



TÉCNICO EN DESARROLLO DE APLICACIONES WEB  
Departamento de Informática

## PROYECTO

MediControl  
**Manual Técnico**

Autor: Esther Herculano Aparicio  
Curso Académico: 2020/21

## CONTENIDO

1	Introducción .....	2
2	Objetivos.....	3
3	Tecnologías involucradas .....	4
3.1	Análisis de requisitos.....	4
3.2	Diseño.....	6
3.2.1	Base de datos.....	6
3.2.2	Estructura del sistema .....	11
3.2.3	Interfaz.....	13
3.3	Tecnologías relacionadas.....	18
3.4	Estructura del proyecto .....	19
4	Proceso de desarrollo .....	21
4.1	Ciclo de vida utilizado.....	21
4.2	Orden de desarrollo de las funcionalidades del proyecto .....	21
5	Proceso de despliegue .....	24
5.1	Hosting web (000webhost) .....	24
5.1.1	Registro y creación del sitio .....	24
5.1.2	Acceso y creación del sitio.....	26
5.1.3	Creación de la base de datos .....	27
5.1.4	Subida de ficheros .....	29
5.1.5	Acceso a la aplicación .....	31
5.2	Servidor local (XAMPP).....	32
5.2.1	Creación de la base de datos .....	32
5.2.2	Acceso a la aplicación .....	34
5.3	Apache + MySQL + FTP (Linux).....	35
5.3.1	Apache .....	35
5.3.2	Instalación de MySQL.....	38
5.3.3	Instalación de FTP.....	40
6	Propuesta de mejora o trabajos futuros .....	41
7	Bibliografía.....	42

## 1 INTRODUCCIÓN

---

Este proyecto consiste en el desarrollo de una aplicación web destinada a proporcionar una forma de gestión para centros médicos o dedicados al ámbito de la salud.

La aplicación consta de varias interfaces cuyo aspecto y funcionalidad dependerán del rol del usuario registrado, el cual podrá tratarse de un cliente (paciente) o un empleado del centro en cuestión.

Por otra parte, al tratarse de una aplicación web, los usuarios podrán acceder a ella desde cualquier dispositivo a través de un navegador web y una conexión a internet.

## 2 OBJETIVOS

---

El objetivo de esta aplicación es proporcionar un área para el usuario de manera que éste pueda acceder a la información relevante a la que pueda tener acceso, ya que los sitios web de estos centros médicos no proporcionan ningún medio mediante el cual gestionar con totalidad las citas del usuario o acceder al historial de consultas realizadas entre otras cosas.

De esta manera se pretende mejorar la experiencia del usuario cliente a través de:

- La posibilidad de gestionar citas online, las cuales hasta el momento y por lo general se han debido realizar o modificar a través de una llamada telefónica o presencialmente en el centro correspondiente.
- La posibilidad de acceder a información relevante, de forma que el cliente podrá conocer los tratamientos que ha llevado a cabo en el centro o cuándo fue su última consulta, entre otras cosas.

Además, se ofrece la posibilidad de gestionar a los clientes, sus citas y consultas, a través de sus empleados y los administradores, a través de esta aplicación, de manera que se pueda proporcionar a los clientes la información mencionada anteriormente.

Nos referimos a permitir la gestión del negocio a través de:

- La automatización de citas, donde la cita que el cliente haya concertado de forma online aparecerá automáticamente en la agenda de los empleados.
- La conservación de información relativa a consultas y tratamientos llevados a cabo, precios de los mismos (cobros), clientes tratados y empleados que han realizado la consulta o tratamiento, empleados registrados...

## 3 TECNOLOGÍAS INVOLUCRADAS

---

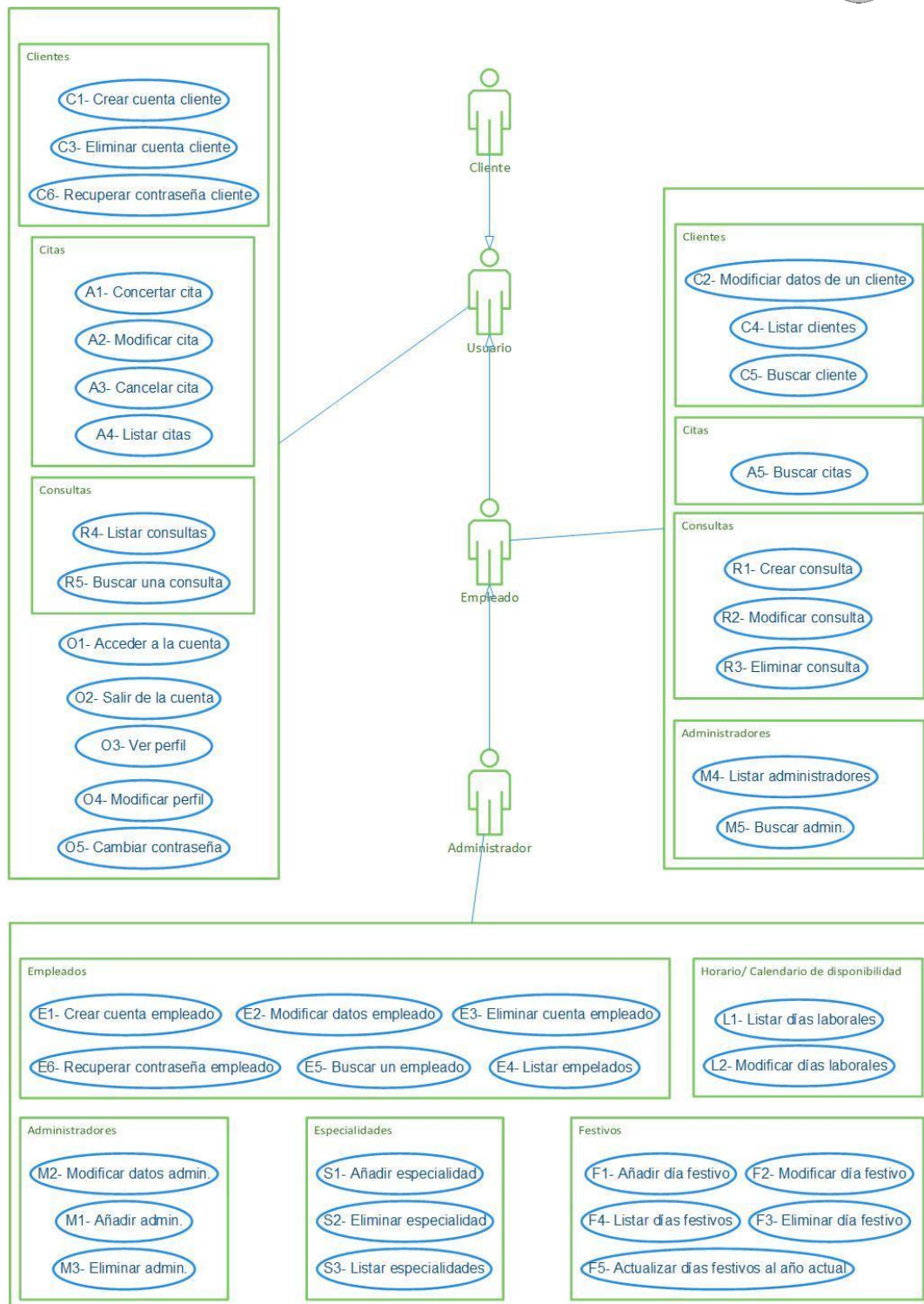
En este apartado veremos las siguientes fases del diseño de la aplicación:

- Análisis de requisitos, donde presentaremos el diagrama de casos de uso.
- Diseño de la base de datos (con un diagrama E/R y el modelo relacional del mismo), de la estructura de la aplicación (mediante un diagrama de clases) y de la interfaz (a través de la guía de estilos).
- Desarrollo, donde indicaremos las tecnologías escogidas para el desarrollo de la aplicación además de la estructuración del proyecto.

### 3.1 ANÁLISIS DE REQUISITOS

A través del diagrama de casos de usos que presentamos a continuación, veremos de forma gráfica y esquemática las funciones que proporciona la aplicación para cada rol de usuario posible.

Cada requisito funcional consta de un prefijo que utilizaremos más adelante para referirnos a ellos y para una mayor organización, se han agrupado las diferentes funciones según el elemento al que afectan, siendo los precedidos por la letra “O” relativos a la cuenta y el perfil del usuario.



Como vemos, hay que permitir en todo momento la gestión de citas, consultas y usuarios por parte de todos los usuarios registrados (clientes, empleados y administradores), para lo cual son necesarios el almacenaje y el acceso a diversa información sobre todos estos elementos.

Por ello, el sistema deberá garantizar los siguientes aspectos, que serán los requisitos no funcionales de la aplicación:

- Disponibilidad 24/7, de manera que todos los usuarios puedan acceder a la aplicación en cualquier momento.
- Confidencialidad de los datos, siendo los datos personales de los clientes únicamente accesibles por los empleados y los datos de éstos, por el administrador.
- Control de accesos.
- Control de errores.

## 3.2 DISEÑO

### 3.2.1 Base de datos

Toda la información manejada por los usuarios para la gestión de cualquier elemento, almacenará en una base de datos.

Para la elaboración del diagrama E/R de esta base de datos, partimos del siguiente resumen de la aplicación como base:

- La aplicación será manejada por usuarios que podrán tener uno de los siguientes roles:
  - o Cliente.
  - o Empleado.
  - o Administrador.
- Cada usuario tendrá una cuenta con la cual acceder a todas las funcionalidades de la aplicación disponibles para él.
- Cuando registramos un nuevo usuario, nos interesa conocer ciertos datos de los mismos:
  - o Datos en común → DNI, nombre, apellidos, dirección de correo electrónico, teléfono de contacto, género y fecha de nacimiento.
  - o Clientes → Compañía aseguradora.
  - o Empleados → Fecha de alta y baja, especialidad y extensión.
- En el caso de los empleados hay que tener en cuenta que podrán ser gestionados por uno o varios administradores.
- Por otra parte, los usuarios podrán gestionar citas a partir de las cuales un empleado generará una consulta con la información sobre pruebas realizadas, tratamientos necesarios, importe de la consulta y situación de pago.
- De las citas tendremos que saber su fecha, hora y asunto.
- Esta fecha podrá ser un día laboral o un día festivo con disponibilidad para realizar citas.
- Por último, debemos almacenar información relativa al calendario de disponibilidad del centro para poder concertar citas y comprobar la disponibilidad de la fecha elegida de forma automática. Esta información será relativa a:
  - o Días laborales normales.
  - o Días festivos.



## **3.2.1.1 Modelo E/R**

Comenzamos modelando la base de datos que precisaremos utilizar para el correcto funcionamiento de la aplicación.

Dado el supuesto anterior, extraeremos las entidades participantes en el mismo junto con los atributos que tendrán y que se corresponderán con la información que necesitamos almacenar.



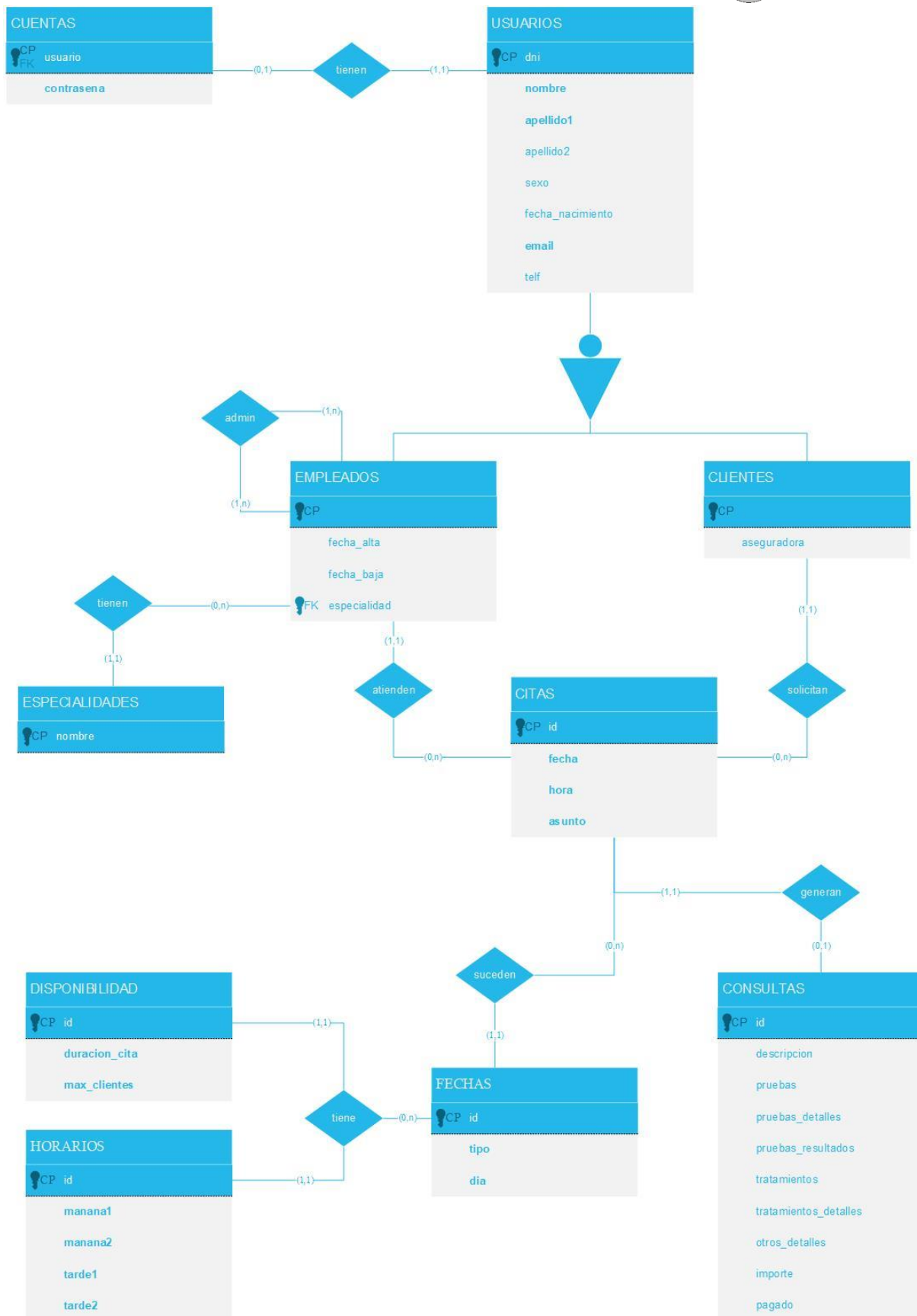
# Proyecto “Desarrollo de Aplicaciones Web”

Título del proyecto: MediControl



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



## 3.2.1.2 Modelo relacional

A partir del modelo E/R presentado en el subapartado anterior, obtendremos las tablas que compondrán nuestra base de datos. Junto con estas tablas se harán las indicaciones oportunas para una mayor comprensión sobre el objetivo de su existencia.

FECHAS				
Nombre	PK	FK	NULL	Tipo
id	Sí	-	No	number
id_disponibilidad	-	DISPONIBILIDAD	No	number
id_horarios	-	FESTIVOS	No	number
tipo	-	-	No	char
dia	-	-	No	varchar

DISPONIBILIDAD				
Nombre	PK	FK	NULL	Tipo
id	Sí	-	No	number
duracion_cita	-	-	No	number
max_clientes	-	-	No	number

HORARIOS				
Nombre	PK	FK	NULL	Tipo
id	Sí	-	No	number
manana1	-	-	No	time
manana2	-	-	No	time
tarde1	-	-	No	time
tarde2	-	-	No	time

En la tabla “FECHAS” almacenamos los días con indicación de su tipo (laboral o festivo). En el caso de días laborales almacenaremos el día de la semana, en caso de ser festivo, la fecha. Accederemos a la disponibilidad y horarios de estos días a través de los ids correspondientes.

Con estas tablas configuramos el calendario y horarios laborales y festivos para posibilitar una gestión de citas automatizada.

CUENTAS				
Nombre	PK	FK	NULL	Tipo
usuario	Sí	USUARIOS	No	varchar
contrasena	-	-	No	varchar

USUARIOS				
Nombre	PK	FK	NULL	Tipo
dni	Sí	-	No	varchar

nombre	-	-	No	varchar
apellido1	-	-	No	varchar
apellido2	-	-	Sí	varchar
sexo	-	-	Sí	char (F/M)
fecha_nacimiento	-	-	Sí	date
email	-	-	No	varchar
telf	-	-	Sí	number

CLIENTES				
Nombre	PK	FK	NULL	Tipo
dni	Sí	USUARIOS	No	varchar
aseguradora	-	-	Sí	varchar

EMPLEADOS				
Nombre	PK	FK	NULL	Tipo
dni	Sí	USUARIOS	No	varchar
fecha_alta	-	-	Sí	date
fecha_baja	-	-	Sí	date
especialidad	-	ESPECIALIDADES	No	varchar
extension	-	-	Sí	varchar

ESPECIALIDADES				
Nombre	PK	FK	NULL	Tipo
nombre	Sí	-	No	varchar

ADMINISTRADORES				
Nombre	PK	FK	NULL	Tipo
dni_admin	Sí	EMPLEADOS	No	varchar
dni_empleado	Sí	EMPLEADOS	No	varchar

CITAS				
Nombre	PK	FK	NULL	Tipo
id	Sí	-	No	number
dni_cliente	-	CLIENTES	No	varchar
dni_empleado	-	EMPLEADOS	Sí	varchar
id_fecha	-	FECHAS	No	number
fecha	-	-	No	date
hora	-	-	No	time
asunto	-	-	No	varchar

Las citas se relacionan con las subclases de la entidad “USUARIOS” y no con éste directamente, de manera que contamos con una referencia a ambos intervinientes en dicha cita: el encargado (empleado) y el cliente (paciente).

Por otra parte, se indicará el id de la fecha de la cita de forma que ambas tablas se encuentren relacionadas. En caso de tratarse de un día laboral se relacionará con el día de la semana (del 1 al 7) que corresponda la fecha indicada; en caso de ser un día festivo se corresponderá con el mismo día en la tabla “fechas”.

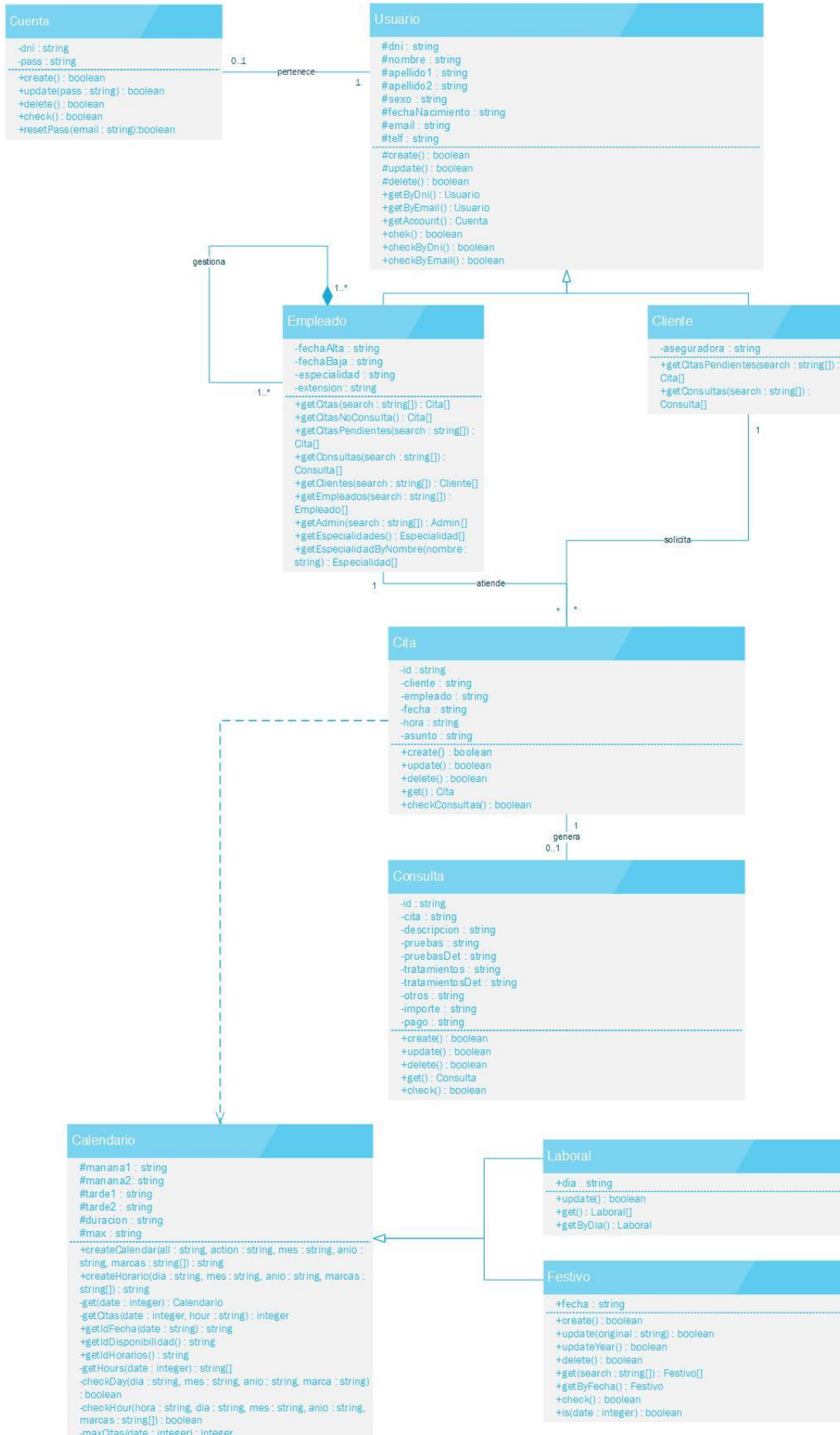
CONSULTAS				
Nombre	PK	FK	NULL	Tipo
id	Sí	-	No	number
id_cita	-	CITAS	No	number
descripción	-	-	Sí	varchar
pruebas	-	-	Sí	boolean
pruebas_detalle	-	-	Sí	varchar
tratamientos	-	-	Sí	boolean
tratamientos_detalle	-	-	Sí	varchar
otros_detalle	-	-	Sí	varchar
importe	-	-	Sí	number
pago	-	-	Sí	boolean

## 3.2.2 Estructura del sistema

De forma similar al apartado anterior, elaboraremos el diagrama de clases, donde, para una mayor claridad, se han omitido los siguientes métodos, comunes a todos los elementos:

- Métodos get (obtención del valor de un atributo concreto).
- Métodos set (asignación del valor de un atributo concreto).
- Métodos set con todos los atributos como argumentos (actúa como constructor de la clase).

Título del proyecto: MediControl



## 3.2.3 Interfaz

En este apartado desarrollaremos la guía de estilos utilizada.

### 3.2.3.1 Estructura

#### 3.2.3.1.1 Sitio web



Como vemos, la aplicación consta de 11 páginas diferentes, donde la estructura a utilizar será la reticular, de manera que permitimos al usuario navegar de una página a otra sin orden aparente y la inter-relación de publicaciones.

#### 3.2.3.1.2 Páginas web

Cada página que compone la aplicación tiene una estructura determinada siendo muy similar en todas ellas.

En todos los casos encontramos:

- Cabecera con el logotipo de la aplicación.
- Contenido de la página.
- Pie de página con un mensaje de copyright.

En la página principal encontraremos además un menú de navegación para acceder al perfil del usuario y cerrar la sesión del mismo.

Con excepción de la página de acceso y la página principal, el resto de páginas contarán con el menú de navegación para el acceso al perfil del usuario y el cierre de sesión, además de un segundo menú de navegación para permitir la navegación entre páginas.

De manera adicional se añade un cuadro de mensajes que aparecerá solo cuando haya mensajes que mostrar y que se ubicará justo antes del contenido de la página.

### 3.2.3.1.3 Contenido de las páginas

Elemento	Contenido
Cabecera	Imagen del logotipo
Menú de navegación del perfil	Enlaces para acceder al perfil del usuario y cerrar sesión
Menú de navegación de páginas	Enlaces a otras páginas
Cuadro de mensajes	Mensajes a mostrar al usuario (campos a rellenar, procesos llevados a cabo con éxito o errores)
Contenido de la página	Formularios, listados y mensajes oportunos
Pie de página	Mensaje de copyright

### 3.2.3.1.4 Estilos para las páginas

Los estilos utilizados se aplican a través de Bootstrap, personalizando solo algunos elementos en concreto. Por tanto, para los menús de navegación utilizaremos la clase navbar y relacionados, para los formularios la clase form-row y relacionados, para las tablas la clase table y relacionados y para los botones la clase btn.

#### 3.2.3.1.4.1 Alineación

El texto se alineará a la izquierda excepto en los siguientes casos, que tendrán una alineación centrada:

- Imagen del logotipo.
- Contenido de tablas.
- Mensaje de bienvenida de la página principal.

#### 3.2.3.1.4.2 Medidas (anchos y altos)

Elemento	Medidas
Cabecera, menús de navegación, cuadro de mensajes, contenido de la página y pie de página	Ancho – 100%
Tablas	Ancho – 100%
Formularios	Ancho – 100% Ancho – 50% a partir de 992px para formulario de acceso y recuperación de contraseña de la página de login
Imagen de logotipo	Ancho – 50% con un mín. de 150px y un máx. de 300px
Calendario de citas	Ancho – 100% hasta 350px



## 3.2.3.1.4.3 Márgenes

Elemento	Medidas de margen
Contenido de la página	Vertical – 50px Horizontal – 5%
Cuadro de mensajes	Horizontal – 5% Los mensajes de información para la cumplimentación de los formularios anularán el margen superior de 3rem de estos para un menor distanciamiento, además de contar con un margen superior de 25px para añadir una separación con los posibles mensajes de confirmación o error
Tablas	Superior – 3rem
Calendario de citas	Inferior – 1rem
Formularios	Superior – 3rem (el primer formulario no debe ampliar espacio con respecto al mensaje que le precede –si existe)
Botones de formularios	Superior – 1.5rem (el conjunto de botones) Inferior – 0.25rem (cada botón individualmente) En el caso de los botones incluidos en las tablas (listados) el margen será en todos los casos de 0.25rem En el caso de grupos de formularios, hay que igualar el margen inferior a 1rem para estar a la misma altura que estos (bootstrap les aplica un margen inferior de 1rem a los grupos de formularios)
Mensaje de bienvenida de la página principal	Vertical – 3rem
Enlaces de la página principal	Vertical – 1rem
Enlaces de los menús de navegación	Vertical y horizontal – 0.5rem

## 3.2.3.1.4.4 Columnas hasta 576px

En todos los casos todos los elementos se situarán en una sola columna al 100%.

El segundo menú de navegación se contraerá en acordeón hasta una resolución 768px.

## 3.2.3.1.4.5 Columnas hasta 768px

La estructura general de una página con una resolución de 576px inclusive hasta 768px es:

- Formularios – 1 columna (100%).
- Campos de los formularios – 2 columnas (50%) salvo destacados (en formularios con un número impar de campos siempre habrá un campo destacado).



- Botones en fila que forman 1 columna salvo en formularios de búsqueda, que se alinearán con el resto de campos en la segunda columna en este caso.
- Tablas – 1 columna (100%).

## 3.2.3.1.4.6 Columnas hasta 992px

La estructura general de una página con una resolución de 768px inclusive hasta 992px es:

- Formularios – 1 columna (100%).
- Campos de los formularios – 3 columnas (33,33%) salvo destacados (en formularios con un número impar de campos siempre habrá un campo destacado).
- En los formularios con un desplegable como único campo, éste y los botones se organizarán al 50%
- Botones en fila que forman 1 columna salvo en formularios de búsqueda, que se alinearán con el resto de campos en la tercera columna en este caso (si solo hay un campo, éste y los botones se repartirán al 50%).
- Tablas – 1 columna (100%).

## 3.2.3.1.4.7 Columnas a partir de 992px

La estructura general de una página con una resolución de 768px inclusive hasta 992px es:

- Formularios – 1 columna (100%).
- Campos de los formularios – 3 columnas (33,33%) salvo destacados (en formularios con un número impar de campos siempre habrá un campo destacado).
- En los formularios con un desplegable como único campo, éste y los botones se organizarán al 50%
- Botones en fila que forman 1 columna salvo en formularios de búsqueda, que se alinearán con el resto de campos en la cuarta columna en este caso (si solo hay un campo este y los botones se repartirán al 50%, si son dos campos será al 33,33%).
- Tablas – 1 columna (100%).

## 3.2.3.2 Colores

Color	Uso
#FFF	<ul style="list-style-type: none"><li>• Fondo del header</li><li>• Texto del menú de navegación del perfil</li><li>• Fondo del menú de navegación de páginas</li></ul>
#000	<ul style="list-style-type: none"><li>• Texto normal para el contenido</li></ul>
#CCC	<ul style="list-style-type: none"><li>• Fondo para los días deshabilitados en el calendario de citas</li></ul>
#666	<ul style="list-style-type: none"><li>• Texto de los mensajes informativos</li><li>• Texto para los días deshabilitados en el calendario de citas</li></ul>
#0055FF	<ul style="list-style-type: none"><li>• Logotipo</li><li>• Enlace de la página de error</li></ul>

	<ul style="list-style-type: none"> <li>Fondo del menú de navegación del perfil</li> <li>Fondo de los botones para crear o actualizar datos</li> <li>Texto del menú de navegación de páginas</li> </ul>
#FFCC99	<ul style="list-style-type: none"> <li>Fondo del pie de página</li> </ul>
#FF9900	<ul style="list-style-type: none"> <li>Enlaces cuando el ratón está encima</li> <li>Días habilitados del calendario de citas</li> <li>Opciones de los menús de navegación</li> <li>Fondo de los botones para recuperar contraseñas</li> </ul>
#993300	<ul style="list-style-type: none"> <li>Borde superior del pie de página</li> <li>Texto del pie de página</li> <li>Enlace visitado de la página de error</li> <li>Fondo para los botones de reseteo de formulario</li> </ul>
#009900	<ul style="list-style-type: none"> <li>Texto de los mensajes de confirmación</li> </ul>
#EE0000	<ul style="list-style-type: none"> <li>Texto de los mensajes de error</li> <li>Fondo de los botones de eliminación</li> </ul>


### 3.2.3.3 Tipografía

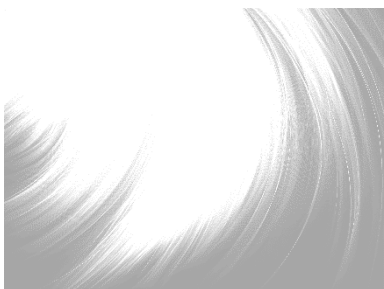
La fuente por defecto será Cambria, indicando como fuentes secundarias Calibri y Verdana (en este orden).

Los tamaños a utilizar serán los siguientes:

Elemento	Medida (px)	Usos
p	16	Texto normal en todos los casos
small	14	Pie de página
big	20	
h1	32	
h2	30	
h3	28	Mensaje de bienvenida en la página principal
h4	26	
h5	24	
h6	22	
legend	22	Títulos de formularios



### 3.2.3.4 Imágenes

Imagen	Medidas	Uso
	<p>Ancho del 50% del ancho de la página con un mín. de 150px y un máx. de 300px.</p> <p>La altura será proporcional, con un mín.</p>	<p>Logotipo de la aplicación siempre visible en el header</p>

	de 65.37px y un máx. de 130.75px.	
	100% del contenido de la página	Fondo del contenido en todas las páginas

### 3.2.3.4.1 Iconos

Los iconos utilizados han sido obtenidos de “Font Awesome”, siendo los siguientes:

Icono	Uso
	Indicación de la página de principal
	Indicación del cierre de sesión del usuario

También utilizaremos la siguiente imagen como icono de la aplicación visible en la pestaña del navegador:



## 3.3 TECNOLOGÍAS RELACIONADAS

Al tratarse de una aplicación web, se tratará de una aplicación multiplataforma que únicamente precisará para su uso de acceso a Internet y un navegador web, además de un servidor web donde alojar el sitio, es decir, la aplicación, y ponerlo a disposición de los usuarios.

Los lenguajes a utilizar para el desarrollo de MediControl serán:

- Client-side: HTML5, CSS3, JS
- Server-side: PHP

No se hará uso de ningún framework, pero sí utilizaremos las siguientes librerías: bootstrap y JQuery.

Por otro lado, el motor de base de datos utilizado será MySQL, ya que se encuentra ampliamente extendido además de ser gratuito y multiplataforma.

De forma resumida, los medios y tecnologías utilizados serán las siguientes:

Sistema operativo	Multiplataforma
Requisitos software	Conexión a Internet Navegador web Servidor web + MySQL + FTP
Lenguaje cliente	HTML5, CSS3, JS
Lenguaje servidor	PHP
Librerías	Bootstrap, JQuery
Base de datos	MySQL + SQL

## 3.4 ESTRUCTURA DEL PROYECTO

Siguiendo el patrón MVC para el desarrollo de este proyecto, nos encontramos con la siguiente estructura de ficheros:

- core - contiene los siguientes ficheros:
  - o autoload.php - encargado de la carga de los ficheros que necesitará utilizar la aplicación.
  - o config.php - almacenará las configuraciones necesarias para el buen funcionamiento de la aplicación en forma de constantes.
  - o DB.php - definirá algunas funciones relacionadas con la conexión con la base de datos o la ejecución de sentencias, de manera que haya necesidad de repetir todo el código correspondiente en tantas ocasiones.
  - o helpers.php - con el mismo objetivo que el anterior, reunirá aquellas funciones que ejecutarán acciones repetitivas.
  - o Routes.php - encargado de determinar el controlador al que se desea acceder y la función que desea llevar a cabo.
- views - contiene las vistas de la aplicación que se puede traducir en cada una de las páginas de la aplicación a la que desea acceder el usuario, modulando y agrupando algunos de sus elementos con el fin de evitar la repetición excesiva de código y que se organizarán en los siguientes directorios:
  - o assets - contendrá las carpetas “img”, “css” y “js” para almacenar los ficheros a los que hacen referencia sus nombres.
  - o contents:
    - forms/update - formularios de actualización que se incluirán en las vistas cuando sea necesario.
    - lists - listado a mostrar en las páginas oportunas.
    - selects - menús desplegables susceptibles de repetición utilizados principalmente en formularios.

- msg.php - contiene una función para generar un array que contendrá los mensajes generados por el servidor, de manera que se mantenga una estructura determinada y la información necesaria almacenada.
- modules - contendrá las diferentes partes en las que se suele dividir una página web y, por tanto, se repetirá en todas ellas (o la mayoría):
  - header.php - incluye desde la declaración doctype hasta la cabecera o elemento header de la página.
  - msg.php - cuadro de mensajes que aparecerá en todas las páginas.
  - navPerfil.php.php
  - nav.php
  - footer.php - contiene el pie de página o elemento footer de la página, además de los scripts (ficheros JS) y las etiquetas de cierre de los elementos body y html.
- models - contiene las clases (o entidades que forman el sistema) y que permitirá la comunicación con la base de datos.
- controllers - encargados de cargar las vistas tras obtener todos los datos necesarios, haciendo uso de los modelos o clases.
- index.php

## 4 PROCESO DE DESARROLLO

### 4.1 CICLO DE VIDA UTILIZADO

Para la realización de este proyecto se va a utilizar un ciclo de vida iterativo, ya que así podremos ir refinando los requisitos e incrementar funcionalidad en cada una de las iteraciones o versiones.

Este documento presenta las cuatro primeras iteraciones llevadas a cabo, presentando al final del mismo algunas de las mejoras y ampliaciones funcionales que se quieren integrar en el proyecto.

### 4.2 ORDEN DE DESARROLLO DE LAS FUNCIONALIDADES DEL PROYECTO

Para la realización de este proyecto se va a utilizar un ciclo de vida iterativo, ya que así podremos ir refinando los requisitos e incrementar funcionalidad en cada una de las iteraciones o versiones.

Este documento presenta las primeras iteraciones a desarrollar, presentando al final del mismo algunas de las ampliaciones funcionales y mejoras que se quieren integrar.

		Iteración 1	Iteración 2	Iteración 3	Iteración 4
O- Login y Perfil	O1	X			
	O2	X			
	O3	X			
	O4	X			
	O5	X			
A- Citas	A1	X			
	A2	X			
	A3	X			
	A4	X			
	A5	X			
R- Consultas	R1	X			
	R2	X			
	R3	X			
	R4	X			
	R5	X			
C- Clientes	C1	X			
	C2		X		
	C3	X			
	C4	X			
	C5		X		
	C6		X		
E- Empleados	E1	X			

	E2		X		
	E3	X			
	E4	X			
	E5		X		
	E6		X		
M- Administradores	M1	X			
	M2		X		
	M3	X			
	M4	X			
	M5		X		
	M6		X		
L- Laborales	L1			X	
	L2			X	
F- Festivos	F1			X	
	F2			X	
	F3			X	
	F4			X	
	F5			X	
S- Especialidades	S1				X
	S2				X
	S3				X

El orden para desarrollar cada una de estas funciones será dado por el orden de aparición de cada una de las páginas web que se mostrarán:

1. Página de login: acceder a la cuenta (cualquier rol), recuperar contraseña de clientes y crear cuenta cliente.
2. Primer menú de navegación: salir de la cuenta y ver el perfil de usuario.
3. Perfil del usuario: modificar los datos personales del usuario, recuperar la contraseña y eliminar la cuenta cliente.
4. Página de citas
5. Página de consultas
6. Página de clientes
7. Página de empleados
8. Página de administradores
9. Página de especialidades
10. Página de información de días laborales
11. Página de información de días festivos

Las páginas desde el punto 4 al 8 seguirán la misma estructura:

1. Añadir
2. Buscar
3. Listar
4. Modificar
5. Eliminar

6. Recuperar contraseña (solo para cliente, empleado y administrador)

Donde el orden de desarrollo será el siguiente:

1. Añadir
2. Listar
3. Eliminar
4. Buscar
5. Modificar
6. Recuperar contraseña

En el caso de las especialidades, las funciones relacionadas con las mismas serán elaboradas en último lugar, ya que se trata de una modificación o mejora de la aplicación que se desarrollará tras la aplicación base.

Por último, a pesar de estar presentes, la página principal solo mostrará (además del primer menú de navegación) los enlaces para navegar de una página a otra, al igual que el segundo menú de navegación. La única función de estos elementos es permitir la navegación de una página a otra.



## 5 PROCESO DE DESPLIEGUE

### 5.1 HOSTING WEB (000WEBHOST)

Como primera opción para el despliegue de la aplicación web MediControl utilizaremos un hosting web. En este caso se utilizará [000webhost](https://000webhost.com).

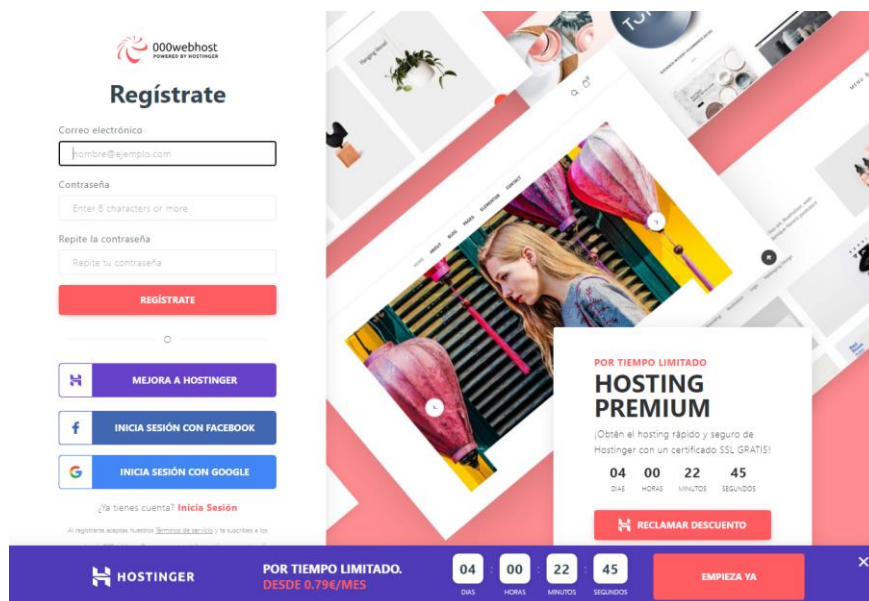
Seguiremos los siguientes pasos:

- Registro o login.
- Creación del sitio.
- Creación de la base de datos.
- Subida de ficheros.
- Modificación del fichero /core/config.php.
- Acceso a la aplicación.

#### 5.1.1 Registro y creación del sitio

Tras acceder a la página principal de 000webhost seguiremos los siguientes pasos para realizar nuestro registro:

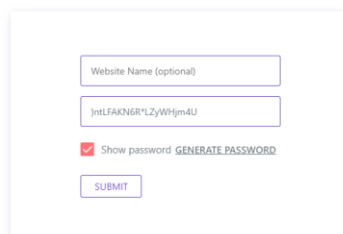
- Clic sobre “INICIA SESIÓN” en la parte superior derecha.
- Clic en “Regístrate” en la parte inferior del formulario de acceso o login.
- Rellenamos los campos del formulario de registro con nuestros datos.
- Clic sobre el botón “REGÍSTRATE”.
- Activamos la cuenta a través del enlace que se nos enviará a la dirección de correo electrónico que hallamos introducido.



A continuación, seguiremos los pasos mostrados tras el registro hasta llegar al formulario de creación de nuestro sitio web o proyecto. En este formulario indicamos el nombre de nuestro sitio y su contraseña, y continuaremos hacia el siguiente paso.

A great start is half the work

## Name Your Project

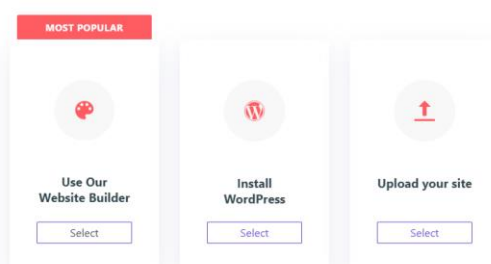


The screenshot shows a web form titled "Name Your Project". It contains two input fields: "Website Name (optional)" and a password field containing "jnlFAKNGR\*LZyWhjm4U". Below the password field is a checkbox labeled "Show password" which is checked, and a link "GENERATE PASSWORD". At the bottom is a "SUBMIT" button.

En el siguiente paso se nos propondrá la subida de los ficheros. Seleccionando esta opción se nos redirigirá al gestor de archivos que veremos en apartados siguientes.

Here is your tool box

## Choose your weapon of mass creation

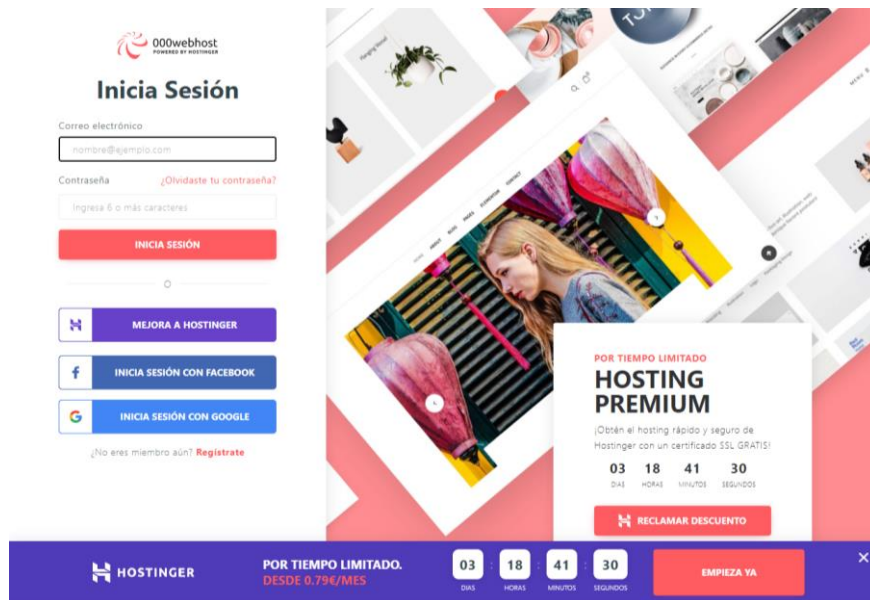


The screenshot shows a web form titled "Choose your weapon of mass creation". It features three cards under the heading "MOST POPULAR". The first card is "Use Our Website Builder" with a red puzzle piece icon and a "Select" button. The second card is "Install WordPress" with a red WordPress logo icon and a "Select" button. The third card is "Upload your site" with a red upload icon and a "Select" button.

## 5.1.2 Acceso y creación del sitio

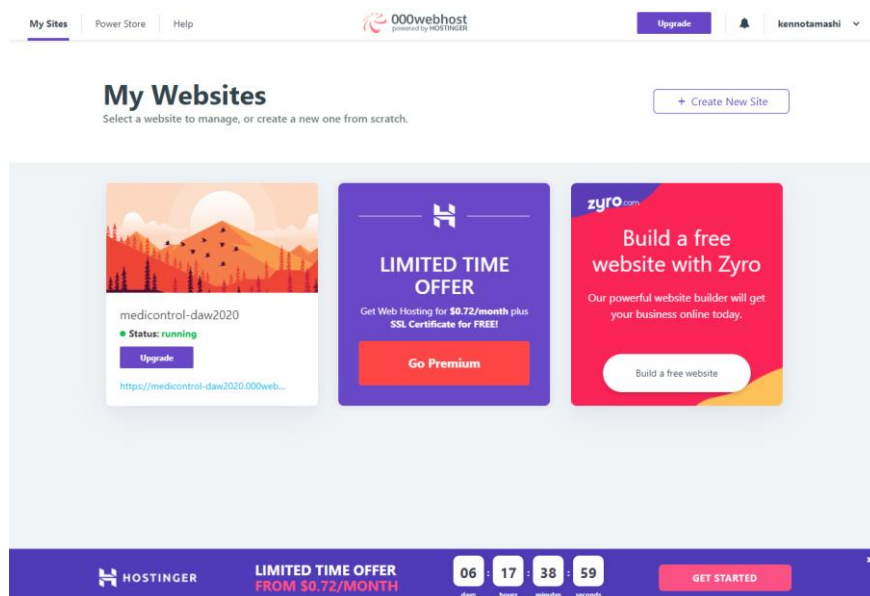
Si nos encontramos registrados previamente seguiremos los siguientes pasos:

- Clic sobre “INICIA SESIÓN” en la parte superior derecha.
- Rellenamos los campos del formulario de acceso con nuestros datos.
- Clic sobre el botón “INICIAR SESIÓN”.



Para la creación de un sitio:

- Clic sobre “Create New Site”.
- Indicamos nombre y contraseña del nuevo sitio.



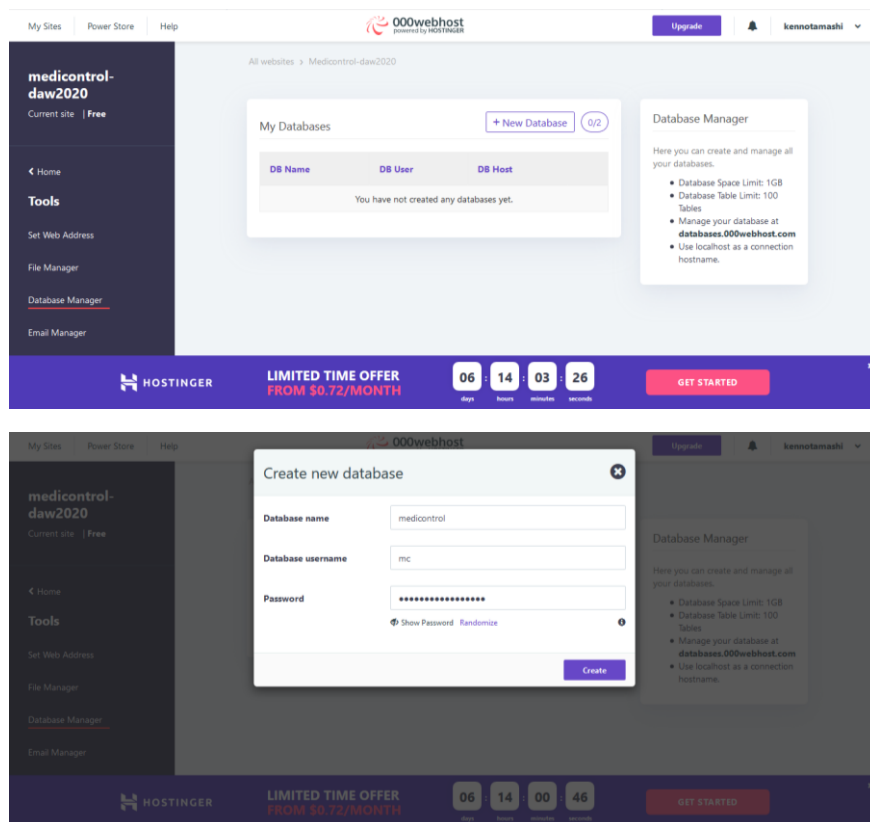
### 5.1.3 Creación de la base de datos

Antes de continuar con la subida de los ficheros del proyecto, veremos cómo crear la base de datos e importar las tablas a utilizar.

Partiendo de la página principal de nuestro cPanel (donde creamos nuestro sitio web), seleccionamos “Manage Website” tras colocar el puntero del ratón sobre la imagen del sitio.

Seguiremos entonces los siguientes pasos:

- Seleccionamos “Tools” en el menú izquierdo.
- Seleccionamos “Database Manager”.
- Hacemos clic sobre “New Database”.
- Rellenamos el formulario de creación de base de datos con el nombre, usuario y contraseña de la misma.

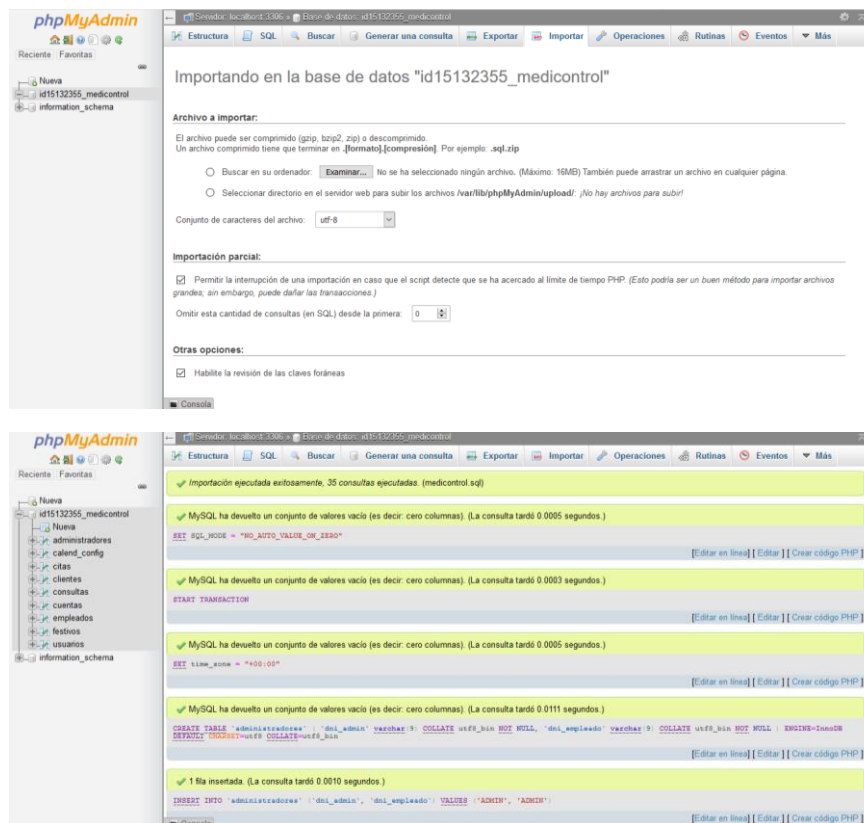


Los datos para acceder a la base de datos se encontrarán en esta misma página, donde se listan las bases de datos que tenemos creadas.

Por último, para importar las tablas de la base de datos que utilizará la aplicación:

- Hacemos clic sobre “Manage” en la base de datos que vayamos a utilizar.
- Seleccionamos “PhpMyAdmin”.

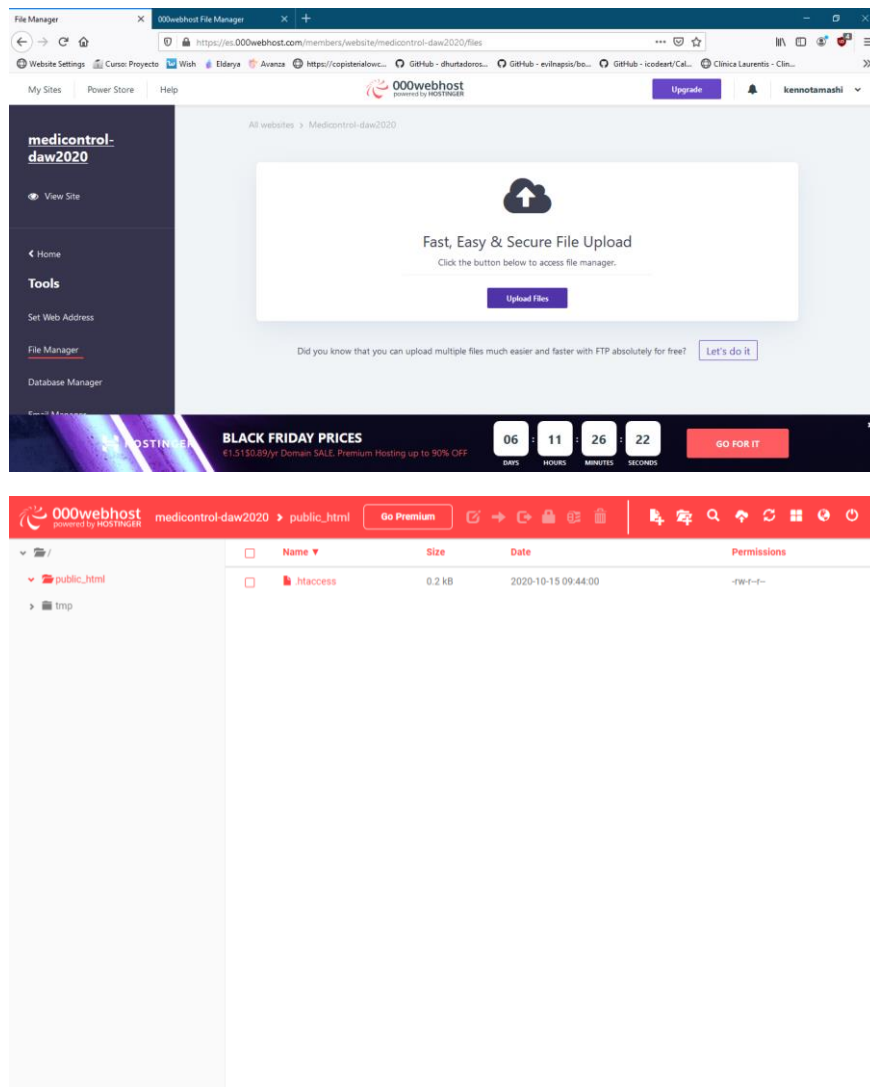
- Seleccionamos nuestra base de datos en el listado situado a la izquierda de la ventana.
- Hacemos clic sobre “Importar” en el menú superior.
- A través de “Examinar...” buscamos el fichero .sql con las instrucciones para la creación de las tablas e inserción de datos iniciales.
- Hacemos clic sobre “Continuar” al final de la página.



### 5.1.4 Subida de ficheros

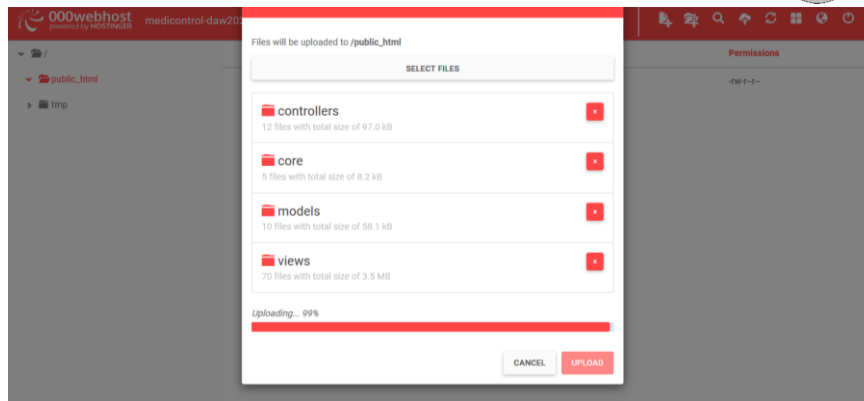
Para la carga de los ficheros del proyecto a 000webhost, procedemos de la siguiente forma:

- Seleccionamos “Tools” en el menú izquierdo.
- Seleccionamos “File Manager”.
- Hacemos clic sobre “Upload Files”.
- Accedemos a la carpeta “public\_html”.
- Arrastramos los archivos del proyecto hacia la ventana.
- Procedemos a la carga de los ficheros con “UPLOAD”.



The screenshot shows the 000webhost File Manager interface. The top navigation bar includes 'My Sites', 'Power Store', and 'Help'. The left sidebar contains 'Tools' and 'File Manager'. The main area displays a 'Fast, Easy & Secure File Upload' message with an 'Upload Files' button. Below this, a 'BLACK FRIDAY PRICES' banner is visible. The bottom section shows the file manager interface with a table of files and folders.

Name	Size	Date	Permissions
public_html			
htaccess	0.2 kB	2020-10-15 09:44:00	-rw-r--r--



### 5.1.4.1 *Modificación del fichero /core/config.php*

Para finalizar, necesitamos modificar los datos para la conexión a la base de datos. Estos datos se encuentran en `public_html/core/config.php`.

Haciendo doble clic sobre este fichero podremos editarlo. La porción de código a modificar es la siguiente:

```
// Parámetros de configuración de la BD
define('HOST', 'localhost');
define('DBNAME', 'medicontrol');
define('USER', 'medicontrol');
define('PASS', '');
```

La información a usar es la que se indicó en el “Database Manager” tras crear la base de datos:

- En ‘HOST’ sustituimos ‘localhost’ por lo indicado en “DB Host”.
- En ‘DBNAME’ sustituimos ‘medicontrol’ por lo indicado en “DB Name”.
- En ‘USER’ sustituimos ‘medicontrol’ por lo indicado en “DB User”.
- En ‘PASS’ indicamos la contraseña utilizada al crear a base de datos.

## 5.1.5 Acceso a la aplicación

Para acceder a la aplicación y comenzar a utilizarla accedemos a la página principal de 000webhost y, en la parte inferior del recuadro que nos permite gestionar nuestro sitio, hacemos clic sobre un enlace similar a siguiente:

<https://medicontrol-daw2020.000webhostapp.com/>

### 5.1.5.1 *Usuario administrador por defecto*

Una vez nuestra aplicación está configurada y lista para usarse, es importante cambiar las credenciales de acceso del usuario administrador por defecto, las cuales son:

Usuario: ADMIN

Contraseña: admin

Para cambiar estos datos accedemos a la aplicación y al perfil del usuario a través de la opción del menú de navegación “Ver perfil”.

En el apartado de datos personales podremos cambiar su nombre y seguido de este bloque, cambiar la contraseña. Añadimos también una dirección de correo electrónico ya que será esta la que se utilizará para el reenvío de contraseñas.

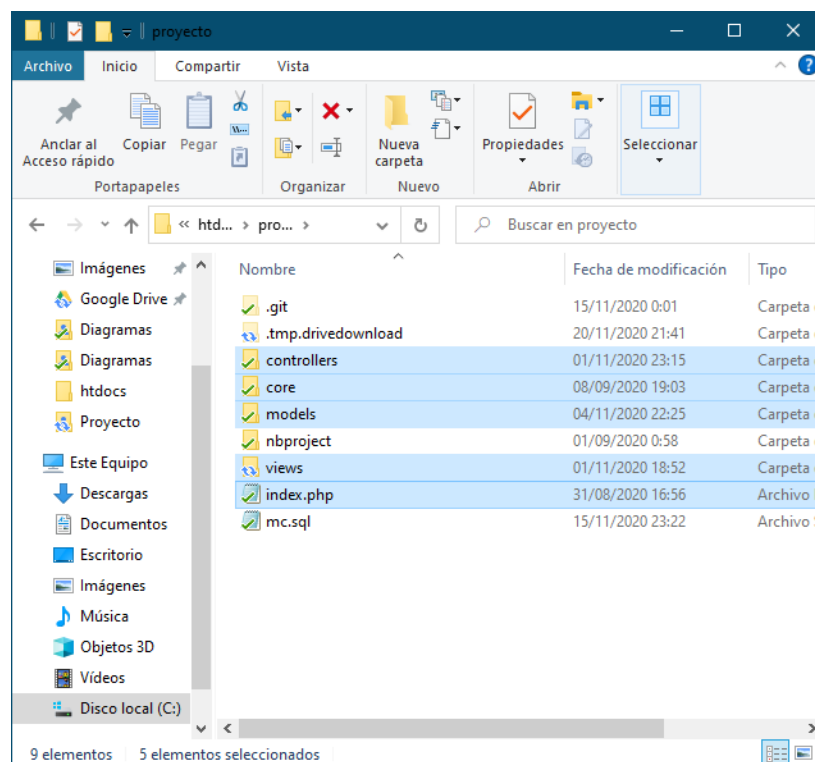
Por otra parte, también es importante configurar los horarios del calendario laboral para poder habilitar los días de citas. Esto se realizará a través de una cuenta administrador y en la opción “Calendario” del menú de navegación.



## 5.2 SERVIDOR LOCAL (XAMPP)

Para instalar y utilizar la aplicación en modo local, haremos uso XAMPP, que consiste principalmente en el servidor web Apache, gestor de base de datos MySQL, servidor FTP (además de intérpretes para scripts en PHP y Perl).

Se deberá descargar e instalar XAMPP. Una vez instalado, nos ubicaremos dentro del directorio “xampp/htdocs” y aquí copiaremos en una carpeta con el nombre de nuestro proyecto los ficheros del mismo.



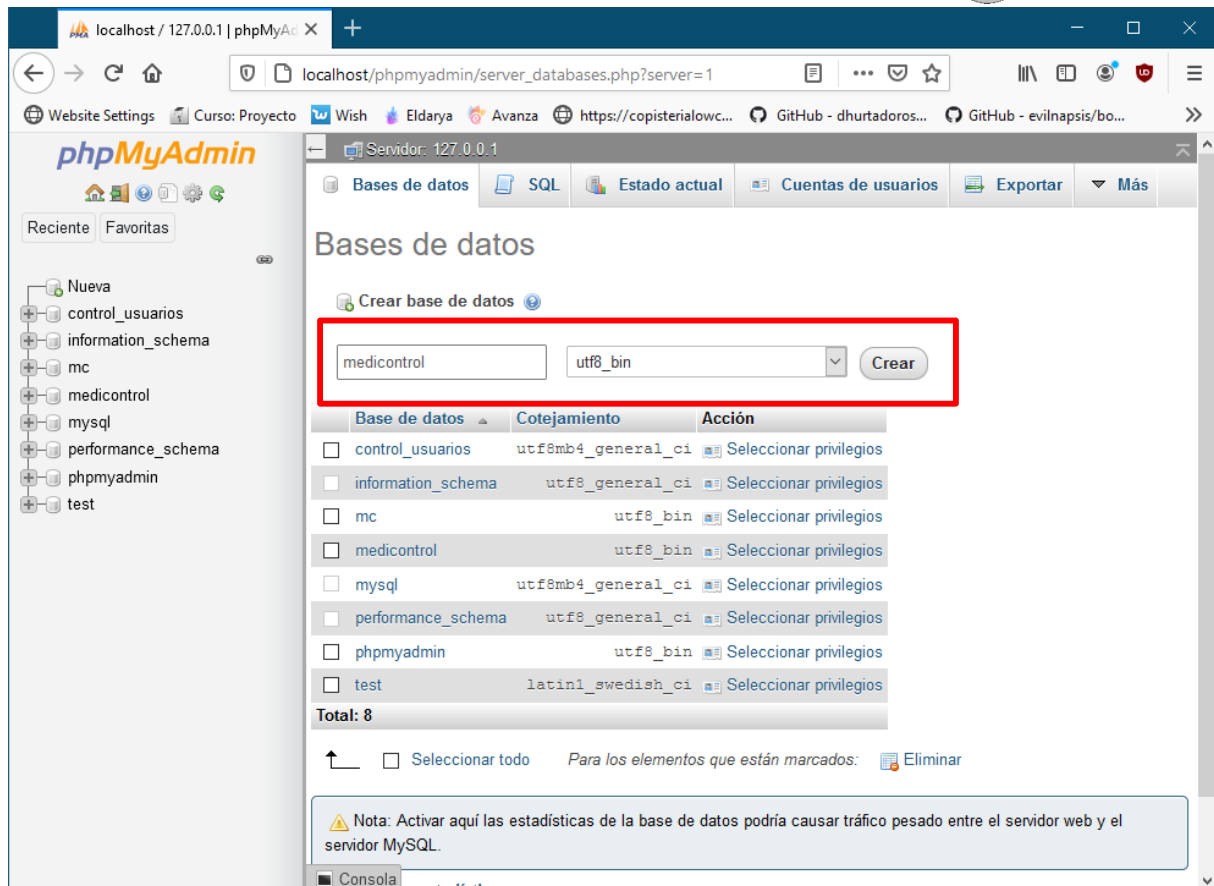
Los ficheros necesarios son los que se encuentran seleccionados en la imagen superior (index.php, controllers, core, models, views).

### 5.2.1 Creación de la base de datos

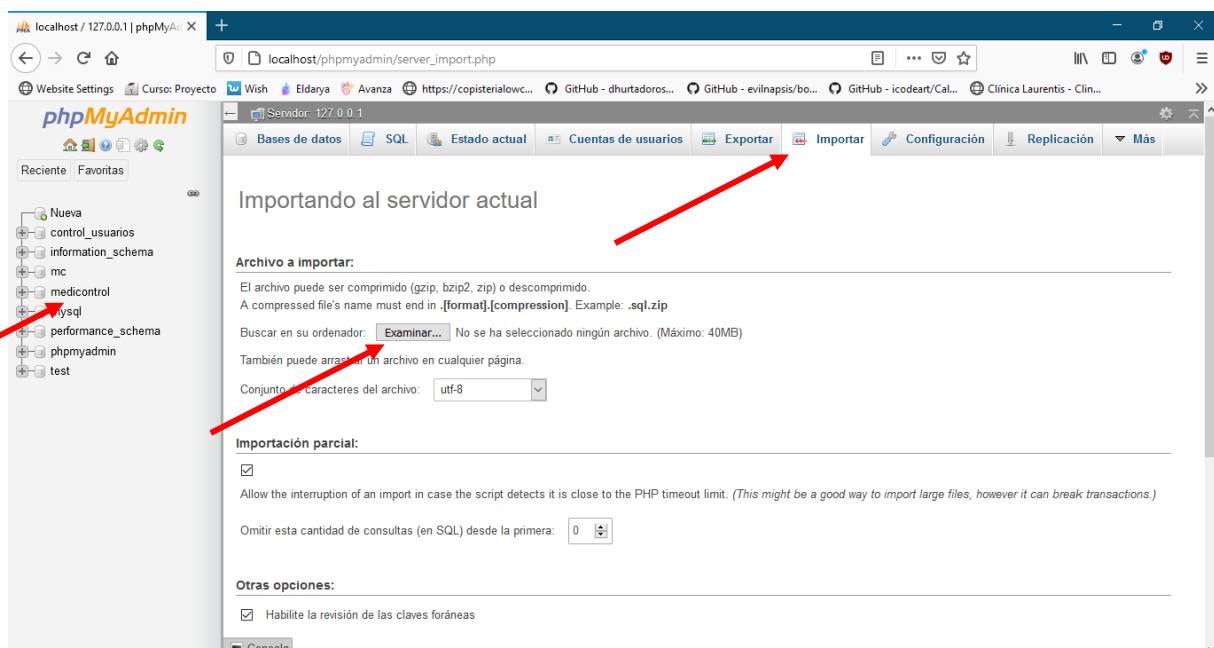
El fichero mc.sql será el script SQL utilizado para importar las tablas y datos iniciales de la aplicación en la base de datos.

Para crear la base de datos accedemos desde cualquier navegador a la dirección <http://localhost/phpmyadmin> (los servicios de apache y mysql deben estar previamente activados desde el panel de control de XAMPP).

A continuación, pulsamos sobre “Nueva” en el inicio del listado de bases de datos en el panel izquierdo para crear una nueva base de datos. Introducimos el nombre de la misma y seleccionamos la codificación “utf8\_bin”. Para finalizar, pulsamos sobre “Crear”.

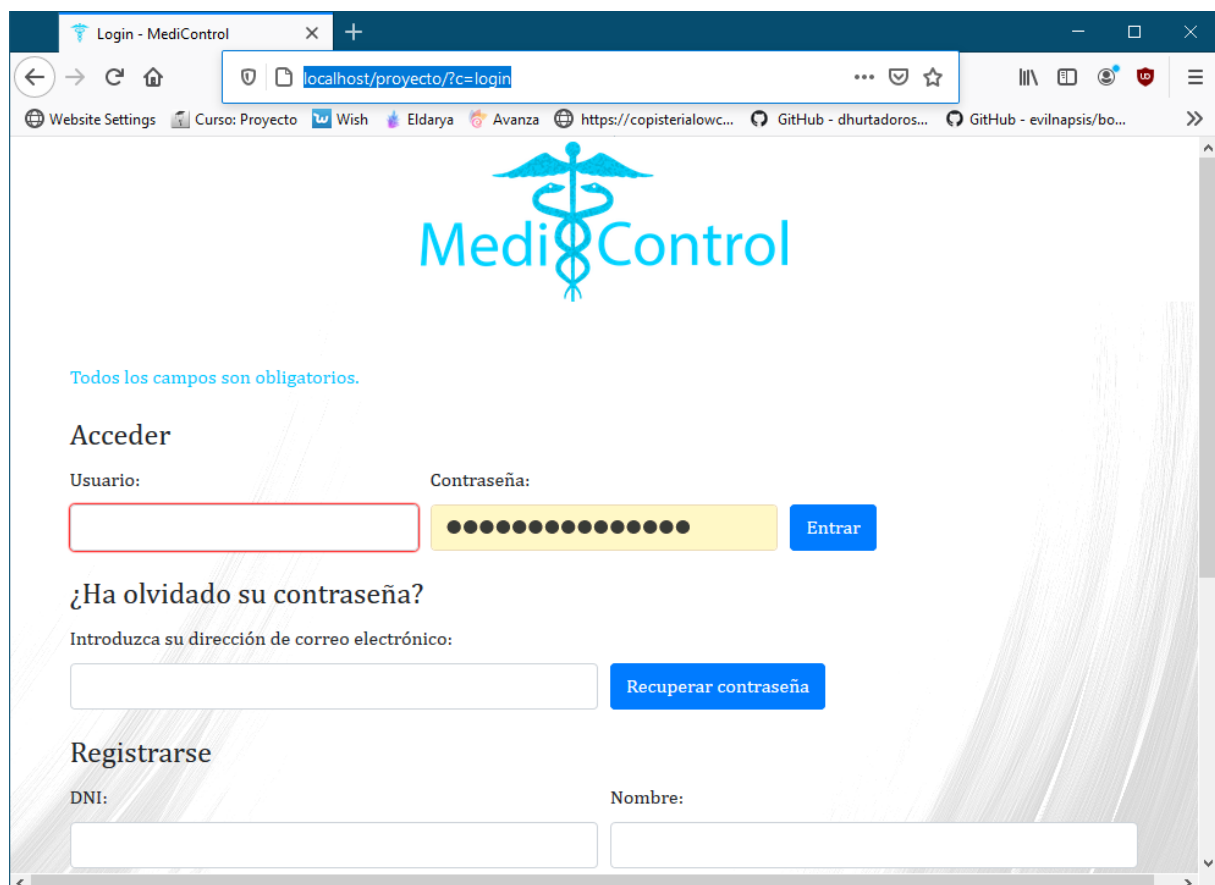
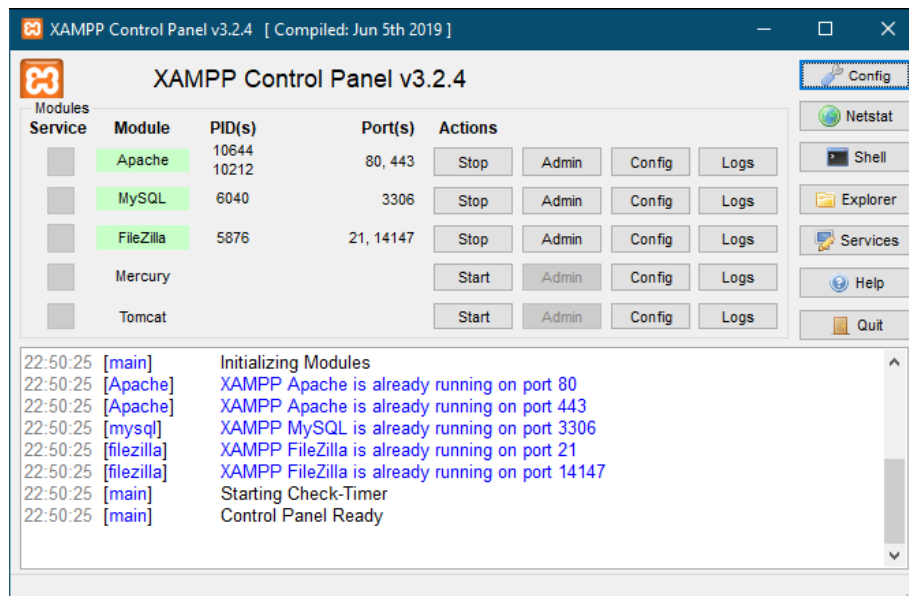


Para importar las tablas desde el script incluido en la carpeta del proyecto, seleccionamos la base de datos recién creada y pulsamos sobre Importar en el menú superior. A través del botón “Examinar...” en la primera sección de la página titulada “Archivo a importar”, buscamos el script SQL con las sentencias a ejecutar y pulsamos “Continuar” al final de la página.



### 5.2.2 Acceso a la aplicación

Para acceder a la aplicación deberemos abrir el panel de control de XAMPP y activar las opciones Apache, MySQL y FileZilla. Entonces, accederemos a la dirección [http://localhost/nombre\\_proyecto](http://localhost/nombre_proyecto) desde cualquier navegador.



## 5.3 APACHE + MYSQL + FTP (LINUX)

A continuación, veremos la instalación de los componentes necesarios para el despliegue de la aplicación por separado y en un sistema operativo Linux.

Antes de comenzar, actualizaremos el sistema con:

```
apt-get update  
apt-get upgrade
```

### 5.3.1 Apache

#### 5.3.1.1 *Instalación*

Comenzamos con la instalación del servidor web de Apache, que será el que nos permita crear nuestro sitio y alojar los ficheros del proyecto.

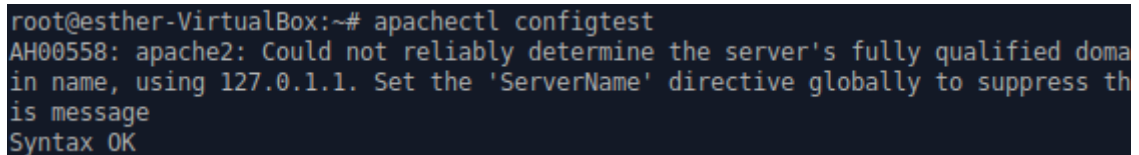
Los pasos a realizar serán los enumerados a continuación:

1. Instalación de Apache:

```
apt-get install apache2
```

2. Comprobar la configuración del servicio (debe aparecer Syntax OK):

```
apachectl configtest
```



```
root@esther-VirtualBox:~# apachectl configtest  
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message  
Syntax OK
```

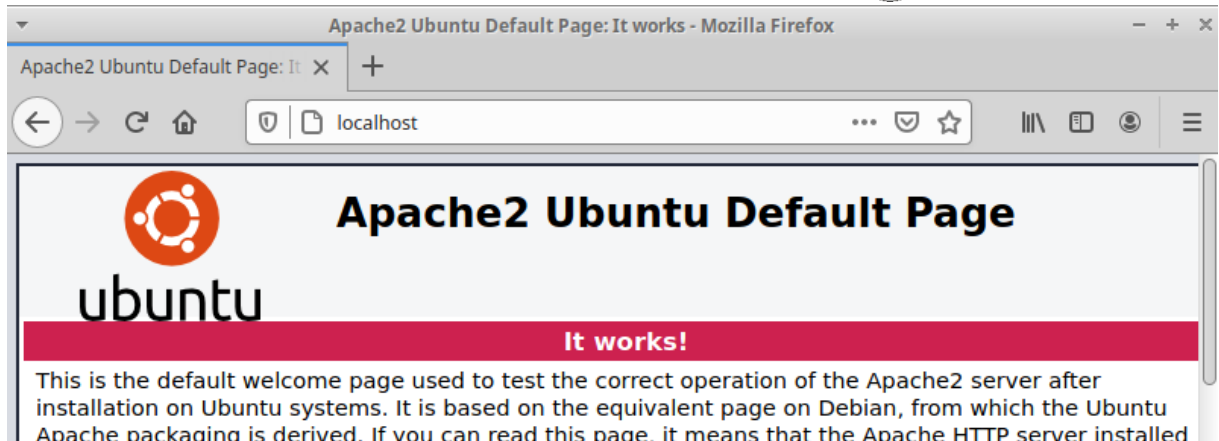
3. Comprobar el estado del servicio:

```
service apache2 status
```

4. Activar el servicio si no se encuentra activo:

```
service apache2 start
```

5. Comprobar que Apache funciona a través de un navegador web, accedemos a la dirección <http://localhost> desde el equipo donde se encuentra activa o indicamos la IP del servidor en lugar de localhost si accedemos desde otro equipo (en cualquier caso debe aparecer la página por defecto de Apache con el mensaje “It works!”).



## 5.3.1.2 Creación del sitio

Para crear nuestro sitio web, donde alojar la aplicación de proyecto MediControl, realizaremos los siguientes pasos:

1. Mover la carpeta del proyecto a /var/www/html:

```
mv /home/esther/Descargas/proyecto /var/www/html/medicontrol
```

2. Si la carpeta se encuentra comprimida, movemos este fichero a /var/www/html, donde la descomprimimos (en este caso utilizaremos un fichero .zip), es importante ubicarnos primero en el directorio donde queremos almacenar el contenido del fichero comprimido:

```
mv /home/esther/Descargas/proyecto.zip /var/www/html
cd /var/www/html
unzip proyecto.zip
```

3. Cambiar el nombre de esta carpeta por el nombre del sitio (en el primer caso, asignamos el nuevo nombre al mismo tiempo que cambiamos su ubicación), suponiendo que nos encontramos en /var/www/html y la carpeta obtenido tras la descomprensión se denomina “proyecto”:

```
mv proyecto medicontrol
```

4. Configurar el virtualhost, creando el archivo de configuración de nuestro sitio en /etc/apache2/sites-available (en este caso copiaremos el fichero de configuración del sitio por defecto y lo modificaremos):

```
cd /etc/apache2/sites-available
cp 000-default.conf medicontrol.conf
nano medicontrol.conf
```

```
<VirtualHost 10.0.2.15:80>
# The ServerName directive sets the request scheme, hostname and port to
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/medicontrol

ServerName www.medicontrol.com
ServerAlias medicontrol.com medicontrol.es www.medicontrol.es
DirectoryIndex index.php
```

En la etiqueta VirtualHost en la primera línea del documento indicamos la IP del servidor y el puerto que utiliza apache (por defecto será el puerto 80).

Por último, modificamos el resto del fichero para modificar la ubicación de nuestro sitio en “DocumentRoot” y añadir las directivas “ServerName”, “ServerAlias” y “DirectoryIndex”.

5. Hacer que las peticiones del dominio vayan a nuestro servidor, modificando el fichero /etc/hosts:

```
nano /etc/hosts
```

GNU nano 4.8	hosts	Modificado
127.0.0.1	localhost	
127.0.1.1	esther-VirtualBox	
10.0.2.15	www.medicontrol.com	

En este fichero añadimos la IP de nuestro servidor y el “ServerName” indicado en el paso anterior.

6. Activar el sitio web y reiniciar el servidor:

```
a2ensite medicontrol.conf
service apache2 restart
```

### 5.3.1.3 Instalación de PHP

Para poder utilizar la aplicación MediControl correctamente debemos instalar el módulo PHP de Apache, ya que se utilizará este lenguaje:

```
apt-get install php
apt-get install libapache2-mod-php
```

En este punto, la aplicación mostrará varios errores. En este punto veremos cómo solucionar el error que indica que las funciones mb\_x no se encuentran definidas. Para ello, solo tenemos que instalarlas:

```
apt-get install php-mbstring
```

## 5.3.2 Instalación de MySQL

Para la instalación del servidor de bases de datos MySQL:

```
apt-get install mysql-server
```

A continuación, configuramos nuestro servidor ejecutando el script de seguridad, donde deberemos indicar la contraseña que deseemos y aceptar en las preguntas siguientes de manera que se eliminarán algunos usuarios anónimos y la base de datos de prueba, y se deshabilitarán los inicios de sesión root remotos:

```
mysql_secure_installation
```

Comprobamos el estado del servicio y lo activamos si no se encuentra ya activo:

```
service mysql status  
service mysql start
```

En este punto, solucionaremos dos errores que aparecerán en relación con la base de datos.

En primer lugar, se nos indicará que no se encuentra instalado el driver necesario para utilizar conexiones PDO. Teniendo en cuenta la versión de MySQL que hemos instalado, utilizaremos el siguiente comando:

```
apt-get install php7.4-mysql
```

Y reiniciamos el servidor Apache para aplicar los cambios:

```
service apache2 restart
```

El último error que deberemos configurar mediante una instalación será aquel que nos indica un acceso denegado. Para solucionarlo, actualizaremos la contraseña del usuario root (es el que utilizamos en este caso) y reiniciaremos el servidor apache:

```
mysql -u root  
flush privileges;  
alter user 'root'@'localhost' identified by "";  
set password for 'root'@'localhost' = "";  
update mysql.user set authentication_string = "" where user = 'root' and host =  
'localhost';  
service mysql restart
```

### 5.3.2.1 Creación de la base de datos e importación de tablas y datos

Para conectarnos al servidor utilizaremos el usuario root con indicación de su contraseña (con -u indicamos el usuario, con -p indicamos la contraseña del usuario):

```
mysql -u root -p
```



Para crear la base de datos ejecutamos el comando SQL (en este caso el nombre de la base de datos será “medicontrol”):

```
create database medicontrol;
```

Comprobamos su creación listando las bases de datos existentes con:

```
show databases;
```

```
root@esther-VirtualBox:/# mysql --user=root --password=
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.22-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database medicontrol;
Query OK, 1 row affected (0.25 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| medicontrol |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)
```

Por último, salimos del servidor (exit) y para importar las tablas de la base de datos junto con sus datos, ejecutamos el siguiente comando, indicando al final la ruta donde se encuentra el fichero con extensión .sql a importar:

```
mysql -u root -p medicontrol /var/www/html/medicontrol/mc.sql
```

Comprobamos el resultado volviéndonos a conectar al servidor, seleccionando la base de datos a utilizar y listando sus tablas:

```
mysql -u root -p
use medicontrol;
show tables;
```



```
mysql> use medicontrol;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_medicontrol |
+-----+
| administradores       |
| citas                 |
| clientes              |
| consultas             |
| cuentas               |
| disponibilidad        |
| empleados             |
| especialidades        |
| fechas                |
| horarios              |
| usuarios              |
+-----+
11 rows in set (0.00 sec)
```

### 5.3.2.2 Configuración de los de conexión de la aplicación

Para finalizar, hay que configurar el fichero `/var/www/html/medicontrol/core/config.php` para indicar el host, el nombre de la base de datos, el usuario con el que nos conectaremos a la base de datos y su contraseña:

```
/**
 * Parámetros de configuración de la BD
 */
define('HOST', 'localhost');
define('DBNAME', 'medicontrol');
define('USER', 'root');
define('PASS', '');
```

### 5.3.3 Instalación de FTP

Por último, instalaremos un servidor FTP con:

```
apt-get install proftpd
```

Comprobamos que se encuentre activo; en caso contrario lo activamos:

```
service proftpd status
service proftpd start
```

No será necesaria ninguna configuración adicional para este servidor.

## 6 PROPUESTA DE MEJORA O TRABAJOS FUTUROS

---

En este proyecto se ha desarrollado una aplicación con unas funciones que podemos considerar básicas. Se pretende aumentar la funcionalidad de MediControl para permitir:

- La vista de los formularios de adición cuando el usuario lo solicite (al igual que los formularios de modificación, al hacer clic sobre un botón se podrá acceder al formulario).
- El registro de los empleados como usuarios cliente.
- La gestión de sucursales y asignación de clientes y usuarios a cada sucursal.
- El envío de un email de confirmación al modificar una cita.
- La consulta de sucursales con los datos de contacto de los mismos, en forma de un listado disponible para todos los usuarios.
- El contacto con el centro a través de un formulario o un sistema de mensajes.
- Limpiar los filtros de búsqueda (vaciar los campos de búsqueda y rehacer la búsqueda sin filtros) añadiendo otro botón.
- El control sobre los menores de edad, añadiendo para el registro de los mismos (al introducir la fecha de nacimiento y calcular la edad) un nuevo campo donde introducir los datos de algún tutor.
- La descarga de los historiales de consultas en formato excel o pdf.
- La asignación de horarios a los empleados y cálculo de la disponibilidad para las citas a través de ellos.
- La generación de citas para usuarios no registrados.
- La creación de consultas sin necesidad de cita previa.
- La paginación de resultados.

## 7 BIBLIOGRAFÍA

---

<https://www.pildorasinformaticas.es/>

<https://es.stackoverflow.com/>

HDMagazine – Manual de MVC

<https://aspgems.com/metodologia-de-desarrollo-de-software-ii-modelo-de-diseno-iterativo/>

<https://openwebinars.net/>

<https://www.w3schools.com/>