Desarrollo de una aplicación web y base de datos para el seguimiento de pacientes con problemas de movilidad cervical

Borja González Díaz

Septiembre 2017

A toda mi familia y en especial a mi padre Peps.

**Resumen**

Con la llegada de las nuevas tecnologías al mercado la mayoría de ámbitos profesionales se están viendo forzados a adaptarse mediante la digitalización de todos sus sistemas, actualmente manuales y prácticamente obsoletos. Quedarse atascado en estos sistemas manuales significaría una gran desventaja a la hora de competir contra otros sistemas digitalizados, que son mucho más rápidos y eficientes, con una necesidad nula o mínima de mantenimiento. Esta disyuntiva también se aplica a los sistemas médicos, que se pueden beneficiar enormemente de la digitalización de la información relativa a los pacientes, permitiendo análisis más avanzados y una mayor agilidad para tratar a los pacientes.

Con un enfoque moderno a las aplicaciones web he construido una aplicación web y su correspondiente base de datos que permiten al usuario acceder rápidamente a los datos de los pacientes y mostrando toda la información disponible de estos.

Utilizando varios lenguajes de programación, plataformas y sistemas he conseguido satisfacer las necesidades dicha aplicación, cumpliendo así sus requisitos.

**Agradecimientos**

Tras estos años de carrera …

**Índice general**

Introducción 6

Funcionamiento de la aplicación web 6

Estado del arte 6

Tecnologías 6

Frameworks y lenguajes 6

Javascript 6

CSS 6

HTML 6

Despliegue de aplicaciones 6

Bases de Datos 6

Diseño 6

Descripción del problema 7

Requisitos 7

Requisitos Funcionales 7

Requisitos no Funcionales 7

Casos de uso 7

Matriz de trazabilidad 7

Arquitectura del sistema 7

Especificación de la API 7

Implementación 7

# Introducción

## Funcionamiento de la aplicación web

El objetivo de esta aplicación web es facilitar el acceso y visualización de datos de movimientos de pacientes.

La página permite el acceso a una base de datos de pacientes de la que se puede consultar datos de movimientos (Sagital, Coronal y Transversal). También es posible consultar la evolución con el tiempo de estos movimientos y ver si están dentro de unos parámetros de normalidad, dependiendo del sexo del paciente.

En la sección de pacientes se listarán todos los pacientes disponibles y podremos acceder a sus datos presionando en el icono del gráfico de cada paciente. También tenemos disponible la opción de añadir o borrar pacientes.

# Estado del arte

## Tecnologías

## Frameworks y lenguajes

### Javascript

### CSS

### HTML

## Despliegue de aplicaciones

## Bases de Datos

# Diseño

En este capítulo se intenta analizar en profundidad el problema que se pretende resolver, intentando abarcar la mayor cantidad de información posible y estudiando los principales aspectos que habrían de cubrirse en el desarrollo del proyecto. En este punto deberían surgir las características principales del sistema a desarrollar, por lo que se establecen también los casos de uso que deberá poder efectuar el sistema resultado del proyecto para cumplir los requisitos que se establezcan.

## Descripción del problema

El objetivo principal es construir una aplicación web que permita el acceso de uno o más usuarios o clientes, y que estos puedan trabajar con la base de datos obteniendo información de los pacientes. El problema se podría dividir en varios apartados, diseño de la arquitectura del sistema para satisfacer las necesidades descritas previamente, implementación de dicho sistema, despliegue de las aplicaciones y pruebas.

## Requisitos

La fase de análisis incluye el estudio y la definición de los requisitos del sistema. Los requisitos definen en detalle los servicios que el sistema debería poder ofrecer, identificando las restricciones y características de este.

### Requisitos Funcionales

RF1 – El sistema debe permitir el acceso a la aplicación web desde cualquier dispositivo.

RF2 – Una vez dentro la aplicación web, el sistema debe mostrar una sección de inicio con una breve descripción de las distintas funcionalidades de la pagina.

RF3 – El sistema ofrecerá un listado de los clientes con su nombre, apellidos y un link para acceder a sus datos de movimiento.

RF4 – El usuario tendrá la opción de añadir pacientes, saltando un error en el caso de que el usuario no rellene algún dato solicitado.

RF5 – El usuario tendrá la opción de borrar pacientes, borrando así todos sus datos asociados.

RF6 – Para cada paciente, el sistema mostrará una tabla con los tres movimientos disponibles y una opción de mostrar la evolución de cada movimiento.

RF7 – El sistema debe proporcionar a los clientes la opción de añadir datos de movimientos con una fecha asociada. Solo se admitirán ficheros CSV.

RF8 - El sistema debe proporcionar a los clientes la opción de borrar datos de movimientos.

RF9 – El sistema deberá ser capaz de mostrar los datos de movimientos y los datos de evolución mediante gráficos animados e interactivos. Dentro de los gráficos de evolución el sistema marcará unos límites de normalidad dependiendo del sexo del paciente.

RF10 – El sistema permitirá observar la progresión del movimiento con el tiempo.

RF11 – El sistema deberá ser capaz de almacenar los datos de los pacientes entre sesiones.

### Requisitos no Funcionales

## Casos de uso

Los casos de uso representan las diferentes posibles interacciones que pueden existir entre los diferentes actores dentro de un sistema. Los actores de un sistema son entidades externas al sistema que pueden interactuar con el. Los actores pueden ser humanos u objetos (servidores, aplicaciones, servicios...). Los actores que interactúan con nuestra aplicación son los siguientes:

* Usuarios del sistema a través del navegador de cualquier dispositivo.
* El cliente que representa al navegador con sus distintos archivos CSS, HTML y JavaScript.
* El servidor que atiende las peticiones del cliente/navegador y está formado por un archivo JavaScript con acceso a una base de datos SQLite.
* La base de datos SQLite que responde a peticiones del cliente a través del servidor.

CU1: Acceso a la página web.

El usuario introduce la dirección del servidor en su navegador accediendo instantáneamente al servidor.

Actores implicados: Usuario/cliente y Servidor.

Precondiciones: El sistema debe estar correctamente funcionando.

Secuencia de acciones:

1. El cliente accede a la aplicación web introduciendo la dirección IP del servidor en su navegador.

CU2: Acceso a la pestaña de pacientes/Listado de pacientes.

El usuario accede a un listado de pacientes.

Actores implicados: Usuario/cliente, Servidor y Base de Datos.

Precondiciones: El sistema debe estar correctamente funcionando.

Secuencia de acciones:

1. El cliente accede a la aplicación web introduciendo la dirección IP del servidor en su navegador.
2. El cliente accede a la sección de pacientes mediante la pestaña de pacientes.
3. El navegador solicita la lista de pacientes a la base de datos.
4. El cliente recibe la lista de pacientes y se la presenta al usuario en una tabla.

CU3: Añadir un paciente.

Actores implicados: Usuario/cliente, Servidor y Base de Datos.

Precondiciones: El sistema debe estar correctamente funcionando.

Secuencia de acciones:

1. El cliente se descarga los ficheros web introduciendo la dirección IP del servidor en su navegador.
2. El cliente accede a la sección de pacientes mediante la pestaña de pacientes.
3. El usuario introduce el nombre, apellidos y sexo del paciente y presiona el botón “Añadir”.
4. El navegador manda la petición, para añadir el paciente, al servidor.
5. El servidor añade el nuevo paciente a la base de datos, persistiendo los cambios.

Post-condiciones: El nuevo paciente tiene que ser mostrado inmediatamente en el listado de pacientes, sin que el usuario tenga que refrescar la página.

CU4: Borrar un paciente.

Actores implicados: Usuario/cliente, Servidor y Base de Datos.

Precondiciones: El sistema debe estar correctamente funcionando.

Secuencia de acciones:

1. El cliente se descarga los ficheros web introduciendo la dirección IP del servidor en su navegador.
2. El cliente accede a la sección de pacientes mediante la pestaña de pacientes.
3. El navegador solicita la lista de pacientes a la base de datos.
4. El cliente recibe la lista de pacientes y se la presenta al usuario en una tabla.
5. El usuario elige el paciente a borrar y presiona el botón de borrado correspondiente, ya que cada paciente tiene su botón de borrado.
6. El navegador pregunta al usuario si desea realizar esta operación, teniendo en cuenta que borraría todos los datos de movimientos asociados a ese paciente.
7. El usuario acepta la operación, teniendo la opción de cancelarla.
8. El navegador manda la petición para borrar el paciente al servidor.
9. El servidor borra al paciente y todos sus datos de movimientos asociados de la base de datos, persistiendo los cambios.

Post-condiciones: La lista de pacientes ha de actualizarse, mostrando todos los pacientes menos el borrado. Se actualiza de forma automática. Todos los movimientos asociados a este paciente tienen que desaparecer.

CU5: Listado de movimientos de un paciente.

El usuario accede a un listado de movimientos del paciente.

Actores implicados: Usuario/cliente, Servidor y Base de Datos.

Precondiciones: El sistema debe estar correctamente funcionando.

Secuencia de acciones:

1. El cliente se descarga los ficheros web introduciendo la dirección IP del servidor en su navegador.
2. El cliente accede a la sección de pacientes mediante la pestaña de pacientes.
3. El navegador solicita la lista de pacientes a la base de datos.
4. El cliente recibe la lista de pacientes y se la presenta al usuario en una tabla.
5. El cliente elige el paciente del que quiere mostrar sus datos de movimiento presionando el botón adecuado, ya que cada paciente tiene su propio botón para mostrar sus datos.
6. El navegador se posiciona en la pestaña de Datos.
7. El navegador solicita la lista de movimientos del paciente a la base de datos.
8. El navegador recibe la lista de movimientos y se la presenta al usuario en una tabla.

CU6: Añadir datos de movimiento.

El usuario añade un set de movimientos al paciente.

Actores implicados: Usuario/cliente, Servidor y Base de Datos.

Precondiciones: El sistema debe estar correctamente funcionando.

Secuencia de acciones:

1. El cliente se descarga los ficheros web introduciendo la dirección IP del servidor en su navegador.
2. El cliente accede a la sección de pacientes mediante la pestaña de pacientes.
3. El navegador solicita la lista de pacientes a la base de datos.
4. El cliente recibe la lista de pacientes y se la presenta al usuario en una tabla.
5. El cliente elige el paciente del que quiere mostrar sus datos de movimiento presionando el botón adecuado, ya que cada paciente tiene su propio botón para mostrar sus datos.
6. El navegador se posiciona en la pestaña de Datos.
7. El usuario selecciona un archivo, que solo podrá ser del tipo CSV, y una fecha de medición del set movimientos y presiona el botón de “Añadir datos”.
8. El navegador manda la petición para añadir los datos de movimiento al servidor.
9. El servidor añade el set de datos de movimientos al la base de datos persistiendo los cambios.

Post-condiciones: El nuevo set de datos tiene que ser mostrado inmediatamente en el listado de datos de movimientos, sin que el usuario tenga que refrescar la página.

CU6: Borrar datos de movimiento.

El usuario borra un set de movimientos del paciente.

Actores implicados: Usuario/cliente, Servidor y Base de Datos.

Precondiciones: El sistema debe estar correctamente funcionando.

Secuencia de acciones:

1. El cliente se descarga los ficheros web introduciendo la dirección IP del servidor en su navegador.
2. El cliente accede a la sección de pacientes mediante la pestaña de pacientes.
3. El navegador solicita la lista de pacientes a la base de datos.
4. El cliente recibe la lista de pacientes y se la presenta al usuario en una tabla.
5. El cliente elige el paciente del que quiere mostrar sus datos de movimiento presionando el botón adecuado, ya que cada paciente tiene su propio botón para mostrar sus datos.
6. El navegador se posiciona en la pestaña de Datos.
7. El navegador solicita la lista de movimientos del paciente a la base de datos.
8. El navegador recibe la lista de movimientos y se la presenta al usuario en una tabla.
9. El usuario elige los movimientos a borrar y presiona el botón de borrado correspondiente, ya que cada set de movimientos tiene su botón de borrado.
10. El navegador pregunta si desea realizar esta operación.
11. El usuario acepta la operación, teniendo la opción de cancelarla.
12. El navegador manda la petición para borrar los datos de movimiento al servidor.
13. El servidor borra el set de datos de movimiento del paciente de la base de datos y persiste los cambios.

Post-condiciones: La nueva lista de movimientos, sin el set de movimientos borrado, tiene que ser mostrada inmediatamente en el listado de movimientos, sin que el usuario tenga que refrescar la página.

CU7: Mostar datos del paciente.

El usuario muestra un movimientos del paciente.

Actores implicados: Usuario/cliente, Servidor y Base de Datos.

Precondiciones: El sistema debe estar correctamente funcionando.

Secuencia de acciones:

1. El cliente se descarga los ficheros web introduciendo la dirección IP del servidor en su navegador.
2. El cliente accede a la sección de pacientes mediante la pestaña de pacientes.
3. El navegador solicita la lista de pacientes a la base de datos.
4. El cliente recibe la lista de pacientes y se la presenta al usuario en una tabla.
5. El cliente elige el paciente del que quiere mostrar sus datos de movimiento presionando el botón adecuado, ya que cada paciente tiene su propio botón para mostrar sus datos.
6. El navegador se posiciona en la pestaña de Datos.
7. El navegador solicita la lista de movimientos del paciente a la base de datos.
8. El navegador recibe la lista de movimientos y se la presenta al usuario en una tabla.
9. El usuario elige el movimiento (Sagital, Coronal, Transversal) con su fecha asociada presionando en el botón adecuado.
10. El navegador muestra por pantalla un gráfico con los datos requeridos.

Post-condiciones: El navegador tiene que borrar el contenido del gráfico anterior (en el caso de que exista) y graficar los datos requeridos.

CU8: Mostar datos de evolución del paciente.

El usuario muestra la evolución de un movimiento del paciente.

Actores implicados: Usuario/cliente, Servidor y Base de Datos.

Precondiciones: El sistema debe estar correctamente funcionando.

Secuencia de acciones:

1. El cliente se descarga los ficheros web introduciendo la dirección IP del servidor en su navegador.
2. El cliente accede a la sección de pacientes mediante la pestaña de pacientes.
3. El navegador solicita la lista de pacientes a la base de datos.
4. El cliente recibe la lista de pacientes y se la presenta al usuario en una tabla.
5. El cliente elige el paciente del que quiere mostrar sus datos de movimiento presionando el botón adecuado, ya que cada paciente tiene su propio botón para mostrar sus datos.
6. El navegador se posiciona en la pestaña de Datos.
7. El navegador solicita la lista de movimientos del paciente a la base de datos.
8. El navegador recibe la lista de movimientos y se la presenta al usuario en una tabla.
9. El usuario elige la evolución del movimiento (Sagital, Coronal, Transversal) que quiere mostrar presionando el botón “Evolución del movimiento”.
10. El navegador muestra por pantalla un gráfico con los datos requeridos.

Post-condiciones: El navegador tiene que borrar el contenido del gráfico anterior (en el caso de que exista) y graficar los datos requeridos.

## Matriz de trazabilidad

## Arquitectura del sistema

## Especificación de la API

# Implementación

Pruebas

Resultados y conclusiones