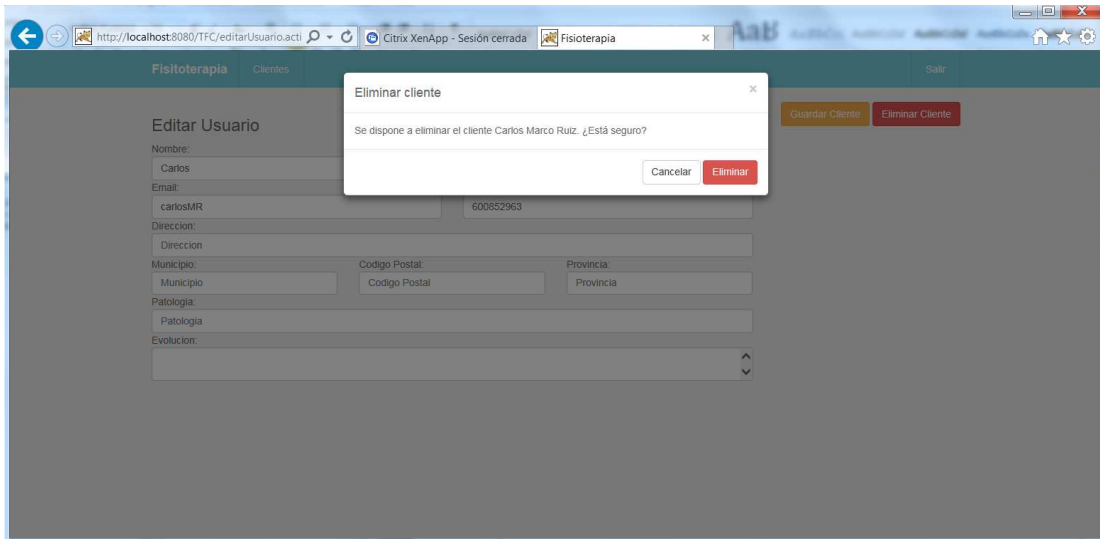


### Paso 2.3 Visualizamos los datos del cliente

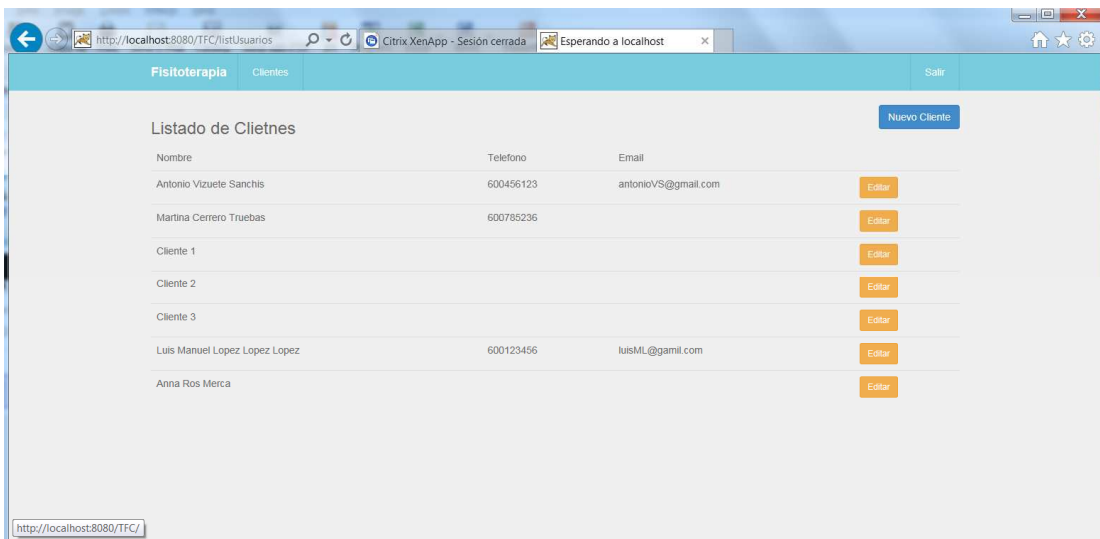


### 2.4 Al pulsar el botón de eliminar el sistema nos pregunta si queremos eliminarlo.

## Memoria.TFC-J2EE.UOC. 014/15 Gestor de clientes y citas (GCC)



2.5 Pulsamos Eliminar y comprobamos que el usuario ya no aparece en el listado.



### Caso de uso: Insertar documentos a un cliente

- 1.1. Actor principal: Administrador
- 1.2. Precondición: Estar logado
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Editar cliente

2. Escenario principal:

- 2.1. El usuario se conecta a la web y se loga.
- 2.2. Desde la pantalla de clientes el usuario podrá solicitar la edición del cliente.
- 2.3 En la pantalla de detalle de usuario se mostrará un botón desde el cuál se podrá insertar documentos

Comentar que este caso de uso no ha se ha implementado por falta de tiempo en el desarrollo.

#### **Caso de uso: Visualizar documentos a un cliente.**

- 1.1. Actor principal: Administrador
- 1.2. Precondición: Estar logado
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Editar cliente

#### 2. Escenario principal:

- 2.1. El usuario se conecta a la web y se loga.
- 2.2. Desde la pantalla de clientes el usuario podrá solicitar la edición del cliente.
- 2.3 En la pantalla de detalle de usuario se mostrará un listado con los documentos insertados para dicho usuario

Comentar que este caso de uso no ha se ha implementado por falta de tiempo en el desarrollo.

#### **Caso de uso: Ver calendario.**

- 1.1. Actor principal: Administrador
- 1.2. Precondición: Estar logado
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Alta cita

#### 2. Escenario principal:

- 2.1. El usuario se conecta a la web y se loga.
- 2.2. Desde la pantalla inicial el usuario podrá ver el calendario de citas que tiene

Fisioterapia    Clientes    Citas    Salir

### Citas

enero 2015

Mes    Semana    Dia    Hoy    <    >

lun.	mar.	mie.	jue.	vie.	sab.	dom.
29	30	31	1	2	3	4
5	6 15:30 Rocio Loza	7	8	9	10	11
12 18:36 Rocio Loza	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

Por defecto el calendario se visualiza el mes actual pero tambien se puede ver por semana y por día



Fisioterapia Clientes Citas Salir

Citas

12 de enero de 2015

Mes Semana **Dia** Hoy < >

	lunes
Todo el día	
14	
15	
16	
17	
18	
19	18:36 - 20:36 Rocio Lozano de Sosa
20	
21	

Fisioterapia Clientes Citas Salir

Citas

12 - 18 de ene. de 2015

Mes **Semana** Dia Hoy < >

	lun. 12/1	mar. 13/1	mie. 14/1	jue. 15/1	vie. 16/1	sab. 17/1	dom. 18/1
Todo el día							
14							
15							
16							
17							
18							
19	18:36 - 20:36 Rocio Lozano de Sosa						
20							
21							

**Caso de uso: Alta cita.**

- 1.1. Actor principal: Administrador
- 1.2. Precondición: Estar logado y que el cliente exista
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Ver calendario

**2. Escenario principal:**

- 2.1. El usuario se conecta a la web y se loga.
- 2.2. Desde la pantalla inicial el usuario podrá ver el calendario de citas que tienen
- 2.3. El usuario podrá dar de alta una cita para una hora, día y cliente determinado.

Las citas se dan de alta desde el detalle de cliente.

**Editar Cliente**

Nombre: Rocio Apellidos: Lozano de Sosa

Email: rlozanos@everis.com Telefono: 115446

Dirección: Calle

Municipio: Andorra Código Postal: 00000 Provincia: Andorra

Patología: Patología

Evolución:

**Citas:**

Nueva Cita

Inicio	Fin	Descripción
12/01/2015 18:36	12/01/2015 20:36	
06/01/2015 15:30	06/01/2015 17:37	

## Memoria.TFC-J2EE.UOC. 014/15 Gestor de clientes y citas (GCC)

The screenshot shows the 'Nueva Cita' (New Appointment) modal form. The form has the following fields:

- Start date and time:
- End date and time:
- Description:

Buttons: Cancelar, Guardar Cita

The background shows the 'Editar Cliente' form for 'Rocio' with fields for Name, Email, Address, Municipality, Postal Code, Province, Pathology, and Evolution. A 'Citas' table is visible at the bottom with columns for Inicio, Fin, and Descripción.

Se informa la fecha y hora de inicio y fecha/hora fin. Si se desea se puede insertar un comentario.

The screenshot shows the 'Nueva Cita' modal form with the following data entered:

- Start date and time:
- End date and time:
- Description:

Buttons: Cancelar, Guardar Cita

The background shows the 'Editar Cliente' form for 'Rocio' and the 'Citas' table with the following data:

Inicio	Fin	Descripción
12/01/2015 18:36	12/01/2015 20:36	
06/01/2015 15:30	06/01/2015 17:37	

Se comprueba que en el listado de citas se visualiza la nueva cita que hemos creado.

Editar Cliente Atras [Guardar Cliente](#) [Eliminar Cliente](#)

Nombre:  Apellidos:

Email:  Telefono:

Direccion:

Municipio: Codigo Postal: Provincia:

Patologia:

Evolucion:

Citas:

[Nueva Cita](#)

Inicio	Fin	Descripción	
12/01/2015 18:36	12/01/2015 20:36		<a href="#">Eliminar Cita</a>
06/01/2015 15:30	06/01/2015 17:37		<a href="#">Eliminar Cita</a>
16/01/2015 12:47	16/01/2015 18:47	Tercera Visita	<a href="#">Eliminar Cita</a>

### Caso de uso: Alta usuario.

- 1.1. Actor: Solo Administrador
- 1.2. Precondición: Estar logado
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Listado de usuarios

#### 2. Escenario principal:

- 2.1. El usuario se conecta a la web y se loga.
- 2.2. Desde la pantalla de administración de usuario los administradores podrán visualizar el listado de usuario y password dados de alta en el sistema, además aparecerá un botón de "Alta usuario"
- 2.3 Se inserta usuario y password y el usuario le da al botón de aceptar
- 2.4 Se visualiza en el listado de usuarios el nuevo usuario dado de alta

Desde el botón "Nuevo Usuario" se acceder a la pantalla de alta usuario.

Nombre	Nombre de usuario	Email	
usuario	usuario	usuario@usuario.com	Editar
Usuario de puebas	User1	user@gmail.com	Editar

Los campos a insertar son Nombre, usuario, mail y password

Nuevo Usuario

Atras Guardar Usuario

Nombre:  Nombre de Usuario:  Email:

Clave:  Repita la clave:

### Caso de uso: Listado usuario.

- 1.1. Actor: Solo Administrador
- 1.2. Precondición: Estar logado
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Alta de usuario, baja de usuarios

#### 2. Escenario principal:

- 2.1. El usuario se conecta a la web y se loga.
- 2.2. Desde la pantalla de administración de usuario los administradores podrán visualizar el listado de usuario y password dados de alta en el sistema

Nombre	Nombre de usuario	Email	
usuario	usuario	usuario@usuario.com	Editar
Usuario de puebas	User1	user@gmail.com	Editar

### Caso de uso: Baja usuario.

- 1.1. Actor: Solo Administrador
- 1.2. Precondición: Estar logado
- 1.3. Post-condición: Ninguna
- 1.4. Casos de uso relacionados: Listado de usuarios


#### 2. Escenario principal:

- 2.1. El usuario se conecta a la web y se loga.
- 2.2. Desde la pantalla de administración de usuario los administradores podrán visualizar el listado de usuario y password dados de alta en el sistema, a
- 2.3. Al acceder al detalle de un usuario deberá mostrarse un botón de Eliminar usuario
- 2.3 Al pulsar "eliminar" el sistema solicita confirmación de la eliminación del usuario.

Desde la pantalla de listado de usuarios se acceder a la pantalla de edición mediante el botón "Editar"

Nombre	Nombre de usuario	Email	
usuario	usuario	usuario@usuario.com	Editar
Usuario de puebas	User1	user@gmail.com	Editar

En la pantalla de edición podemos ver el botón Eliminar Usuario

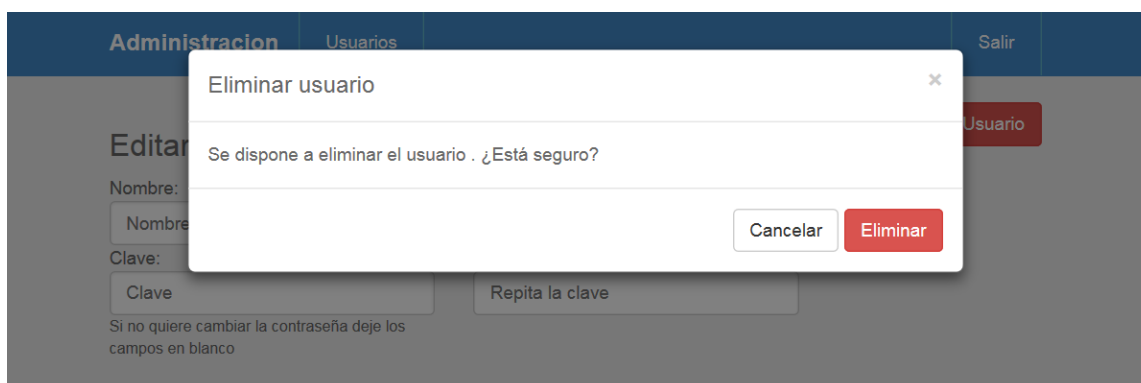


The screenshot shows the 'Editar Usuario' form. At the top, there is a navigation bar with 'Administracion', 'Usuarios', and 'Salir'. Below the navigation bar, there are three buttons: 'Atras' (blue), 'Guardar Usuario' (orange), and 'Eliminar Usuario' (red). The form itself has the following fields and labels:

- Nombre:
- Nombre de Usuario:
- Email:
- Clave:
- Repita la clave:

Below the fields, there is a note: 'Si no quiere cambiar la contraseña deje los campos en blanco'.

Al pulsarlo el sistema nos preguntará si realmente queremos eliminar el usuario



The screenshot shows a dialog box titled 'Eliminar usuario' with a close button (X) in the top right corner. The dialog contains the text: 'Se dispone a eliminar el usuario . ¿Está seguro?'. At the bottom of the dialog, there are two buttons: 'Cancelar' (white) and 'Eliminar' (red).

---

*Funcionalidades futuras*

De cara a un futuro se plantean una serie de mejoras en la aplicación que contribuirán a una mejora significativa en la aplicación proporcionando mayor uso al usuario y una mayor robustez al sistema.

- Añadir la gestión de las citas en el módulo de citas. De esta forma evitaría al usuario tener que entrar en la edición de un cliente para cambiar una cita anularla, etc...
- Poder imprimir el calendario
- Añadir a la página principal una página de welcome donde se explique los tratamientos etc...
- Añadir más módulos enfocados a los clientes:
  - Quienes somos
  - Donde estamos
  - Tratamientos
  - Promociones
  - Contacto
  
- Añadir más datos al paciente.
- Incorporar un sistema de mensajería por mail automático para enviar promociones a los clientes, felicitaciones navideñas
- Avisos del sistema a modo Outlook que ofrecen a los usuarios un sistema para recordar tareas importantes como llamar a clientes, felicitar a un cliente, volver a enviar promociones....etc.
- Parametrizar el sistema para un usuario puede parametrizar la aplicación dependiendo de si es una clínica, una peluquería.... Etc.

## *Conclusiones*

---

En los inicios del proyecto pensé que la carga de trabajo me sería mucho fácil dada mi carrera profesional actual y a la certificación en java 1.5 que poseo. Aun partiendo con una buena base la fase de instalación de software me ha llevado el doble de tiempo y esfuerzo que inicialmente había estimado. Montar la aplicación desde cero, conectar con la base de datos y acceder a la aplicación a través del servidor Tomcat es lo que sin duda más me ha costado.

Las entregas parciales, sin duda son la clave del proyecto, te ayudan a marcarte el ritmo adecuado que garantice el éxito del proyecto, así como tener un feedback con el tutor.

En mi caso ha sido muy útil este soporte ya que he tenido que ampliar las funcionalidades de la aplicación puesto que la propuesta inicial fue demasiado básica.

Al realizar este proyecto he podido aprender numerosas lecciones tanto técnicas como de otros campos, como por ejemplo:

- Montar un servidor Web, crear una base de datos y conectarla con una aplicación.



Hasta ahora solo había visto aplicaciones ya creadas y cuyo servidor se crea por otros equipos de la empresa.

- Ley de protección de datos.  
Las empresas actuales se encuentran con leyes muy estrictas sobre la ley de protección de datos. Esto asegura la confidencialidad de los clientes pero dificulta el trabajo a los pequeños empresarios.
- Valoración de las tareas que no técnicas (documentación, planificación....). Las horas dedicadas a esta parte han superado el tiempo que había estimada.  
Además he aprendido que este tiempo de tareas son tan importantes como que un programa funcione bien.
- Al tener que implementar el calendario, me resultó muy complejo el desarrollo y tuve que investigar sobre framework que ya estuvieran desarrollados e implementarlos en mi aplicación. Finalmente utilicé el framework fullCalendar desarrollado por Adam

## *Glosario*

---

- ECLIPSE:

Plataforma de software de código abierto independiente para desarrollar aplicaciones.

- JAVA:

Es un lenguaje de programación desarrollado por Sun Microsystems. Es independiente de la plataforma en la que se use.

- ITIS:

Ingeniería técnica de Informática de sistemas.

- TFC:

Trabajo final de carrera.

- TOMCAT:

Software libre que sirve de servidor web.

## Bibliografía

---

- LEY DE PROTECCION DATOS: <http://www.ayudaleyprotecciondatos.es/>
- OpenProject: Software para realizar el diagrama de Gant [www.openproj.softonic.com](http://www.openproj.softonic.com)
- Pacesart UML Diagramer: Software utilizado para hacer el diagrama de los casos de usos.
- <http://www.agpd.es> Web sobre documentación de la ley de protección de datos.
  
- <http://getbootstrap.com/>
  
- <http://fullcalendar.io/>
- 

## Anexo 1

---

### Manual de Instalación:

#### 1. Instalación de Java SE 6.0:

- a. Descargamos el paquete JDK del siguiente enlace:  
<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html>
- b. Instalamos el paquete descargado
- c. Damos de alta una nueva variable de sistema
  - i. Entramos a la consola de comandos
  - ii. Configuramos la variable JAVA\_HOME, para ello tecleamos en la consola **set JAVA\_HOME=C:\Ruta\al\jdk** y confirmamos
  - iii. Añadimos esta variable a las rutas del sistema, para ello en la consola de comandos escribimos lo siguiente: **set PATH=%PATH%;%JAVA\_HOME%**

## 2. Instalación de Apache Tomcat:

- d. Descargamos el paquete de Apache Tomcat 7.0 del siguiente enlace:  
<http://apache.rediris.es/tomcat/tomcat-7/v7.0.57/bin/apache-tomcat-7.0.57.exe>
- e. Instalamos el servicio Tomcat
- f. Para comprobar que esté en funcionamiento entramos en <http://localhost:8080/> y debe aparecer el siguiente mensaje "If you're seeing this, you've successfully installed Tomcat. Congratulations!"

The screenshot shows the Apache Tomcat 7.0.57 installation success page. At the top, there is a navigation bar with links: Home, Documentation, Configuration, Examples, Wiki, Mailing Lists, and Find Help. Below the navigation bar, the page title is "Apache Tomcat 7.0.57" and the Apache Software Foundation logo is visible. A green banner displays the message: "If you're seeing this, you've successfully installed Tomcat. Congratulations!". To the left of the banner is the Tomcat logo (a cat). To the right, there are three buttons: "Server Status", "Manager App", and "Host Manager". Below the banner, there is a "Recommended Reading" section with links to "Security Considerations HOW-TO", "Manager Application HOW-TO", and "Clustering/Session Replication HOW-TO". A "Developer Quick Start" section contains links for "Tomcat Setup", "First Web Application", "Realms & AAA", "JDBC DataSources", "Examples", "Servlet Specifications", and "Tomcat Versions". At the bottom, there are three yellow boxes: "Managing Tomcat" (with links for Release Notes, Changelog, Migration Guide, Security Notices), "Documentation" (with links for Tomcat 7.0 Documentation, Tomcat 7.0 Configuration, Tomcat Wiki), and "Getting Help" (with links for FAQ and Mailing Lists, tomcat-announce, tomcat-users, tomcat-dev).

## 3. Instalación de MySQL

- g. Entramos a <http://dev.mysql.com/downloads/mysql/> y descargamos el instalador de MySQL según el sistema operativo que tengamos.
- h. Instalamos el software
- i. Al finalizar la instalación, arrancará el configurador de la instancia del servicio que requerirá contraseña para el usuario root, en esta caso será vertrigo.

## 4. Preparación de la base de datos

- j. Configuramos la variable MYSQL\_DIR, para ello en la consola de comandos escribimos: **set MYSQL\_DIR=C:\Program Files\MySQL\MySQL Server 5.6\bin**  
**set MYSQL\_DIR=C:\Program Files\MySQL\MySQL Server 5.6\bin**  
Es muy importante que la ruta acabe en “\bin”
- k. Configuramos la variable de del sistema PATH, para ello en la consola tecleamos lo siguiente: **set PATH=%PATH%;%JAVA\_HOME%;%MYSQL\_DIR%**
- l. Desde la consola de comando vamos al directorio donde tenemos el script “fisioterapia.sql” y ejecutamos para acceder a la línea de comandos de MySQL:  
Mysql --user=root --password= vertrigo  
Debe aparecer un prompt en pantalla como el siguiente: **mysql>**

m. Ejecutamos

```
CREATE USER 'fisioterapia'@'localhost' IDENTIFIED BY 'fisioterapia';
```

```
GRANT ALL PRIVILEGES ON fisioterapia . * TO 'fisioterapia'@'localhost';
```

```
SET PASSWORD FOR 'fisioterapia'@'localhost' = PASSWORD('fisioterapia');
```

n. Ejecutamos los siguientes comandos (importante respetar los “;”):

**i. CREATE DATABASE fisioterapia;**

**ii. USE fisioterapia**

**iii. source fisioterapia.sql**

o. Para comprobar que se han creado los objetos en la base de datos, ejecutamos: **SHOW TABLES;** Deberíamos obtener el siguiente resultado:

```
+-----+  
| Tables in fisioterapia |  
+-----+  
| fisio_t_admins      |  
| fisio_t_clientes   |  
| fisio_t_ficheros    |  
| fisio_t_patologias |  
| fisio_v_clientes   |  
+-----+  
5 rows in set (0.00 sec)
```

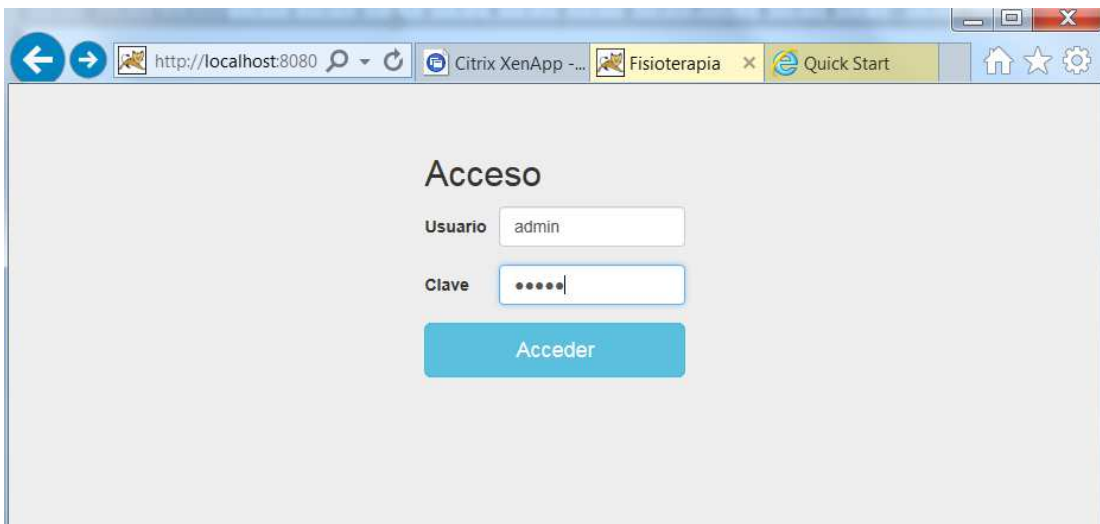
```
Administrador: C:\Windows\system32\cmd.exe - mysql --user=root --password=vertrigo
mysql> SHOW TABLES
-> ;
ERROR 1046 (3D000): No database selected
mysql> ii.      USE fisioterapia
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'ii.
USE fisioterapia' at line 1
mysql> USE fisioterapia
ERROR 1049 (42000): Unknown database 'fisioterapia'
mysql> USE fisioterapia
Database changed
mysql> show tables;
+-----+
| Tables_in_fisioterapia |
+-----+
| fisio_t_admins         |
| fisio_t_clientes      |
| fisio_t_ficheros       |
| fisio_t_patologias     |
| fisio_u_clientes      |
+-----+
5 rows in set (0.03 sec)

mysql> _
```

5. Copiamos el war en la ruta de instalación de Tomcat en el directorio "webapp"

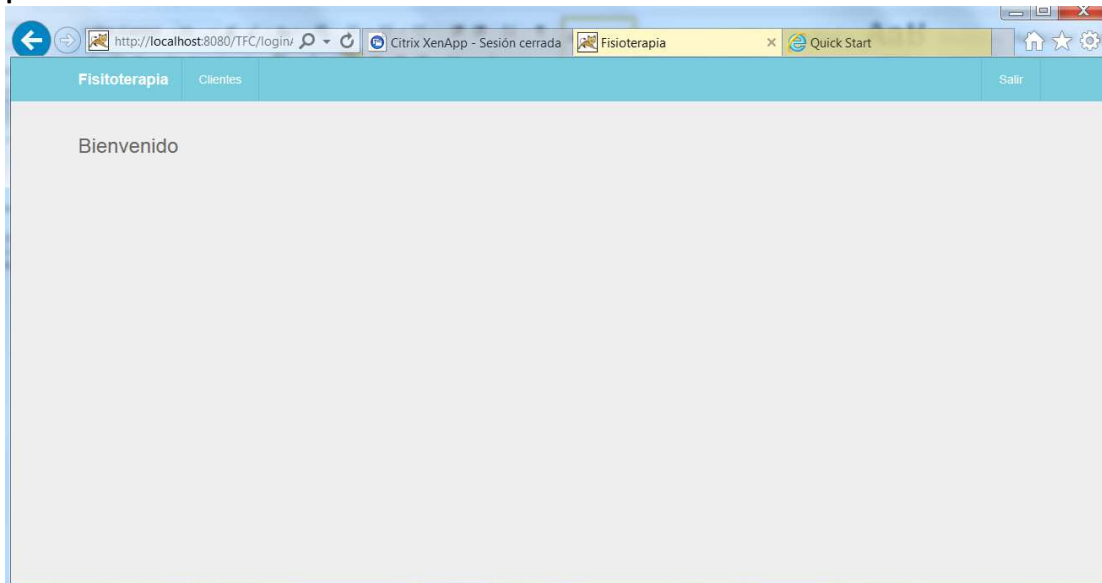
6. Accedemos a <http://localhost:8080/TFC>

Al acceder a la aplicación nos mostrará la pantalla inicial de loggon



Actualmente solo hay dado de alta el usuario administrador:  
**usuario:admin**

password: admin



## Anexo 2: Sentencias DDL

```
SET FOREIGN_KEY_CHECKS=0;  
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";  
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;  
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;  
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;  
/*!40101 SET NAMES utf8 */;
```

```
--  
-- Base de datos: `fisioterapia`  
--
```

```
--  
-- Estructura de tabla para la tabla `fisio_t_admins`  
--
```

```
DROP TABLE IF EXISTS `fisio_t_admins`;  
CREATE TABLE IF NOT EXISTS `fisio_t_admins` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(45) DEFAULT NULL,  
  `password` varchar(45) DEFAULT NULL,  
  `nombre` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

```
--  
-- Volcado de datos para la tabla `fisio_t_admins`  
--
```

```
INSERT INTO `fisio_t_admins` (`id`, `username`, `password`, `nombre`)
VALUES
(1, 'admin', 'admin', 'Admin');

-- -----
--
-- Estructura de tabla para la tabla `fisio_t_citas`
--

DROP TABLE IF EXISTS `fisio_t_citas`;
CREATE TABLE IF NOT EXISTS `fisio_t_citas` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `fechaIni` datetime NOT NULL,
  `fechaFin` datetime NOT NULL,
  `id_cliente` int(11) NOT NULL,
  `descripcion` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`, `fechaIni`, `id_cliente`),
  KEY `id_cliente` (`id_cliente`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
--
-- Estructura de tabla para la tabla `fisio_t_clientes`
--

DROP TABLE IF EXISTS `fisio_t_clientes`;
CREATE TABLE IF NOT EXISTS `fisio_t_clientes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_usuario` int(11) NOT NULL,
  `nombre` varchar(100) NOT NULL,
  `apellidos` varchar(200) NOT NULL,
  `email` varchar(100) NOT NULL,
  `telefono` varchar(20) NOT NULL,
  `direccion` varchar(100) NOT NULL,
  `codpostal` varchar(5) NOT NULL,
  `municipio` varchar(100) NOT NULL,
  `provincia` varchar(100) NOT NULL,
  PRIMARY KEY (`id`, `id_usuario`),
  KEY `id_usuario` (`id_usuario`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

--
-- volcado de datos para la tabla `fisio_t_clientes`
--

INSERT INTO `fisio_t_clientes` (`id`, `id_usuario`, `nombre`,
`apellidos`, `email`, `telefono`, `direccion`, `codpostal`, `municipio`,
`provincia`) VALUES
(1, 1, 'Antonio', 'Vizquete Sanchis', 'antoniovizquete@gmail.com',
'600661298', 'C/ Labradores, 23', '03690', 'San Vicent del Raspeig',
'Alicante'),
(2, 1, 'Rocio', 'Lozano de Sosa', 'rlozanos@everis.com', '115446',
'Calle', '00000', 'Andorra', 'Andorra');

-- -----
--
-- Estructura de tabla para la tabla `fisio_t_ficheros`
--

DROP TABLE IF EXISTS `fisio_t_ficheros`;
```

```
CREATE TABLE IF NOT EXISTS `fisio_t_ficheros` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `id_cliente` int(11) NOT NULL,  
  `fichero` mediumblob NOT NULL,  
  PRIMARY KEY (`id`,`id_cliente`),  
  KEY `id_cliente` (`id_cliente`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;  
  
-----  
  
--  
-- Estructura de tabla para la tabla `fisio_t_patologias`  
--  
  
DROP TABLE IF EXISTS `fisio_t_patologias`;  
CREATE TABLE IF NOT EXISTS `fisio_t_patologias` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `id_cliente` int(11) NOT NULL,  
  `patologia` varchar(255) NOT NULL,  
  `evolucion` text NOT NULL,  
  PRIMARY KEY (`id`,`id_cliente`),  
  KEY `id_cliente` (`id_cliente`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;  
  
--  
-- Volcado de datos para la tabla `fisio_t_patologias`  
--  
  
INSERT INTO `fisio_t_patologias` (`id`, `id_cliente`, `patologia`,  
  `evolucion`) VALUES  
(1, 1, 'Esguince', '  
          10/11/2014: Empezo\r\n10/12/2014: acabo\r\n\r\n          ');  
  
-----  
  
--  
-- Estructura de tabla para la tabla `fisio_t_usuarios`  
--  
  
DROP TABLE IF EXISTS `fisio_t_usuarios`;  
CREATE TABLE IF NOT EXISTS `fisio_t_usuarios` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(45) DEFAULT NULL,  
  `password` varchar(45) DEFAULT NULL,  
  `nombre` varchar(45) DEFAULT NULL,  
  `email` varchar(255) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;  
  
--  
-- Volcado de datos para la tabla `fisio_t_usuarios`  
--  
  
INSERT INTO `fisio_t_usuarios` (`id`, `username`, `password`, `nombre`,  
  `email`) VALUES  
(1, 'usuario', 'usuario', 'usuario', 'usuario@usuario.com');  
  
-----  
  
--  
-- Estructura stand-in para la vista `fisio_v_clientes`  
--  
DROP VIEW IF EXISTS `fisio_v_clientes`;
```



```
CREATE TABLE IF NOT EXISTS `fisio_v_clientes` (  
  `id` int(11)  
  , `id_usuario` int(11)  
  , `nombre` varchar(100)  
  , `apellidos` varchar(200)  
  , `email` varchar(100)  
  , `telefono` varchar(20)  
  , `direccion` varchar(100)  
  , `codpostal` varchar(5)  
  , `municipio` varchar(100)  
  , `provincia` varchar(100)  
  , `patologia` varchar(255)  
  , `evolucion` text  
  , `fichero` mediumblob  
);  
-----  
  
--  
-- Estructura Stand-in para la vista `fisio_v_lista_citas`  
--  
DROP VIEW IF EXISTS `fisio_v_lista_citas`;  
CREATE TABLE IF NOT EXISTS `fisio_v_lista_citas` (  
  `id` int(11)  
  , `fechaIni` datetime  
  , `fechaFin` datetime  
  , `id_cliente` int(11)  
  , `descripcion` varchar(255)  
  , `nombre` varchar(301)  
);  
-----  
  
--  
-- Estructura para la vista `fisio_v_clientes`  
--  
DROP TABLE IF EXISTS `fisio_v_clientes`;  
  
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY  
DEFINER VIEW `fisio_v_clientes` AS select `c`.`id` AS  
`id`, `c`.`id_usuario` AS `id_usuario`, `c`.`nombre` AS  
`nombre`, `c`.`apellidos` AS `apellidos`, `c`.`email` AS  
`email`, `c`.`telefono` AS `telefono`, `c`.`direccion` AS  
`direccion`, `c`.`codpostal` AS `codpostal`, `c`.`municipio` AS  
`municipio`, `c`.`provincia` AS `provincia`, `p`.`patologia` AS  
`patologia`, `p`.`evolucion` AS `evolucion`, `f`.`fichero` AS `fichero`  
from (( `fisio_t_clientes` `c` left join `fisio_t_patologias` `p`  
on((`c`.`id` = `p`.`id_cliente`))) left join `fisio_t_ficheros` `f`  
on((`c`.`id` = `f`.`id_cliente`)));  
-----  
  
--  
-- Estructura para la vista `fisio_v_lista_citas`  
--  
DROP TABLE IF EXISTS `fisio_v_lista_citas`;  
  
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY  
DEFINER VIEW `fisio_v_lista_citas` AS select `ci`.`id` AS  
`id`, `ci`.`fechaIni` AS `fechaIni`, `ci`.`fechaFin` AS  
`fechaFin`, `ci`.`id_cliente` AS `id_cliente`, `ci`.`descripcion` AS  
`descripcion`, concat(`c`.`nombre`, ' ', `c`.`apellidos`) AS `nombre` from  
( `fisio_t_citas` `ci` join `fisio_t_clientes` `c` on((`ci`.`id_cliente` =  
`c`.`id`))) where 1;  
-----  
  
--
```

```
-- Restricciones para tablas volcadas
--
--
-- Filtros para la tabla `fisio_t_citas`
--
ALTER TABLE `fisio_t_citas`
  ADD CONSTRAINT `fisio_t_citas_ibfk_1` FOREIGN KEY (`id_cliente`)
REFERENCES `fisio_t_clientes` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
--
-- Filtros para la tabla `fisio_t_clientes`
--
ALTER TABLE `fisio_t_clientes`
  ADD CONSTRAINT `fisio_t_clientes_ibfk_1` FOREIGN KEY (`id_usuario`)
REFERENCES `fisio_t_usuarios` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
--
-- Filtros para la tabla `fisio_t_ficheros`
--
ALTER TABLE `fisio_t_ficheros`
  ADD CONSTRAINT `fisio_t_ficheros_ibfk_1` FOREIGN KEY (`id_cliente`)
REFERENCES `fisio_t_clientes` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
--
-- Filtros para la tabla `fisio_t_patologias`
--
ALTER TABLE `fisio_t_patologias`
  ADD CONSTRAINT `fisio_t_patologias_ibfk_1` FOREIGN KEY (`id_cliente`)
REFERENCES `fisio_t_clientes` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
SET FOREIGN_KEY_CHECKS=1;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```