



# CRONOMED

Tu tratamiento médico bajo control

Memoria TFG Desarrollo de Aplicaciones Web

## ABSTRACT

Con el avance de las tecnologías se ha perdido la agilidad que teníamos para memorizar y ese esfuerzo se ha delegado a las “máquinas”. Con esa necesidad, surge la idea de desarrollar una aplicación sencilla para controlar un tratamiento médico prescrito por un facultativo, visualizando la dosis y la pauta horaria de administración.

Para el desarrollo de la aplicación se utilizó el **patrón de arquitectura MVC o Modelo Vista Controlador**, utilizando como lenguaje de programación la tecnología Java, Tomcat como servidor de aplicación, MySQL como base de datos para la persistencia de datos, SQL como lenguaje de transacciones con la base de datos y phpmyadmin como interfaz de ésta, además de tecnologías HTML, CSS y Bootstrap en el diseño de las vistas.

Después de validar la aplicación, se concluye que puede gestionar de forma sencilla un tratamiento médico a nivel personal y grupal, pudiendo integrar tratamientos asociados a otros miembros de la familia.

---

With the advancement of technologies, the agility we had to memorize has been lost and that effort has been delegated to “machines”. With this need, the idea of developing a simple application arises to control a medical treatment prescribed by a doctor, visualizing the dose and the time schedule of administration.

For the development of the application, the **MVC architecture pattern or Model View Controller** was used, using Java technology as programming language, Tomcat as application server, MySQL as database for data persistence, SQL as transaction language with the database and phpmyadmin as its interface, as well as HTML, CSS and Bootstrap technologies in the design of the views.

After validating the application, it is concluded that it can easily manage medical treatment at a personal and group level, being able to integrate treatments associated with other family members.

## ÍNDICE

Abstract .....	1
Objetivos .....	4
Diagrama RFTP.....	5
Introducción.....	9
Justificación del Proyecto .....	11
Desarrollo .....	12
Estructura de carpetas del proyecto.....	13
Patrón de diseño.....	13
Entorno de desarrollo.....	17
Entorno de servidor.....	17
Gestión del proyecto.....	17
Control de versiones.....	18
Dependencias .....	19
Despliegue de la aplicación .....	20
Despliegue en servidor .....	20
Despliegue en servidor integrado eclipse IDE.....	21
Diseño .....	22
Diagrama de casos de uso.....	22
Diagrama de clases .....	31
Diagrama entidad relación .....	33
Diagrama de base de datos .....	35
Diagrama de flujo de navegación.....	37
Metodología.....	38
Diagrama de Gantt .....	39
Tecnologías .....	43
Eclipse .....	43

JSP.....	43
Java.....	43
Bootstrap.....	43
Apache-Tomcat.....	43
Phpmyadmin .....	44
Canvas.....	44
Draw.io.....	44
Pruebas y Documentación.....	45
Validación y pruebas.....	45
Manual de usuario .....	55
Conclusiones.....	56
Trabajos futuros .....	57
Bibliografía .....	58
Videos.....	58
Blogs.....	58
Referencias de páginas .....	58
Tesis y trabajos de grado.....	59
Glosario .....	60
Anexo I.....	61
Manual de Usuario.....	61

## **OBJETIVOS**

### **Como objetivo general:**

Og\_Gestionar un tratamiento médico prescrito por un facultativo, visualizando las pautas horarias y dosis de administración de una manera simple y amigable.

### **En cuanto a los objetivos específicos:**

Oe.1\_Diseñar y desarrollar una interfaz de usuario fácil de entender y de navegación intuitiva: crear una aplicación web fácil de usar que permita al usuario gestionar la información de un tratamiento médico de manera sencilla.

Oe.2\_Crear una base de datos para almacenar la información del tratamiento médico: diseñar y desarrollar una base de datos que permita el almacenamiento y consulta de la información concerniente a un tratamiento médico.

Oe.3\_Desarrollar un sistema de gestión del tratamiento: permitir al usuario realizar registros y seguimiento del tratamiento, y proporcionar horarios de recordatorios de medicación.

Oe.4\_Facilitar la gestión del tratamiento médico: permitir al usuario actualizar la situación de un tratamiento en cualquier momento, incluyendo los medicamentos recetados y las pautas en relación con la dosis y el horario en el que debe ser suministrados.

Oe.5\_Realizar pruebas para asegurar la calidad del desarrollo: incluir pruebas de usuario y de calidad de datos, para garantizar que la aplicación sea fiable y efectiva.

El objetivo final es mejorar la calidad y la satisfacción del usuario al proporcionar una solución tecnológica para la gestión de tratamiento médico.

Para conseguir estos objetivos, se ha desarrollado un diagrama RFTP, planteando los requisitos de la aplicación, funcionalidades que deben cumplir y, las tareas y pruebas para desarrollarla y garantizar la estabilidad.

## Diagrama RFTP

### R01 - Base de Datos y MVC

F01 - Diseño y relación de tablas

T01 - Diseñar tablas, estructura y diagrama entidad relación

F02 - Desarrollo modelo vista controlar y servlets

T01 - Desarrollo modelo vista controlador y organización de servlets

F03 - Conexión a la BD

T01 - Programar conexión a la bd

### R02 - Inicio de sesión

F01 - Ventana de inicio de sesión

T01 - Diseñar ventana de inicio de sesión

F02 - Botón de login

T01 - Programar verificación de usuario y contraseña

T02 - Programar servlet de login

F03 - Validaciones de usuario y contraseña

T01 - Validaciones de formulario inicio de sesión

P01 - Prueba con usuario y password correcto

P02 - Prueba con usuario y password incorrecto

P03 - Prueba con datos vacíos

### R03 - Registro de usuario

F01 - Ventana de registro de usuario

T01 - Diseñar ventana de registro de usuario

F02 - Botón de registro usuario

T01 - Programar inserción nuevo usuario en la tabla usuarios

T02 - Programar servlet de registro de nuevo usuario

F03 - Validaciones de registro de usuario

T01 - Validaciones de registro de usuario

P01 - Prueba con nombre de usuario con caracteres no admitidos

P02 - Prueba con email incorrecto

P03 - Prueba con email ya registrado

P04 - Prueba con password menor de 6 caracteres

P05 - Prueba con datos vacíos

P06 - Prueba de registro con datos correctos

#### **R04 - Cierre de sesión**

F01 - Botón cierre de sesión

T01 - Programar cierre de sesión de usuario

P01 - Prueba de cierre de sesión

#### **R05 - Principal usuario**

F01 - Ventana principal

T01 - Diseñar el home de la ventana principal del usuario

T02 - Programar login de usuario y asociarlo a la sesión

F02 - Navegación entre ventanas

T01 - Programar botones para navegación entre ventanas

#### **R06 - Opción Medicamentos**

F01 - Ventana de listar medicamentos

T01 - Programar método para listar los medicamentos

T02 - Programar servlet para listar los medicamentos

P01 - Prueba para listar medicamentos

#### **R07 - Opción Registro tratamiento**

F01 - Formulario de nuevo tratamiento

T01 - Diseñar ventana de nuevo tratamiento

F02 - Listar tratamientos asociados al usuario

T01 - Programar método listar tratamiento asociado al usuario con sesión

T02 - Programar servlet para listar tratamientos por usuario

P01 - Prueba para listar tratamientos del usuario

F03 - Agregar nuevo tratamiento

T01 - Programar inserción de nuevo tratamiento

T02 - Programar servlet para insertar nuevo tratamiento

P01 - Prueba agregar tratamiento sin completar formulario

F04 - Modificar tratamiento existente

T01 - Programar modificación de tratamiento existente

TO2 - Programar servlet para modificar y actualizar tratamiento existente  
PO1 - Prueba actualizar sin seleccionar tratamiento previamente

FO5 - Finalizar tratamiento  
TO1 - Programar finalización de tratamiento  
TO2 - Programar servlet para finalización de tratamiento  
PO1 - Prueba finalización de tratamiento

FO6 - Validaciones de tratamiento  
TO1 - Programar validaciones de tratamiento  
PO2 - Prueba con nombre de paciente con caracteres especiales  
PO2 - Prueba con número de dosis incorrecta  
PO3 - Prueba con pauta horaria incorrecta  
PO4 - Prueba con duración de tratamiento incorrecto  
PO5 - Prueba con datos correctos

## **R08 - Opción tratamientos En curso**

FO1 - Listar tratamientos por pauta horaria  
TO1 - Diseñar vista de tratamiento por horas  
TO2 - Programar método calcular tratamiento con pauta horaria  
TO3 - Programar servlet para listar tratamientos en curso con pauta horaria  
PO1 - Prueba listar tratamiento con pauta horaria

## **R09 - Opción Finalizados**

FO1 - Listar tratamientos finalizados  
TO1 - Programar listado de tratamientos finalizados  
TO2 - Programar servlet para listar tratamientos finalizados  
PO1 - Prueba listar tratamientos finalizados

FO2 - Eliminar tratamiento  
TO1 - Programar eliminación de tratamiento  
TO2 - Programar servlet de eliminación de tratamiento  
PO1 - Prueba eliminar tratamiento

## **R10 - Principal administrador**

FO1 - Ventana principal del administrador  
TO1 - Diseñar ventana principal y navegación del administrador

## R11 - Mantenimiento de Medicamentos

F01 - Listar medicamentos disponibles

    T01 - Programar servlet listar medicamentos

F02 - Agregar nuevo medicamento

    T01 - Programar opción agregar nuevo medicamento

    T02 - Programar servlet para agregar nuevo medicamento

F03 - Modificar medicamento existente

    T01 - Programar opción modificar medicamento

    T02 - Programar servlet para modificar y actualizar medicamento

F04 - Eliminar medicamento

    T01 - Programar opción eliminar medicamento

    T02 - Programar servlet para eliminar medicamento

    P01 - Prueba eliminar medicamento

F05 - Filtrar medicamentos

    T01 - Programar botón buscar medicamento

    T02 - Programar servlet para filtrar medicamentos

    P01 - Prueba filtrar medicamento

F06 - Validaciones de medicamento

    T01 - Programar validaciones de medicamentos

        P01 - Prueba con caracteres especiales no admitidos

        P02 - Prueba agregar medicamento con datos vacíos

        P03 - Prueba actualizar sin seleccionar medicamento previamente

        P04 - Prueba agregar medicamento correctamente

        P05 - Prueba modificar medicamento correctamente

## INTRODUCCIÓN

El objetivo del proyecto es poder gestionar de forma sencilla un tratamiento médico, para visualizar las pautas horarias de éste para que el usuario tenga en todo momento el control de sus tratamientos.

La idea de desarrollar esta aplicación surge de la necesidad de tener bajo control un tratamiento médico, introduciendo unos pocos datos en un formulario para que la aplicación calcule el horario de las tomas en función de la fecha y hora del equipo en el que se está trabajando. Siempre desde la sencillez porque lo que buscamos es la salud.

Para conseguir el objetivo principal se utiliza una metodología incremental que permite dividir el proyecto en pequeñas partes que se pueden desarrollar y probar independientemente. De esta manera, el proyecto se desarrolla de manera evolutiva.

La aplicación se desarrolla en el entorno de Eclipse IDE for Java EE Developers en su Versión: 2022-12, y como servidor web Tomcat versión 8.5, que interpreta el código Java de los JSP (Java Server Page).

La persistencia de los datos se realiza en una base de datos MySQL incluida en el paquete de libre distribución XAMPP, en esta base de datos se incluyen: Una tabla de usuarios, una tabla de medicamentos y la tabla de tratamientos médicos que van asociados a los usuarios.

Para la construcción del proyecto se utiliza la modularización, el patrón modelo vista controlador (MVC), para separar la lógica de presentación de la lógica de negocio y el modelo de datos.

- Los modelos o plantillas están desarrollados en clases Java.
- Las vistas están desarrolladas en JSP integrando HTML, CSS y Bootstrap.
- Los Servlets están ubicados en la parte del controlador, y son las clases Java que permiten recibir y responder peticiones HTTP Get o Post.
- Las clases Java que ejecutan validaciones u obtienen datos de la Base de Datos también están incluidos en el controlador.

Con esta estructura separada, se realizan modificaciones en las vistas y se desarrollan validaciones para realizar pruebas de control de errores, y se ha creado perfil de usuario administrador para realizar el mantenimiento de la tabla de medicamentos a la que tendrán acceso los usuarios.

## **JUSTIFICACIÓN DEL PROYECTO**

Actualmente, los pacientes a menudo tienen que hacer frente a procesos complejos y confusos al tratar una enfermedad. Una aplicación que permita gestionar un tratamiento médico de forma sencilla puede mejorar la experiencia del paciente, reducir el estrés y la confusión, y aumentar la probabilidad de que sigan el tratamiento correctamente.

Cuando un paciente debe iniciar un tratamiento médico, el facultativo le entrega una Hoja de Medicación que refleja el nombre del fármaco, fecha de inicio y fin, y pautas para ingerir la medicación, pero es el paciente el que debe gestionar personalmente sus alertas a riesgo de que no recuerde los horarios y/o qué medicación le corresponde tomar.

Para las personas mayores es más difícil recordar un tratamiento médico y todas sus pautas, debido en gran parte a la cantidad de medicamentos. Con una aplicación web, pueden realizar un seguimiento de los medicamentos que deben tomar, reduciendo el estrés al manejar múltiples tratamientos al mismo tiempo. Además, la aplicación también puede ser útil para los cuidadores de personas mayores o niños con múltiples afecciones, ya que les permite un registro de las dosis y horarios de administración de medicamentos.

Una aplicación web que gestione ese tratamiento médico puede ser fácilmente accesible desde cualquier lugar y en cualquier momento, lo que proporciona una mayor comodidad a los pacientes.

En resumen, CronoMed es un proyecto de desarrollo web que permite gestionar un tratamiento médico de forma sencilla, sí, mucho énfasis en “sencilla”..., que con introducir unos pocos parámetros puede tener un impacto positivo en la eficacia del tratamiento y en la tranquilidad del paciente.

## DESARROLLO

Se desarrolla una aplicación web para gestionar los tratamientos médicos prescritos por un facultativo, visualizando las pautas horarias y dosis de administración de una manera simple e intuitiva. El nombre escogido es **CronoMed**, que combina los términos "crono", que hace referencia al tiempo, y "med", como abreviatura de "medicación". Juntos forman un nombre descriptivo y fácil de recordar para una aplicación relacionada con la gestión del tiempo de los tratamientos médicos.

Para el desarrollo del proyecto se escoge la tecnología **JSP (Java Server Pages)**. Es una tecnología muy utilizada en el desarrollo de aplicaciones web empresariales y se utiliza junto con el servidor web Apache Tomcat o con otros servidores web compatibles con Java.

La mejor ventaja de escoger esta tecnología es que se puede integrar con el lenguaje de programación Java. JSP facilita la separación del código y el diseño: se puede utilizar HTML y CSS para definir la apariencia visual y agregar código Java para manejar la parte lógica, lo que da potencialidad a la programación. Estas ventajas permiten desarrollar aplicaciones web robustas, escalables y eficientes.

Java sigue siendo un lenguaje altamente demandado y frente a los competidores cercanos se posiciona como una de las tecnologías más punteras de las tecnologías web que se demandan a nivel empresarial.

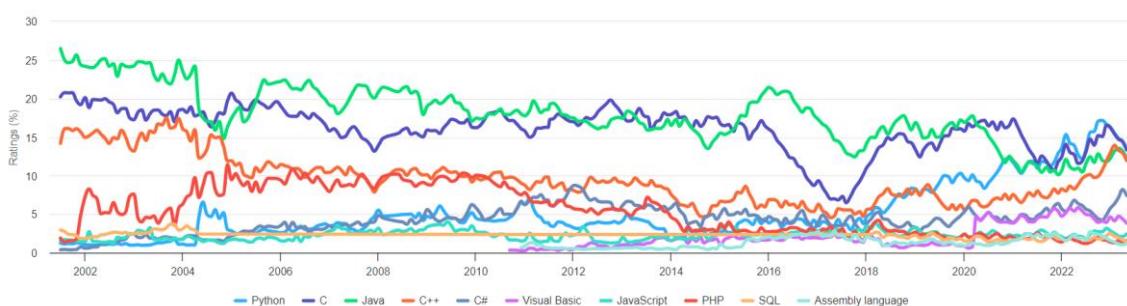


Ilustración 1: Índice de la comunidad de programación TIOBE ([www.tiobe.com](http://www.tiobe.com))

## Estructura de carpetas del proyecto

La aplicación web JavaEE utiliza servlets y páginas JSP con la siguiente estructura de ficheros y directorios:

- El directorio raíz de la aplicación contiene los ficheros JSP con las vistas de la aplicación.
- El directorio WEB-INF contiene la información Web relevante para la aplicación. Aquí se ha colocado la librería JSTL empaquetada que se ha utilizado reiterar las listas con bucles for each.
- El resto de los elementos de la aplicación (imágenes, scripts y estilos), se ha estructurado también por carpetas partiendo del directorio raíz.

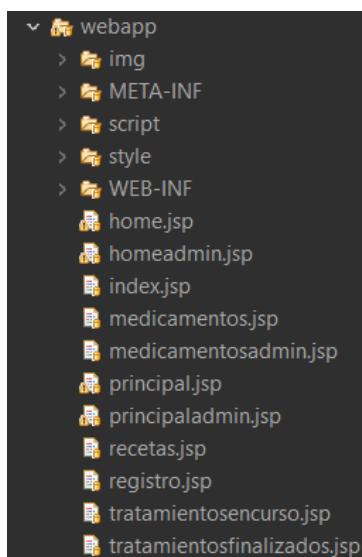


Ilustración 2: Estructura de carpetas del proyecto web

## Patrón de diseño

El proyecto está desarrollado bajo el modelo MVC (Modelo Vista Controlador), con el objeto de separar las responsabilidades de la aplicación en tres componentes principales, el Modelo, la Vista y el Controlador, separando así la lógica de presentación de la lógica de negocio de la aplicación.

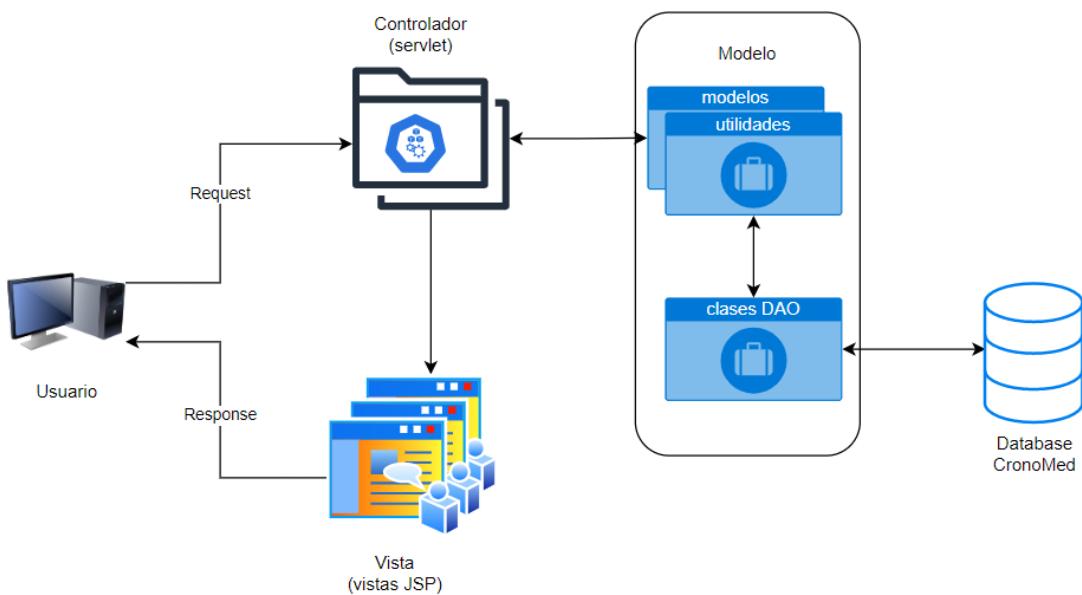


Ilustración 3: Modelo vista controlador de CronoMed

- **Modelo:** representa la lógica de negocio y los datos de la aplicación. Esta capa se refiere a los datos y acceso a datos. Este paquete contiene todas las clases Java de tipo modelo.
- **Vista:** representa la interfaz de usuario y la presentación de los datos al usuario. Esto incluye HTML, CSS y JavaScript en el caso la aplicación.
- **Controlador:** maneja las solicitudes del usuario y actúa como intermediario entre el modelo y la vista. Controla la lógica de la aplicación y toma decisiones basadas en la entrada del usuario. También incluye las clases que se encargan de interactuar con las tablas de la base de datos.

Este patrón proporciona una estructura organizada, un código más limpio, modular y fácil de mantener, lo cual favorece el mantenimiento, la depuración de errores y la escalabilidad de la aplicación.

En el paquete de Utilidades se ha desarrollado una clase Java que contiene todos aquellos métodos que realizan validaciones sobre la aplicación.

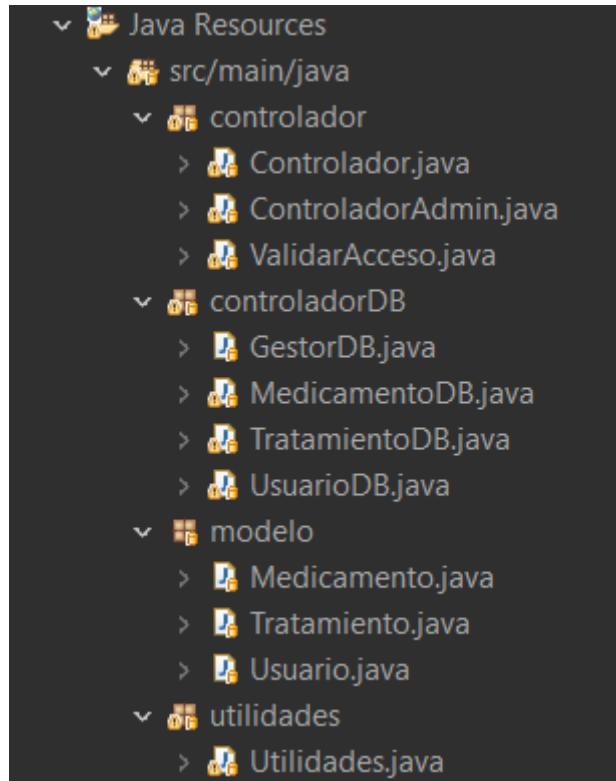


Ilustración 4: Modelo vista controlador del programa

Para crear los Servlets se crea una clase que herede de HttpServlet, que su vez implementa la interfase Servlet. Esta clase tiene el método doPost(HttpServletRequest, HttpServletResponse), los dos parámetros que recibe son de tipo interface.

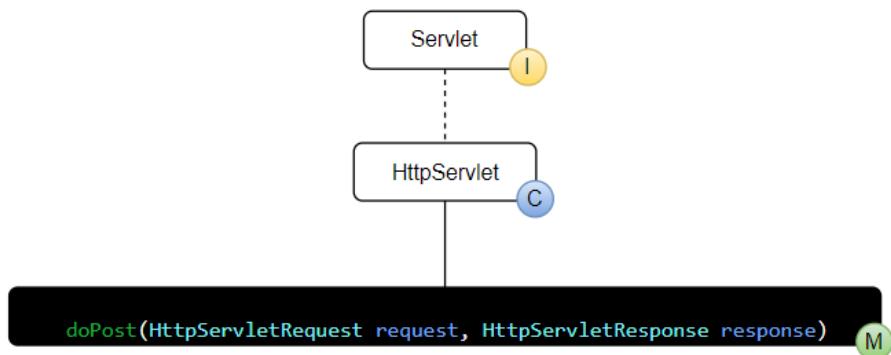


Ilustración 5: Interface Servlet Java

La utilización de los servlets para recibir y enviar respuestas a los JSP, se ejecuta de la siguiente forma:

1. En el “action” del formulario va identificado el nombre del servlet:

```
<form action="NombreServlet" method="post">
```

2. Todos los inputs tienen un nombre asignado en el campo “name”:

```
<input name="name" type="text" />
```

3. En el método doPost del Servlet correspondiente, se capturan los valores marcados en los inputs mediante el método getParameter de la interfaz HttpServletRequest:

```
TipoVariable nombreVariable = request.getParameter("name");
```

4. Se trabaja con los valores capturados, pasándolos como parámetros a los métodos de las clases Java y capturando el return en una variable:

```
resultado = claseJava.metodo(name);
```

5. El resultado se retorna al JSP utilizando el método setAttribute, que recibe dos parámetros: el primero corresponde a un nombre y el segundo al resultado:

```
request.setAttribute("resultado", resultado);
```

6. Para redireccionar a una página JSP, se utiliza el método getRequestDispatcher:

```
request.getRequestDispatcher(paginaWeb.jsp).forward(request, response);
```

7. Para el redireccionamiento entre Servlets, se realiza en el método doGet indicando el nombre del Servlet y las claves y valores a enviar:

```
request.getRequestDispatcher("NombreServlet?clave=valor").forward(request, response);
```

8. Para trabajar desde el JSP, con los datos que han sido enviados desde el Servlet después de haber ejecutado cualquier sentencia sobre la base de datos, se utiliza el nombre de la variable que se ha enviado a través del método “setAttribute” (punto 5) y el getter correspondiente del objeto recibido:

```
${resultado.getAttribute()}
```

## **Entorno de desarrollo**

El proyecto se ha desarrollado bajo el entorno de desarrollo Eclipse IDE for Java EE Developers versión 2022-12, que incluye soporte para tecnologías como Servlets y código JSP (Java Server Page). Se integra con uno de los servidores de aplicaciones web más popular Apache Tomcat.

Ofrece un potente depurador que permite seguir y solucionar problemas en el código de manera eficiente, tiene integración con herramientas de control de versiones como Git, y se pueden probar las aplicaciones web desde el IDE, lo que reduce el tiempo de desarrollo.

## **Entorno de servidor**

Se ha escogido el paquete de software Xampp porque proporciona un entorno de servidor fácil de instalar y de usar, puede ser instalado en cualquier sistema operativo, es de código abierto; esto lo convierte en la opción más popular para desarrollar aplicaciones web.

Su instalación incluye varios softwares que constituyen su acrónimo: el servidor web Apache, sistemas de gestión de bases de datos relacionales MySQL y MariaDB, así como los lenguajes de programación Perl y PHP especialmente diseñado para el desarrollo web.

Apache Tomcat es un servidor que simula que el ordenador es remoto y para poder probar las aplicaciones antes de subirlas a un ordenador real.

## **Gestión del proyecto**

Como herramienta de gestión del proyecto se ha escogido Trello. Es una herramienta visual y gratuita para organizar cualquier tipo de proyecto con la metodología TODO-DOING-DONE, es decir, identificando las tareas a realizar, acometerlas, y finalizarlas. En su versión de pago tiene incluso integración con el diagrama de Gantt.

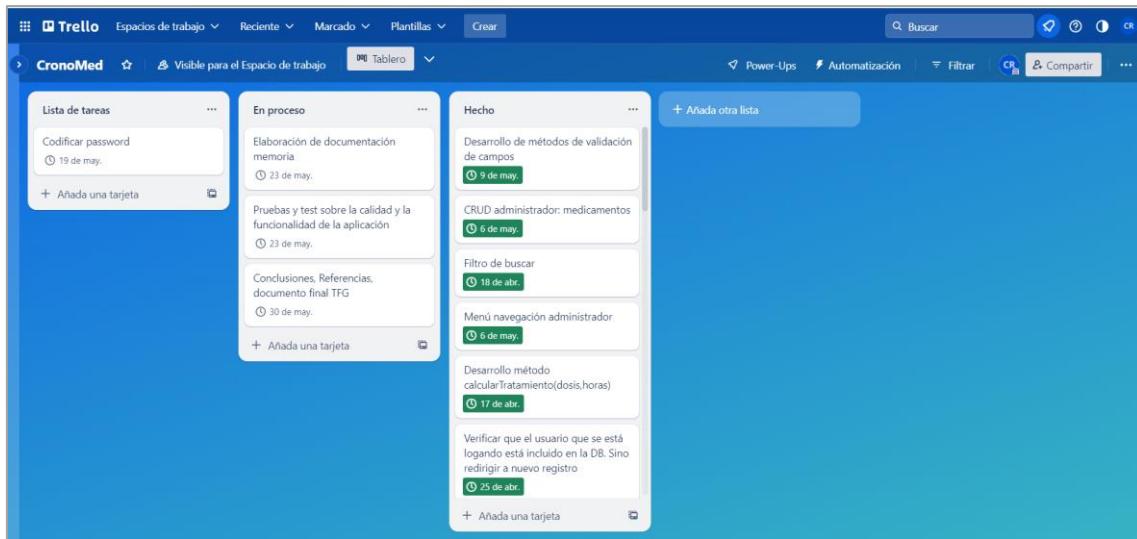


Ilustración 6: Herramienta Trello

## Control de versiones

Para respaldar el trabajo realizado y gestionar los cambios en el código, se han utilizado las herramientas Git y Sourcetree.

Git es un sistema de control de versiones que permite realizar el seguimiento de los cambios realizados en los archivos del proyecto. Sourcetree es una herramienta de interfaz gráfica que proporciona una forma más visual para interactuar con Git. Permite visualizar el historial de cambios y revertirlos en caso necesario, lo que es muy útil en un proyecto tan largo en el que se realizan muchos cambios.

[https://github.com/ClauRM/claudiarubio\\_tfg.git](https://github.com/ClauRM/claudiarubio_tfg.git)

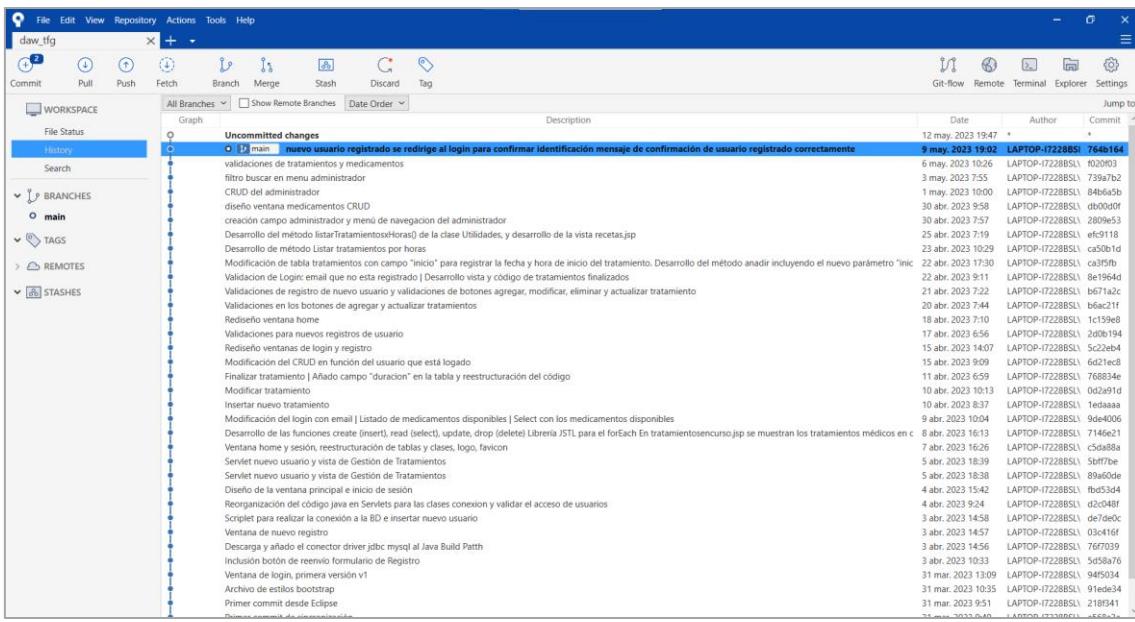


Ilustración 7: Herramienta visual Sourcetree

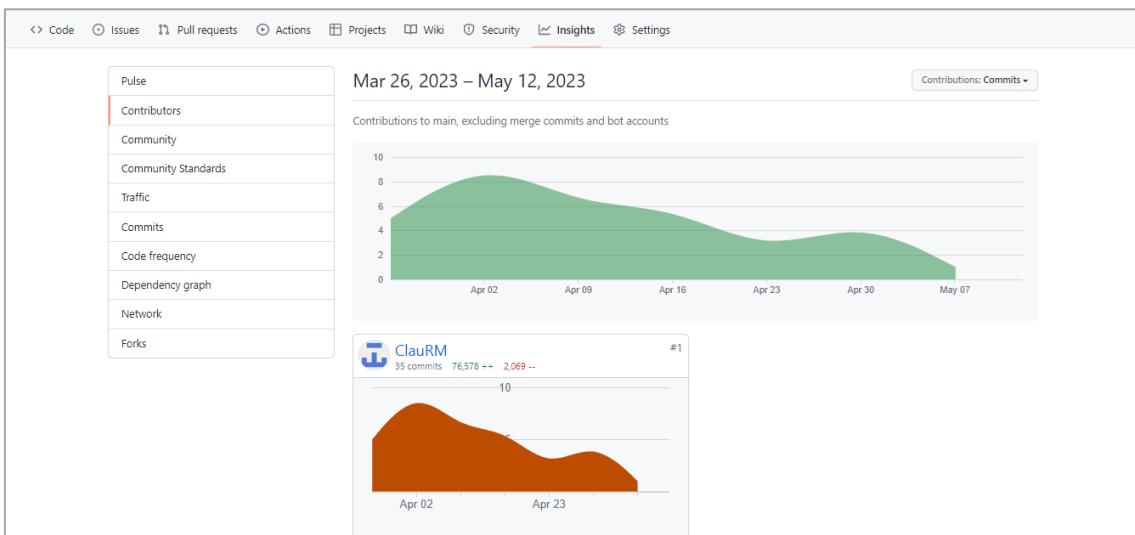


Ilustración 8: Contribuciones realizados en Git

## Dependencias

Para el proyecto se han utilizado librerías JDBC y JSTL.

La librería **JDBC mysql-connector-j-8.0.32.jar** (Java Database Connectivity) que permite realizar la conexión a una BD relacional. Proporciona los métodos para establecer la conexión con la base de datos, enviar consultas, recibir y procesar los resultados con operaciones como insertar, actualizar, eliminar y consultar datos (CRUD).

Este conector se agrega habitualmente al Java Build Path, pero también se agrega al proyecto web debido a que éste no se ejecuta desde la carpeta del espacio de trabajo, sino desde el servidor, por lo que las rutas de referencia cambian.

La librería **JSTL jstl-1.2.jar** (JavaServer Pages Standard Tag Library) permite iterar sobre listas de datos. Se ha utilizado en los JSP para mostrar los datos recibidos desde el controlador en la vista correspondiente.

También se ha utilizado las clases Java Date, DateFormaty SimpleDateFormat que forman parte de la **API de Java 1.8.0\_361** disponibles en el paquete java.util. Se han utilizado para trabajar con las fechas y horas que calculan la frecuencia horaria de administración del tratamiento médico.

Para la encriptación de la contraseña de usuario se ha utilizado la clase MessageDigest disponible en el paquete java.security de la API de Java.

## Despliegue de la aplicación

Para el despliegue de la aplicación CronoMed se pueden utilizar dos estrategias de despliegue:

- El archivo WAR desplegado en un servidor de aplicaciones web.
- Con servidor integrado en Eclipse IDE.

### Despliegue en servidor

Lo primero es empaquetar toda la aplicación en fichero WAR. Para crear este fichero se pincha con el botón derecho del ratón sobre el proyecto en Eclipse, exportar WAR file.

Tomcat incluye una aplicación llamada manager con una interfaz en HTML desde la cual se puede subir y desplegar la aplicación.

<http://localhost:8080/manager/html>

En la parte inferior de esta página se encuentra el formulario “desplegar”. Desde esta opción se puede subir y desplegar aplicación web en el servidor.

### Despliegue en servidor integrado eclipse IDE

Eclipse IDE permite crear un servidor de aplicaciones para los desarrollos en entorno local.

Desde la “view” server de Eclipse IDE es posible crear un servidor de aplicaciones para probar nuestros desarrollos. Para CronoMed se ha utilizado el servidor Tomcat 8.5 integrado en Eclipse.

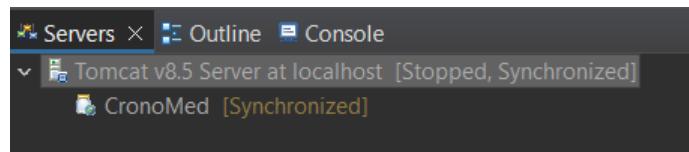


Ilustración 9: Servidor integrado en Eclipse

Para poder desplegar la aplicación utilizando este servidor, previamente se debe asignar un servidor al proyecto, y simplemente hay que pinchar con el botón derecho sobre el proyecto web y elegir la opción Run As - Run on Server.

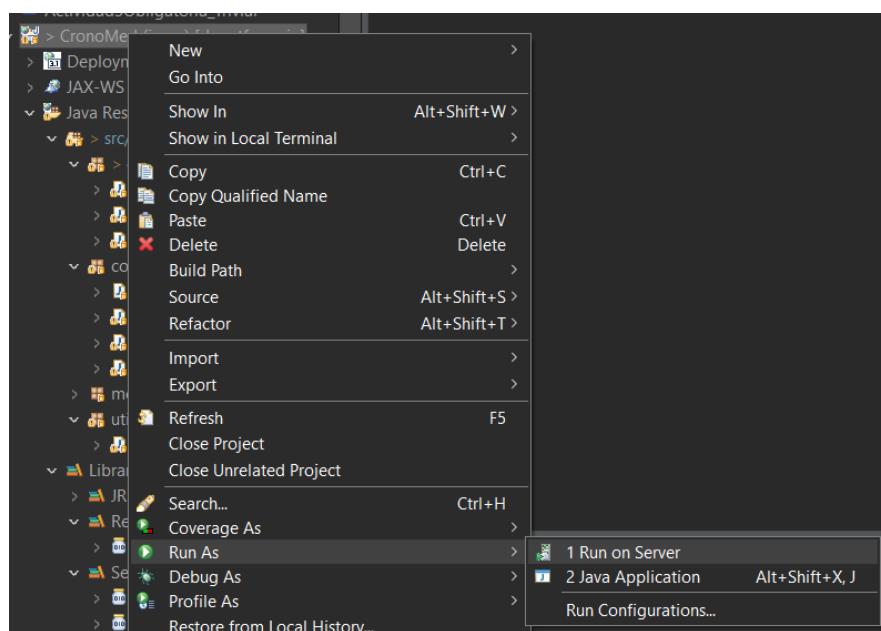


Ilustración 10: Despliegue en servidor integrado en Eclipse

## DISEÑO

En el apartado de análisis se estudian los requisitos y funcionalidades que debe satisfacer la aplicación web. Para ello se han utilizado técnicas estudiadas a lo largo del ciclo formativo, como son los casos de uso.

En el apartado del diseño se muestra todo el ámbito de diseño de CronoMed, el diagrama de clases, diagrama de entidad relación y diagrama de la base de datos, con el objeto de explicar el diseño técnico de la aplicación. Se incluye también un diagrama de flujo de navegación para explicar el aspecto visual y la funcionalidad de la aplicación.

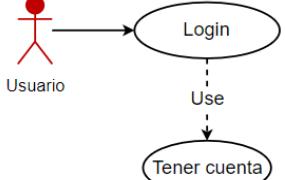
### Diagrama de casos de uso

En el siguiente diagrama de casos de uso se puede observar cómo interactúan los usuarios (o pacientes) con la aplicación CronoMed, y se explican las distintas funcionalidades cuando un usuario realiza alguna acción dentro de la aplicación, especificando las clases que interactúan para que la funcionalidad se lleve a cabo.

En la aplicación identificamos varias interacciones entre el usuario y el sistema durante el proceso gestión de un tratamiento médico e interacciones del Administrador para el mantenimiento de la tabla de medicamentos que está disponible al resto de usuarios.

Cada una de estas acciones del Usuario y del Administrador, incorpora diferentes funcionalidades que se describen a continuación, mediante tablas que representan gráficamente al actor, la acción realizada y clases, vistas y tablas que están afectadas.

En los dibujos, una línea simple indicará la relación entre el usuario y un caso de uso, y las relaciones tipo línea discontinua hacia afuera («incluir») indica que un caso de uso es necesario por otro para poder cumplir la funcionalidad.

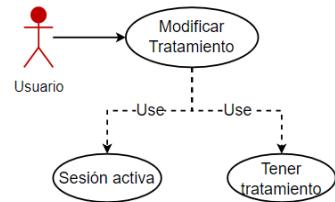
<b>CU01</b>	<b>Login</b>	
Actor principal	Usuario	
Interesados y objetivos	Usuario interesado en acceder a la aplicación para consultar o gestionar un tratamiento médico	 <pre> graph LR     Actor[Usuario] --&gt; Login((Login))     Login -- Use --&gt; TenerCuenta((Tener cuenta))   </pre>
Descripción	El usuario accederá a la aplicación en la que podría consultar el listado de medicamentos disponibles, gestionar un tratamiento, consultar los tratamientos que tiene en curso y/o eliminar un tratamiento	
Precondiciones	El usuario debe estar registrado en la aplicación	
Condición final	Acceso a la aplicación con los datos del usuario que ha iniciado sesión	
Datos de entrada	Formulario de Login	
Datos de salida	Home CronoMed	
Vistas JSP	index.jsp home.jsp	
Tablas BD	usuarios	
Clases Java	ValidarAcceso.java UsuarioDB.java Usuario.java GestorDB.java Controlador.java	

<b>CU02</b>	<b>Registro</b>	
Actor principal	Usuario	
Interesados y objetivos	Usuario interesado en registrarse en la aplicación para iniciar un tratamiento médico	 <pre> graph LR     Actor[Usuario] --&gt; Registro((Registro))   </pre>
Descripción	El usuario se registrará en la aplicación introduciendo una serie de parámetros para tener acceso a las funcionalidades de la aplicación	
Precondiciones	Ninguna	
Condición final	El Usuario registrado accederá a la aplicación haciendo Login para confirmar su identificación	
Datos de entrada	Formulario de Registro con nombre usuario, email y una password	
Datos de salida	Usuario registrado en la aplicación	
Vistas JSP	registro.jsp index.jsp	
Tablas BD	usuarios	
Clases Java	ValidarAcceso.java UsuarioDB.java Usuario.java GestorDB.java Controlador.java	

<b>CU03</b>	<b>Consultar Medicamentos</b>	
Actor principal	Usuario	
Interesados y objetivos	Usuario interesado en consultar el listado de medicamentos disponibles en la aplicación	<pre>     graph LR         Actor[Usuario] --&gt; UseCase{Consultar Medicamentos}         UseCase -.-&gt; Session[Sesión activa]         style UseCase fill:#e0e0ff         style Session fill:#e0e0ff         label "Use" between UseCase and Session     </pre>
Descripción	El Usuario conectado en la aplicación, podrá acceder al listado de medicamentos registrados en la aplicación para su consulta	
Precondiciones	Usuario con sesión iniciada ha clickado en la opción Medicamentos	
Condición final	El Usuario conectado visualiza un listado ordenado alfabéticamente con el nombre del medicamento y la composición	
Datos de entrada	Sesión activa	
Datos de salida	Medicamentos listados	
Vistas JSP	medicamentos.jsp	
Tablas BD	medicamentos	
Clases Java	Controlador.java MedicamentoDB.java GestorDB.java Medicamento.java	

<b>CU04</b>	<b>Agregar Tratamiento</b>	
Actor principal	Usuario	
Interesados y objetivos	Usuario interesado en añadir un nuevo tratamiento para calcular la frecuencia de administración	<pre>     graph LR         Actor[Usuario] --&gt; UseCase{Agregar Tratamiento}         UseCase -.-&gt; Session[Sesión activa]         style UseCase fill:#e0e0ff         style Session fill:#e0e0ff         label "Use" between UseCase and Session     </pre>
Descripción	El Usuario conectado en la aplicación, podrá acceder a la opción "Registro" para indicar nombre del paciente, medicamento, dosis, pauta horaria, duración y observaciones del tratamiento médico y crear un nuevo tratamiento tras pulsar el botón "Agregar"	
Precondiciones	Usuario registrado y con sesión iniciada	
Condición final	El Usuario conectado visualiza el nuevo tratamiento en un listado los tratamientos asociados a su perfil	
Datos de entrada	Sesión activa. Formulario de nuevo tratamiento	
Datos de salida	Tratamiento registrado en la aplicación	
Vistas JSP	tratamientosencurso.jsp	
Tablas BD	usuarios medicamentos tratamientos	
Clases Java	Controlador.java MedicamentoDB.java TratamientoDB.java GestorDB.java Utilidades.java Medicamento.java Tratamiento.java	

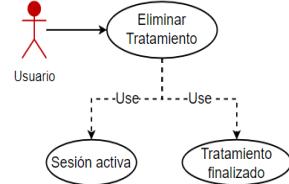
CU05	Modificar Tratamiento
Actor principal	Usuario
Interesados y objetivos	Usuario interesado en modificar un tratamiento existente relacionado a su perfil
Descripción	<p>El Usuario conectado en la aplicación, podrá modificar los datos asociados a un tratamiento médico, desde la opción "Registro". La secuencia de acciones es:</p> <ul style="list-style-type: none"> <li>- Posicionarse sobre un tratamiento médico del listado</li> <li>- Pulsar el botón "Modificar"</li> <li>- Modificar los datos en el formulario</li> <li>- Pulsar el botón "Actualizar"</li> </ul> <p>Tras ello se actualiza la información en el listado de tratamientos</p>
Precondiciones	Usuario registrado y con sesión iniciada, ha pulsado el botón "Modificar" de un tratamiento médico
Condición final	El Usuario conectado modifica los datos de un tratamiento agregado previamente
Datos de entrada	Sesión activa. Formulario actualizar tratamiento
Datos de salida	Tratamiento modificado en la aplicación
Vistas JSP	tratamientosencurso.jsp
Tablas BD	usuarios medicamentos tratamientos
Clases Java	Controlador.java MedicamentoDB.java TratamientoDB.java GestorDB.java Utilidades.java Medicamento.java Tratamiento.java



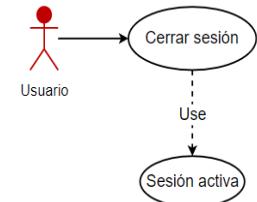
<b>CU06</b>	<b>Finalizar Tratamiento</b>	
Actor principal	Usuario	
Interesados y objetivos	Usuario interesado en finalizar un tratamiento existente relacionado a su perfil	<pre> graph LR     Actor[User] --&gt; UC[Finalizar Tratamiento]     UC -.-&gt; Sesion[Sesión activa]     UC -.-&gt; Tener[Tener tratamiento]     </pre>
Descripción	El Usuario conectado en la aplicación, podrá finalizar un tratamiento médico asociado a su perfil, desde la opción "Registro" y pulsando la opción "Finalizar", actualizando la información en los listados de tratamientos, en curso y finalizados	
Precondiciones	Usuario registrado y con sesión iniciada, ha pulsado el botón "Finalizar" un tratamiento médico	
Condición final	El Usuario conectado finaliza un tratamiento médico	
Datos de entrada	Sesión activa	
Datos de salida	Ninguno	
Vistas JSP	tratamientosencurso.jsp	
Tablas BD	usuarios medicamentos tratamientos	
Clases Java	Controlador.java TratamientoDB.java GestorDB.java Tratamiento.java	

<b>CU07</b>	<b>Consultar Tratamientos en Curso</b>	
Actor principal	Usuario	
Interesados y objetivos	Usuario interesado en consultar la frecuencia de administración de un tratamiento médico	<pre> graph LR     Actor[User] --&gt; UC[Consultar Tratamiento]     UC -.-&gt; Sesion[Sesión activa]     UC -.-&gt; Tener[Tener tratamiento]     </pre>
Descripción	El Usuario conectado en la aplicación, podrá consultar la frecuencia horaria de los tratamientos médicos que tiene en curso	
Precondiciones	Usuario registrado y con sesión iniciada, ha pulsado la opción "En Curso" del menú de navegación o card de la ventana principal	
Condición final	El Usuario conectado visualiza un listado con los tramos horarios en los que deberá administrar una medicación a un paciente y en qué dosis	
Datos de entrada	Sesión activa	
Datos de salida	Tratamientos en curso listado	
Vistas JSP	recetas.jsp	
Tablas BD	usuarios medicamentos tratamientos	
Clases Java	Controlador.java TratamientoDB.java GestorDB.java Tratamiento.java Utilidades.java	

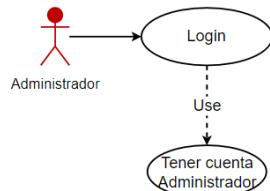
<b>CU08</b>	<b>Eliminar tratamiento</b>
Actor principal	Usuario
Interesados y objetivos	Usuario interesado en eliminar un tratamiento médico de su perfil
Descripción	El Usuario conectado en la aplicación, podrá eliminar los tratamientos médicos finalizados desde la opción "Finalizados"
Precondiciones	Usuario registrado y con sesión iniciada, ha pulsado el botón "Eliminar" sobre un tratamiento del listado de tratamientos "Finalizados"
Condición final	El Usuario conectado elimina un tratamiento del historial de tratamientos asociado a su perfil
Datos de entrada	Sesión activa
Datos de salida	Tratamiento eliminado
Vistas JSP	tratamientosfinalizados.jsp
Tablas BD	usuarios medicamentos tratamientos
Clases Java	Controlador.java TratamientoDB.java GestorDB.java Tratamiento.java



<b>CU09</b>	<b>Cerrar sesión</b>
Actor principal	Usuario
Interesados y objetivos	Usuario interesado en cerrar sesión
Descripción	El Usuario conectado en la aplicación quiere terminar su sesión y salir de la aplicación
Precondiciones	Usuario registrado y con sesión iniciada, ha pulsado el botón "Salir" de su perfil
Condición final	Cierre de sesión y vuelta a la ventana de login inicial
Datos de entrada	Sesión activa
Datos de salida	Ninguna
Vistas JSP	home.jsp
Tablas BD	usuarios
Clases Java	Controlador.java GestorDB.java



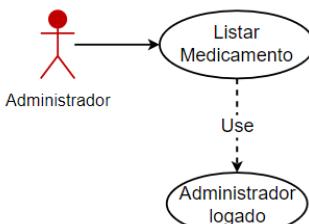
<b>CU10</b>	<b>Login</b>
Actor principal	Administrador
Interesados y objetivos	Usuario Administrador interesado en acceder a la aplicación para mantener la tabla medicamentos
Descripción	El Administrador accederá a la aplicación en la que podrá realizar el mantenimiento de la tabla medicamentos
Precondiciones	Ser usuario Administrador
Datos de entrada	Formulario Login
Datos de salida	Home Administrador
Condición final	Acceso a la aplicación con los datos del Administrador que ha iniciado sesión
Vistas JSP	index.jsp homeadmin.jsp
Tablas BD	usuarios
Clases Java	ValidarAcceso.java UsuarioDB.java Usuario.java GestorDB.java ControladorAdmin.java



```

graph TD
    Actor[Administrador] --> Login((Login))
    Login -- Use --> AdminLogado[Tener cuenta Administrador]
  
```

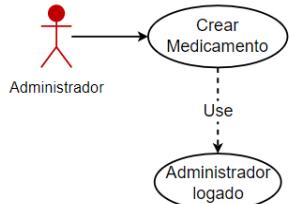
<b>CU11</b>	<b>Listar medicamentos</b>
Actor principal	Administrador
Interesados y objetivos	Usuario Administrador interesado en consultar contenido de la tabla medicamentos
Descripción	El Administrador conectado en la aplicación, podrá consultar el listado completo de medicamentos registrados en la aplicación
Precondiciones	Administrador con sesión iniciada
Condición final	El Administrador conectado visualiza un listado ordenado alfabéticamente con el nombre del medicamento y la composición
Datos de entrada	Administrador con sesión activa
Datos de salida	Listado de medicamentos
Vistas JSP	medicamentosadmin.jsp
Tablas BD	medicamentos
Clases Java	ControladorAdmin.java MedicamentosDB.java Medicamentos.java GestorDB.java



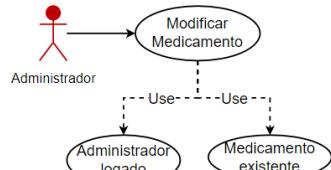
```

graph TD
    Actor[Administrador] --> ListarMedicamento((Listar Medicamento))
    ListarMedicamento -- Use --> AdminLogado[Administrador logado]
  
```

<b>CU12</b>	<b>Crear medicamento</b>
Actor principal	Administrador
Interesados y objetivos	Usuario Administrador interesado en agregar un nuevo medicamento al listado de medicamentos disponibles
Descripción	El Administrador conectado en la aplicación, podrá agregar un nuevo medicamento indicando nombre y composición para añadirlo al listado de medicamentos disponibles en la aplicación
Precondiciones	Administrador conectado pulsa el botón "Agregar"
Condición final	El Administrador conectado y los Usuarios visualizan el nuevo medicamento en el listado de medicamentos
Datos de entrada	Administrador con sesión activa. Formulario de registro/modificación medicamento
Datos de salida	Medicamento creado
Vistas JSP	medicamentosadmin.jsp
Tablas BD	medicamentos
Clases Java	ControladorAdmin.java MedicamentosDB.java Medicamentos.java GestorDB.java Utilidades.java



<b>CU13</b>	<b>Modificar medicamento</b>
Actor principal	Administrador
Interesados y objetivos	Usuario Administrador interesado en modificar el nombre y/o composición de un medicamento
Descripción	El Administrador conectado en la aplicación, podrá modificar los datos de un medicamento existente con el botón "Actualizar" para renderizar la información de la tabla medicamentos
Precondiciones	Administrador conectado pulsa el botón "Modificar" en medicamento
Condición final	El Administrador conectado y los Usuarios visualizan la modificación del medicamento en el listado de medicamentos
Datos de entrada	Administrador con sesión activa. Formulario de registro/modificación medicamento
Datos de salida	Tratamiento modificado
Vistas JSP	medicamentosadmin.jsp
Tablas BD	medicamentos
Clases Java	ControladorAdmin.java MedicamentosDB.java Medicamentos.java GestorDB.java Utilidades.java



<b>CU14</b>	<b>Eliminar medicamento</b>	
Actor principal	Administrador	
Interesados y objetivos	Usuario Administrador interesado en eliminar un medicamento del listado de disponible	<pre> sequenceDiagram     participant Actor as Administrador     participant Object as Eliminar Medicamento     participant ActorLogado as Administrador logado     participant Medicamento as Medicamento existente     Actor-&gt;&gt;Object:      activate Object     Object--&gt;&gt;ActorLogado: Use     ActorLogado--&gt;&gt;Medicamento: Use   </pre>
Descripción	El Administrador conectado en la aplicación, podrá eliminar un medicamento existente	
Precondiciones	Administrador conectado pulsa el botón "Eliminar" sobre un elemento de la lista de medicamentos	
Condición final	El Administrador elimina un medicamento de la tabla principal	
Datos de entrada	Administrador con sesión activa	
Datos de salida	Medicamento eliminado	
Vistas JSP	medicamentosadmin.jsp	
Tablas BD	medicamentos	
Clases Java	ControladorAdmin.java MedicamentosDB.java Medicamentos.java GestorDB.java	

<b>CU15</b>	<b>Buscar medicamentos</b>	
Actor principal	Administrador	
Interesados y objetivos	Usuario Administrador interesado en filtrar medicamento por su nombre y/o composición	<pre> sequenceDiagram     participant Actor as Administrador     participant Object as Filtrar Medicamento     participant ActorLogado as Administrador logado     Actor-&gt;&gt;Object:      activate Object     Object--&gt;&gt;ActorLogado: Use   </pre>
Descripción	El Administrador conectado en la aplicación, podrá buscar un medicamento existente tecleando parte de su nombre	
Precondiciones	Administrador conectado escribe texto y pulsa el botón "Filtrar"	
Condición final	Se renderiza el contenido del listado de medicamentos filtrando por el texto indicado	
Datos de entrada	Administrador con sesión activa	
Datos de salida	Ninguno	
Vistas JSP	medicamentosadmin.jsp	
Tablas BD	medicamentos	
Clases Java	ControladorAdmin.java MedicamentosDB.java Medicamentos.java GestorDB.java	

## Diagrama de clases

En el siguiente diagrama de clases se muestra la estructura de clases que intervienen en la aplicación, y las relaciones entre cada una de ellas.

De derecha a izquierda del diagrama, se han ubicado las clases modelo (usuario, medicamento y tratamiento que se corresponden con las tablas de la base de datos), los Servlets (comunican la página web en jsp con las clases java), las clases DAO (data access object, interfaz entre el código y la base de datos) y finalmente la clase Utilidades encargada de realizar las validaciones de todos los datos.

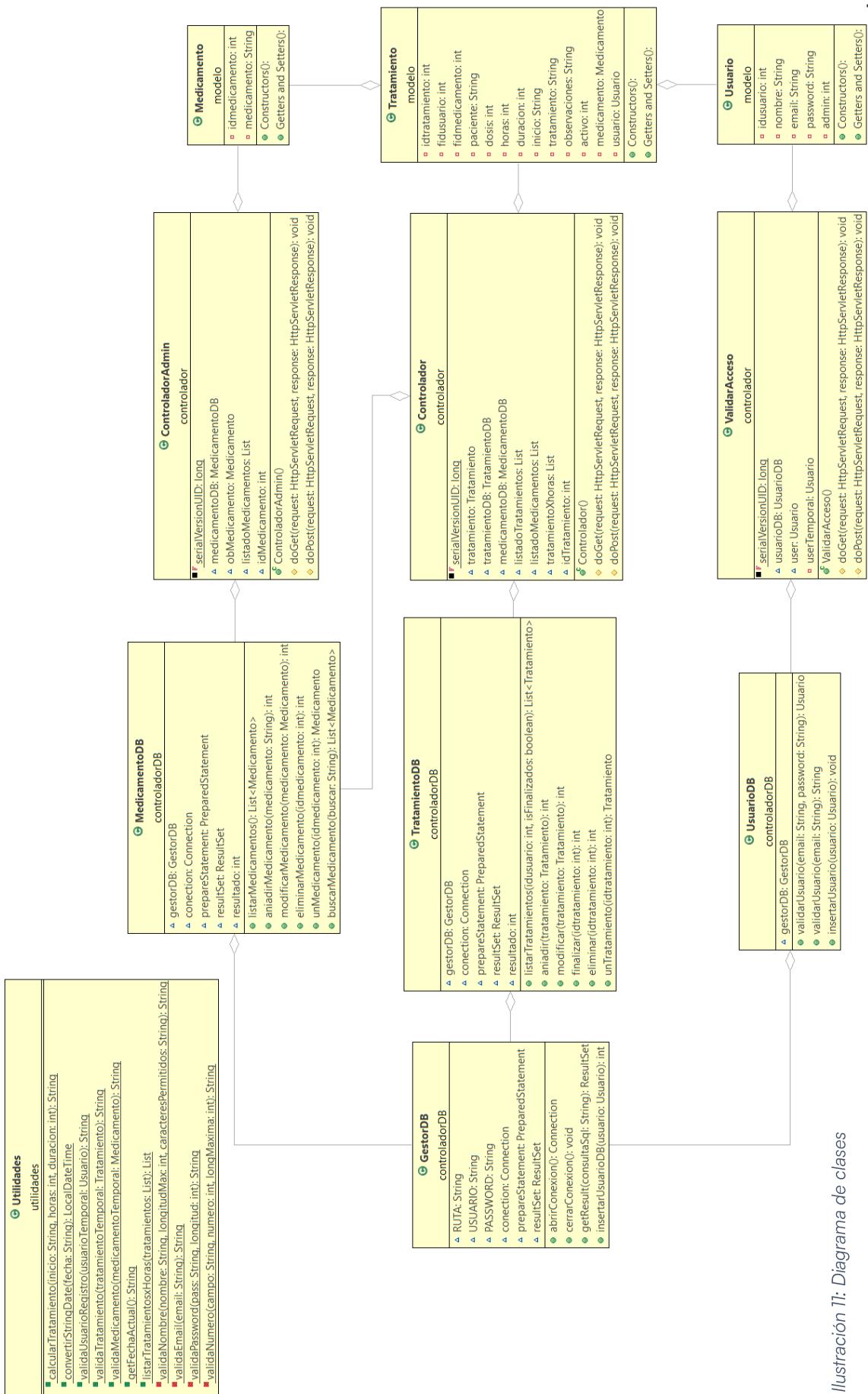


Ilustración 77: Diagrama de clases

## Diagrama entidad relación

El siguiente diagrama muestra la estructura de la base de datos: las tablas, los campos y la relación entre ellas. La base de datos utilizada se compone de las tablas usuario, tratamientos y medicamentos, y representa la relación entre los diferentes componentes que forman parte del sistema de gestión de tratamientos médicos.

La tabla usuario se utiliza para almacenar la información de los usuarios, incluyendo un id autoincremental, nombre, correo electrónico, contraseña y si es administrador o no. La tabla medicamentos, por su parte, almacena la información sobre los medicamentos disponibles, incluyendo un id autoincremental y nombre. La tabla tratamientos se utiliza para almacenar información específica sobre los tratamientos que están siendo aplicados a los pacientes, incluyendo detalles como un id autoincremental, nombre del paciente, la dosis, las horas, la duración, la fecha y hora de inicio del tratamiento, las observaciones y si está activo o no.

La relación entre estas tablas se da a través de las claves foráneas fidusuario y fidmedicamento, que conectan la tabla tratamientos con las tablas usuario y medicamentos respectivamente. Esto permite establecer la relación entre los pacientes y los tratamientos que están recibiendo, así como la relación entre los medicamentos y los tratamientos en los que se están utilizando.

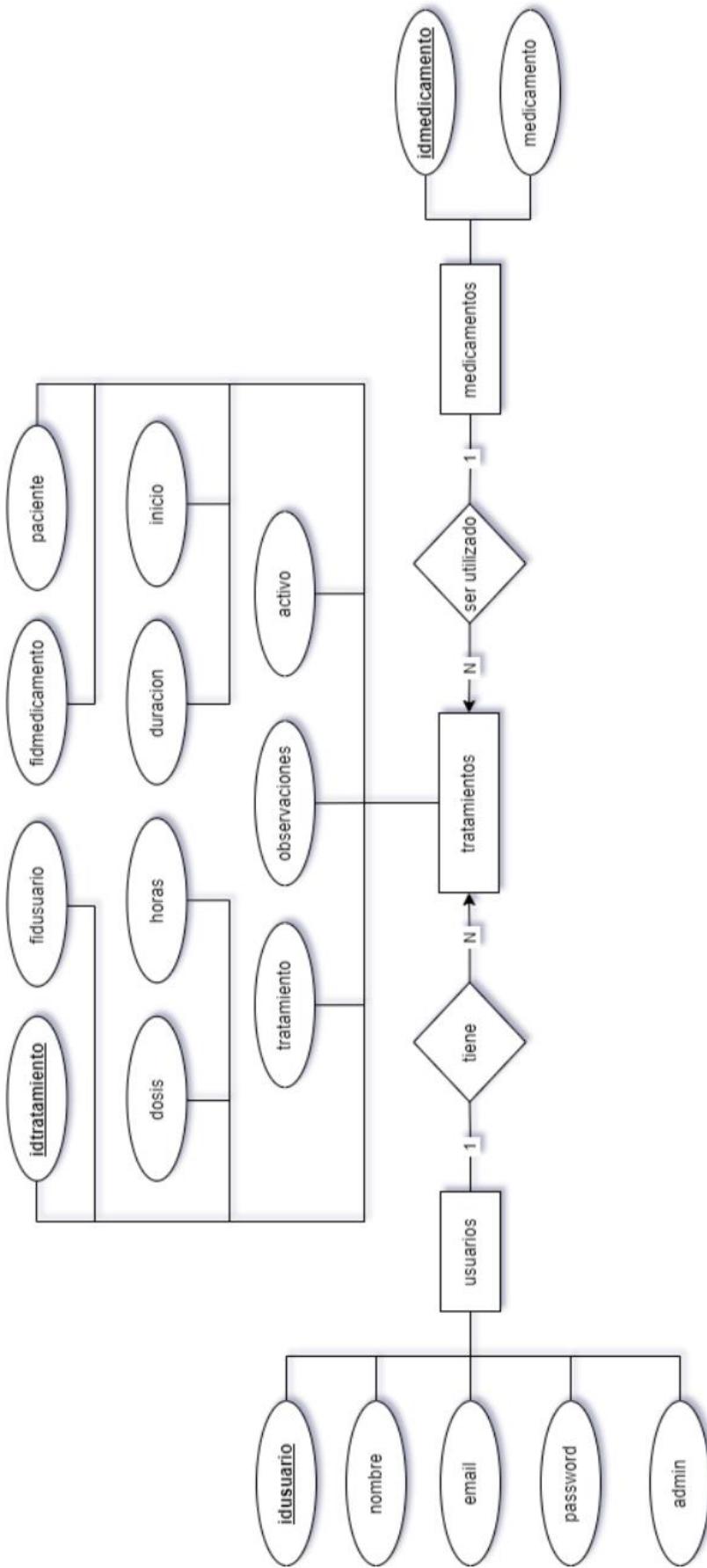


Ilustración 12: Diagrama entidad relación

## **Diagrama de base de datos**

Se utiliza una base de datos relacional en MySQL, basado en la organización de los datos en “tablas” relacionadas entre sí. Las tablas están formadas por filas de datos y por columnas que corresponden al tipo de información que almacena.

La información almacenada en cada celda son los registros. Cada tabla tiene una clave primaria, que es un identificador único de la relación entre tablas.

La tabla con más información y sobre la que actúa la mayor parte de la lógica de la aplicación es la tabla de tratamientos. Esta tabla contiene como primary key el código del idtratamiento, que es único e irrepetible, y como claves foráneas las correspondientes al usuario (fidusuario) y al medicamento (fidmedicamento).

A continuación, se muestra el diagrama de la base de datos, las tablas que interactúan con la aplicación y las relaciones entre ellas.

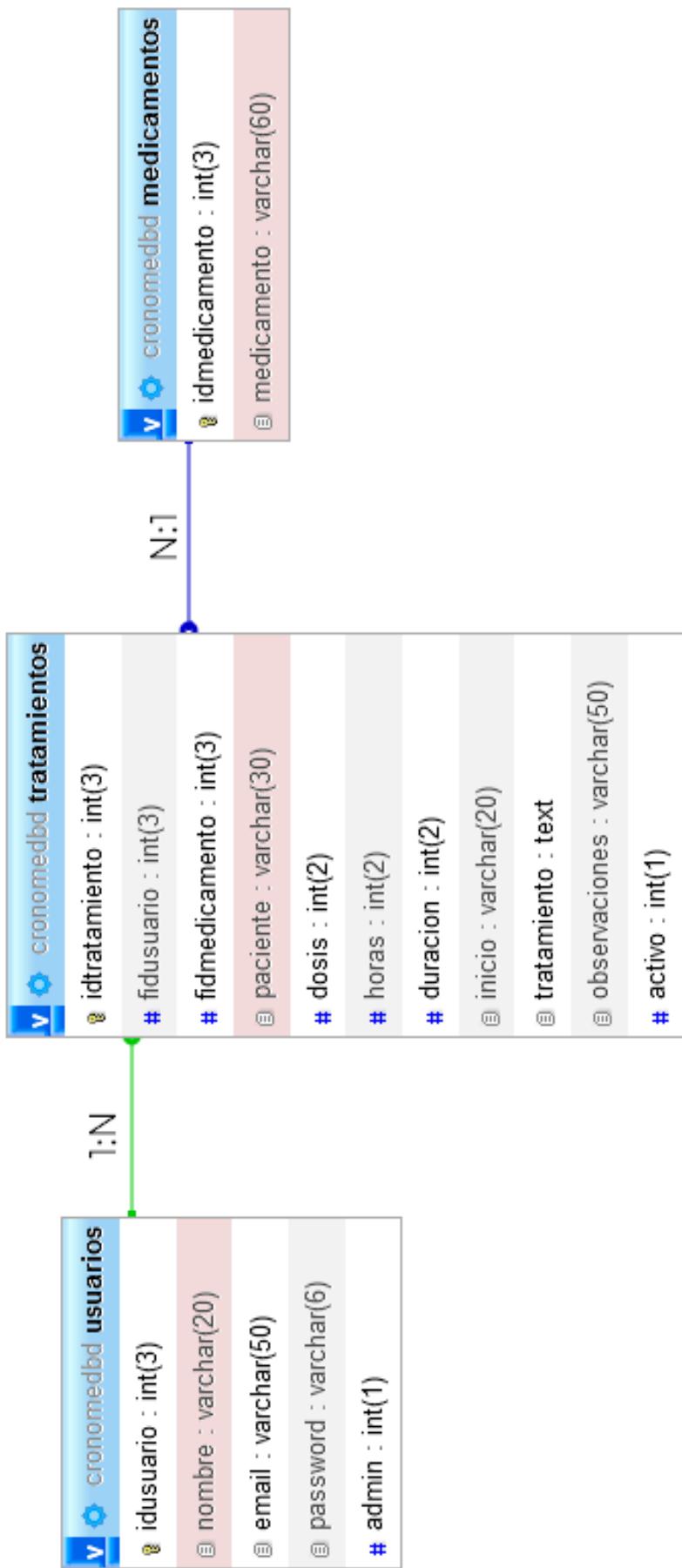


Ilustración 13: Diagrama de Base de Datos

## Diagrama de flujo de navegación

En el siguiente esquema se representa visualmente las páginas que conforman la aplicación y las funcionalidades disponibles en cada vista. La aplicación se ha desarrollado de forma que sea fácilmente comprensible para que los usuarios puedan acceder a cada funcionalidad de manera intuitiva.

Cuando el usuario registrado hace login accede a la ventana principal de CronoMed (home) en el que encontrará una descripción de la web y sus funcionalidades: Consulta de Medicamentos, Registro de Tratamientos, Tratamientos en Curso y Tratamientos Finalizados. Al cerrar la sesión, se le redirige a la ventana del login.

Si se trata de un Nuevo usuario, esté se registra y se le redirige a la ventana del login confirmado su registro y para que pueda acceder a la aplicación haciendo llogin.

El perfil de Administrador accede a la aplicación haciendo login y dispondrá de las funcionalidades necesarias para el mantenimiento de la tabla Medicamentos.

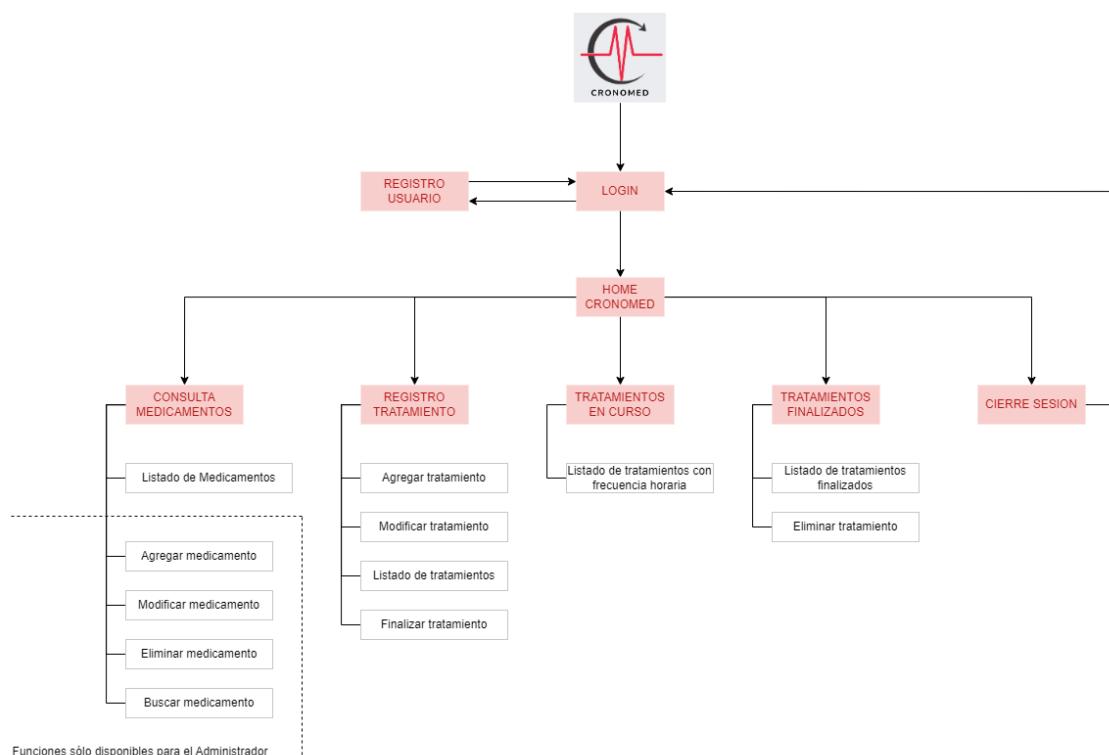


Ilustración 14: Esquema de navegación

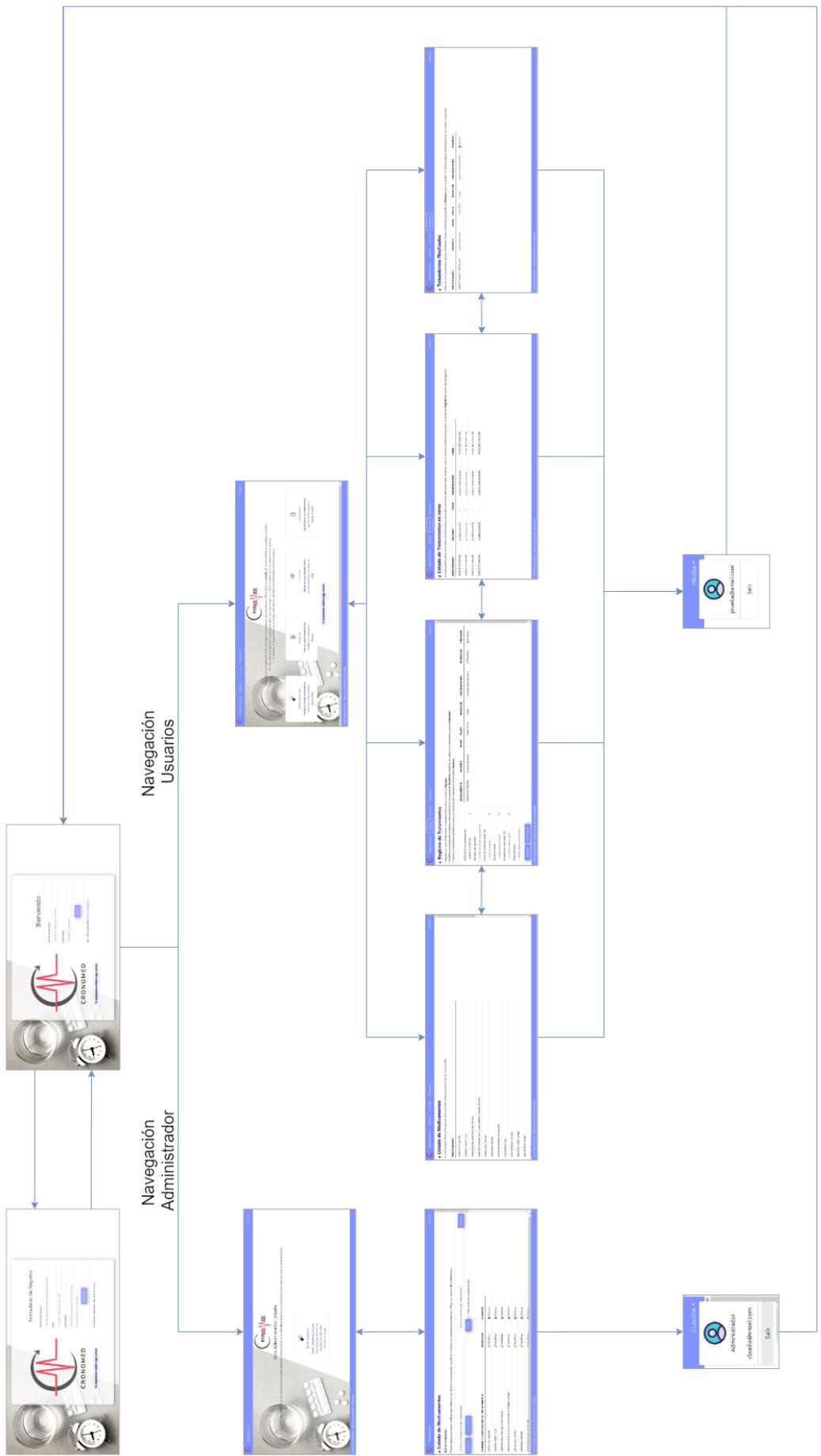


Ilustración 15: Diagrama de flujo de navegación

## **METODOLOGÍA**

Para el desarrollo se ha utilizado una metodología que se centra en la entrega incremental con enfoque en cascada. Se ha partido de una idea inicial del proyecto y se ha ido modificando, teniendo en cuenta la complejidad de las tareas y los plazos de entrega.

Se han tenido en cuenta ciclos de desarrollo iterativos e incrementales, generalmente llamadas "sprints". Cada sprint tiene una duración fija y al final de cada uno se muestra al tutor una versión funcional del proyecto.

Con esta metodología se tiene flexibilidad y adaptación al cambio. Se ha centrado en ofrecer una aplicación funcional desde la primera entrega, en lugar de esperar hasta el final del proyecto para realizar la entrega completa. Para ello se han separado las tareas y se han priorizado las características más importantes lo que permite el ajuste en función de sus necesidades y expectativas.

Se destaca la calidad de la aplicación y las pruebas continuas que se realizan sobre cada iteración, de manera regular, lo cual ayuda a detectar y corregir problemas a tiempo, para garantizar que la aplicación cumple con los requisitos y que funcione correctamente.

### **Diagrama de Gantt**

En el siguiente cuadro se detallan las tareas ejecutadas para desarrollar la aplicación CronoMed, los tiempos utilizados indicando la fecha de inicio y finalización de cada una, así como las horas que han sido requeridas para completarlas. El proyecto se desarrolla en 8 semanas completas, con un total de 60,6 horas empleadas en el desarrollo y 14,0 horas en la elaboración de la memoria.

Cabe resaltar que el tiempo previo empleado en la formación de la tecnología programación web JSP (Java Server Pages) para la creación de la aplicación dinámica en Java, no se ha incluido en el cuadro. Esta tarea de formación se lleva a cabo en 3 semanas, con un total de 20 horas empleadas entre teoría y prácticas.

REQ	FUN	NOMBRE DE LA TAREA	FECHA INICIO	FECHA FIN	DIAS	HORAS
R01	F01	Diseñar tablas, estructura y diagrama entidad relación	31/03/2023	31/03/2023	1	0,8
R01	F02	Desarrollo modelo vista controlador y organización de servlets	31/03/2023	03/04/2023	4	2,3
R01	F03	Programar conexión a la bd	03/04/2023	03/04/2023	1	1,8
R05	F01	Diseñar el home de la ventana principal del usuario	31/03/2023	05/04/2023	6	2,3
R02	F01	Diseñar ventana de inicio de sesión	31/03/2023	04/04/2023	5	1,5
R03	F01	Diseñar ventana de registro de usuario	03/04/2023	03/04/2023	1	1,2
R03	F02	Programar servlet de registro de nuevo usuario	03/04/2023	04/04/2023	2	1,7
R03	F02	Programar inserción nuevo usuario en la tabla usuarios	04/04/2023	05/04/2023	2	1,6
R02	F02	Programar verificación de usuario y contraseña	03/04/2023	03/04/2023	1	1,5
R02	F02	Programar servlet de login	03/04/2023	03/04/2023	1	1,2
R04	F01	Programar cierre de sesión de usuario	04/04/2023	04/04/2023	1	0,5
R05	F02	Programar botones para navegación entre ventanas	05/04/2023	05/04/2023	1	1,0
R07	F02	Programar método listar tratamiento asociado al usuario con sesión	05/04/2023	07/04/2023	3	1,7
R07	F02	Programar servlet para listar tratamientos por usuario	07/04/2023	12/04/2023	6	1,6
R07	F04	Programar modificación de tratamiento existente	05/04/2023	07/04/2023	3	1,7
R07	F04	Programar servlet para modificar y actualizar tratamiento existente	08/04/2023	10/04/2023	3	1,8
R07	F03	Programar inserción de nuevo tratamiento	08/04/2023	09/04/2023	2	1,6
R07	F03	Programar servlet para insertar nuevo tratamiento	09/04/2023	10/04/2023	2	1,6
R05	F01	Programar login de usuario y asociarlo a la sesión	09/04/2023	09/04/2023	1	1,7
R06	F01	Programar método para listar los medicamentos	09/04/2023	09/04/2023	1	0,8
R06	F01	Programar servlet para listar los medicamentos	09/04/2023	09/04/2023	1	0,7
R07	F01	Diseñar ventana de nuevo tratamiento	09/04/2023	09/04/2023	1	1,2
R07	F05	Programar finalización de tratamiento	11/04/2023	11/04/2023	1	0,8
R07	F05	Programar servlet para finalización de tratamiento	11/04/2023	11/04/2023	1	0,7
R03	F03	Validaciones de registro de usuario	17/04/2023	21/04/2023	5	1,4
R07	F06	Programar validaciones de tratamiento	20/04/2023	21/04/2023	2	2,1
R09	F01	Programar listado de tratamientos finalizados	21/04/2023	21/04/2023	1	0,8
R09	F01	Programar servlet para listar tratamientos finalizados	21/04/2023	21/04/2023	1	0,6
R02	F03	Validaciones de formulario inicio de sesión	22/04/2023	22/04/2023	1	0,7
R09	F02	Programar eliminación de tratamiento	22/04/2023	22/04/2023	1	0,8
R09	F02	Programar servlet de eliminación de tratamiento	22/04/2023	22/04/2023	1	0,7
R08	F01	Programar método calcular tratamiento con pauta horaria	22/04/2023	23/04/2023	2	3,8
R08	F01	Programar servlet para listar tratamientos en curso con pauta horaria	23/04/2023	25/04/2023	3	3,2
R08	F01	Diseñar vista de tratamiento por horas	23/04/2023	25/04/2023	3	1,2
R10	F01	Diseñar ventana principal y navegación del administrador	29/04/2023	29/04/2023	1	2,5
R11	F01	Programar servlet listar medicamentos	30/04/2023	30/04/2023	1	0,7
R11	F02	Programar opción agregar nuevo medicamento	30/04/2023	30/04/2023	1	1,0
R11	F02	Programar servlet para agregar nuevo medicamento	30/04/2023	30/04/2023	1	0,7
R11	F03	Programar opción modificar medicamento	01/05/2023	01/05/2023	1	0,9
R11	F03	Programar servlet para modificar y actualizar medicamento	01/05/2023	01/05/2023	1	0,6
R11	F04	Programar opción eliminar medicamento	01/05/2023	01/05/2023	1	0,8
R11	F04	Programar servlet para eliminar medicamento	01/05/2023	01/05/2023	1	0,7
R11	F05	Programar botón buscar medicamento	01/05/2023	03/05/2023	3	0,7
R11	F05	Programar servlet para filtrar medicamentos	01/05/2023	03/05/2023	3	1,3
R11	F06	Programar validaciones de medicamentos	05/05/2023	06/05/2023	2	2,1
R12	F01	Elaboración de la memoria	06/05/2023	19/05/2023	14	14,0
						TOTAL EN HORAS 74,6

En el siguiente gráfico de horas de desarrollo se observa visualmente que la tarea que más ha requerido tiempo es “R08 F01 Programar método calcular tratamiento con pauta horaria”, que coincide con en el objetivo principal del proyecto CronoMed.

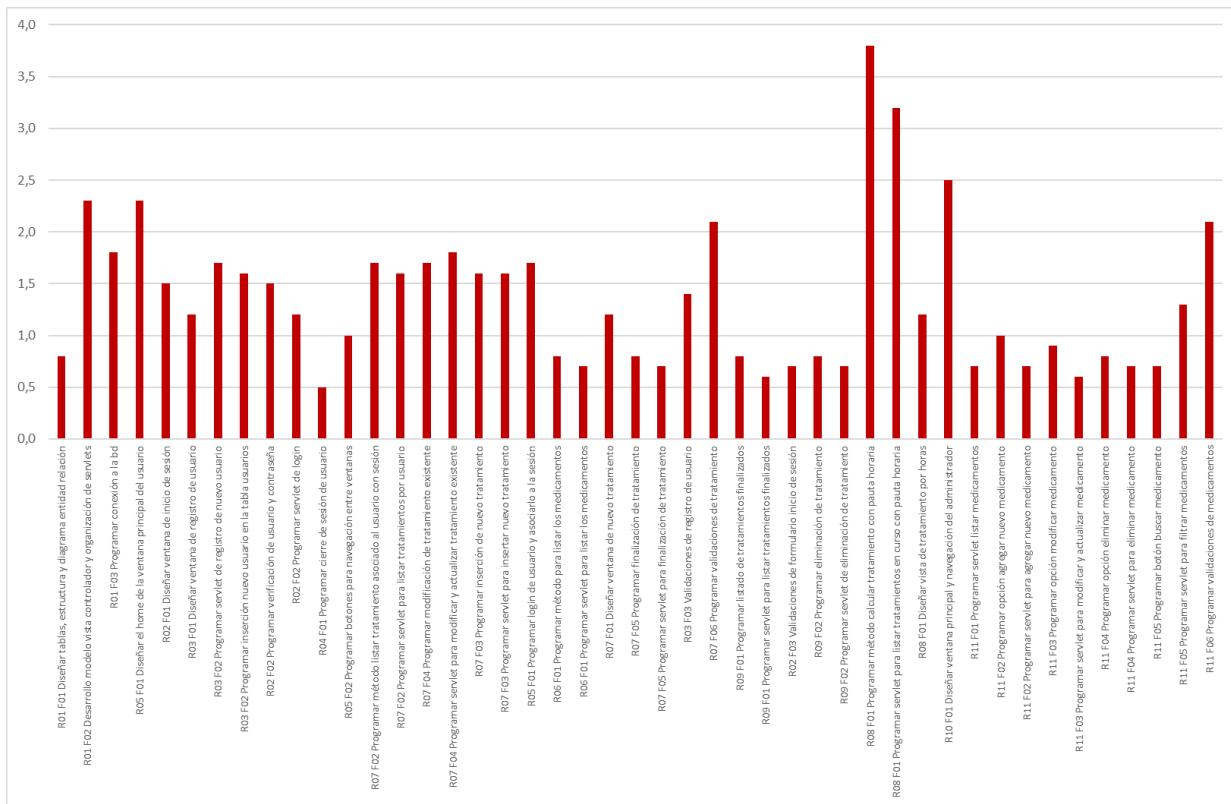


Ilustración 16: Gráfico de horas empleadas por tarea

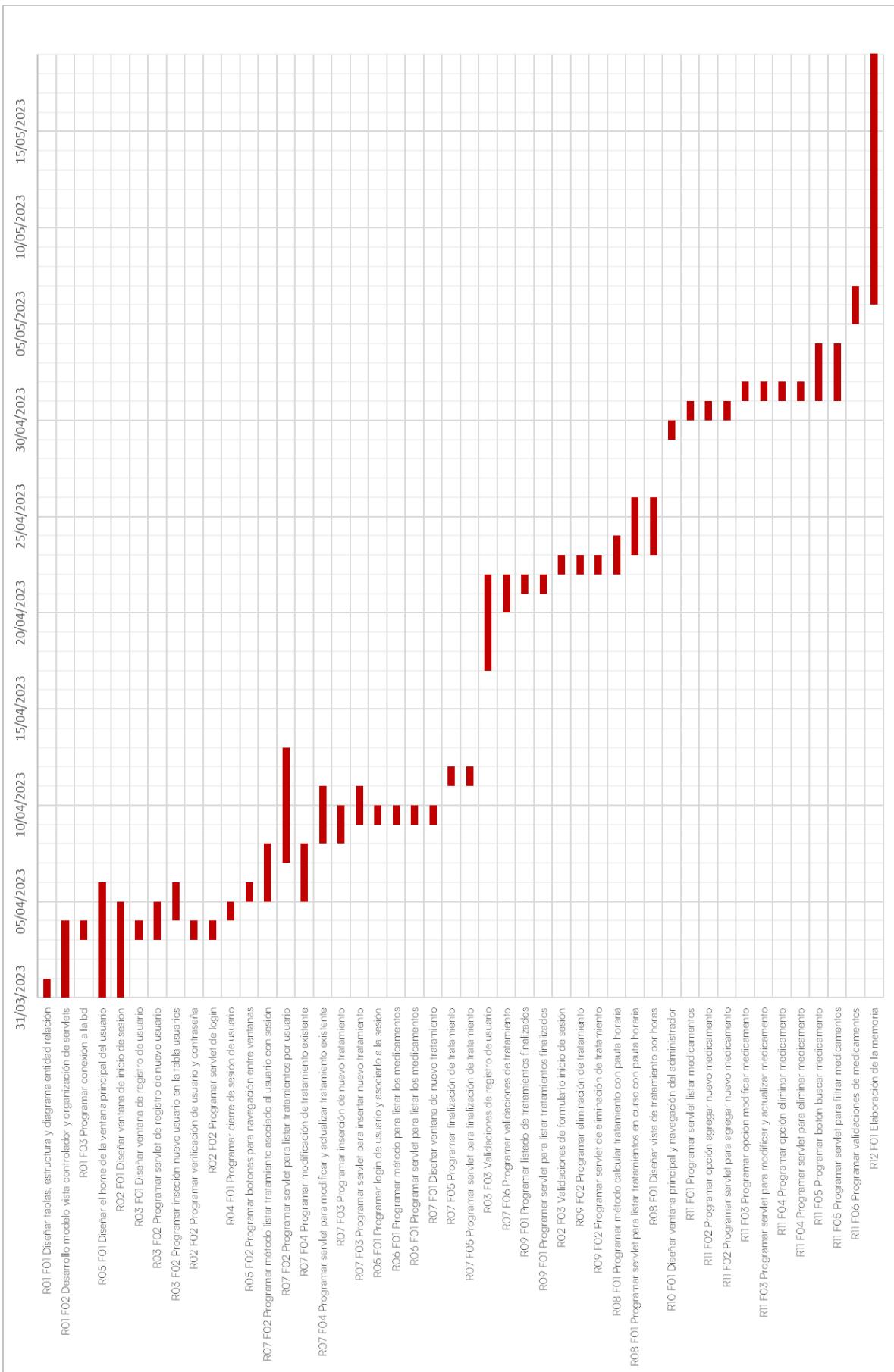


Ilustración 17: Diagrama de Gantt

## **TECNOLOGÍAS**

### **Java**

Este lenguaje de programación orientado a objetos se ha utilizado para implementar la lógica de la aplicación, para interactuar con la base de datos y procesar los datos. Se ha utilizado Java 1.8.0\_361.

### **Eclipse**

Se ha utilizado Eclipse IDE for Enterprise Java and Web Developers en su versión 2022-12 como entorno de desarrollo integrado (IDE). Esta edición Java EE de Eclipse está específicamente diseñada para el desarrollo de aplicaciones empresariales basadas en tecnologías como Servlets y JavaServer Pages (JSP).

### **JSP**

Es una tecnología utilizada para desarrollar páginas web dinámicas en Java. Permite incluir código Java en código HTML para generar el contenido dinámico en el servidor antes de enviarlo al cliente. Se ha utilizado para la creación de las vistas.

### **Bootstrap**

Se ha utilizado como framework de diseño web para crear estilos predefinidos en las vistas, construyendo las interfaces interactivas y enlazándolo con la utilización de un CSS personalizado.

### **Apache-Tomcat**

Tomcat es el servidor o entorno de ejecución para la aplicación web, ya que soporta la ejecución de varias especificaciones de Java como Java Servlets y Java server Pages (JSP).

## **Phpmyadmin**

Es una herramienta de administración con una interfaz gráfica de usuario para gestionar bases de datos MySQL. Con esta herramienta se ha creado la base de datos, las tablas y las relaciones, y se utiliza para la persistencia de datos de la aplicación.

## **Canvas**

Se ha utilizado esta herramienta para crear el contenido gráfico de logos, favicon y nombre del proyecto web.

## **Draw.io**

Esta herramienta de diagramación en línea se ha utilizado para crear diagramas de flujo, diagramas de entidad-relación, diagramas casos de uso para integrarlos a la memoria del proyecto web.

## PRUEBAS Y DOCUMENTACIÓN

### Validación y pruebas

La validación y las pruebas sobre el proyecto se han ido ejecutando al finalizar el diseño y la programación de cada funcionalidad. Mediante este checklist se comprueba que la aplicación cumple con los requisitos marcados en los objetivos.

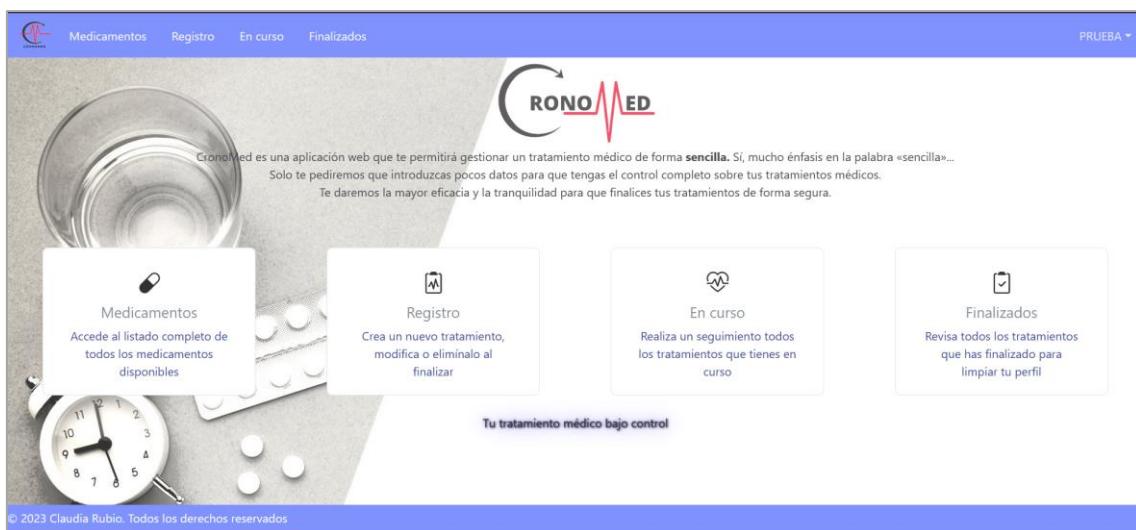


Ilustración 18: R02.F03.P01 - Prueba con usuario y password correcto

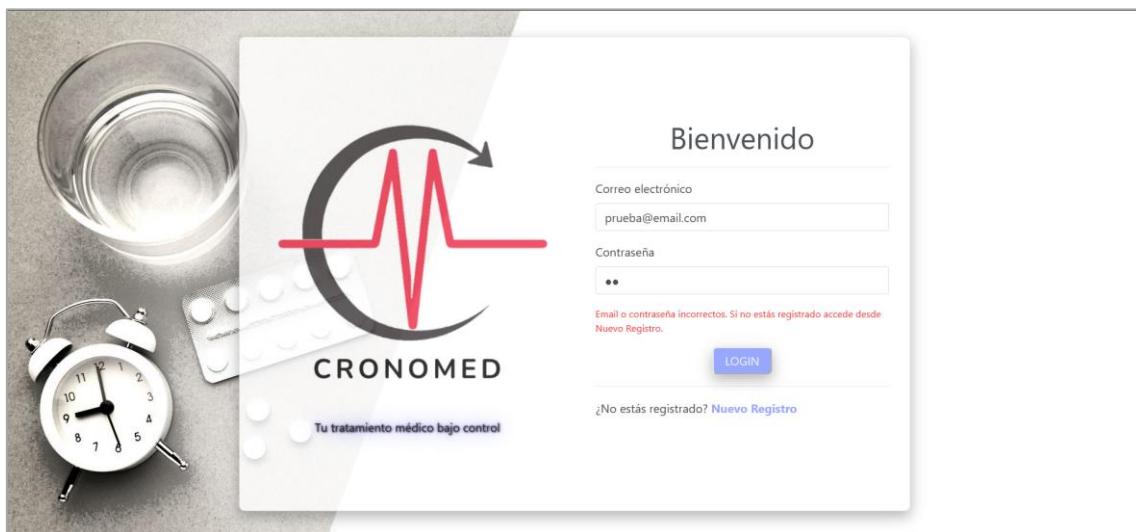


Ilustración 19: R02.F03.P02 - Prueba con usuario y password incorrecto

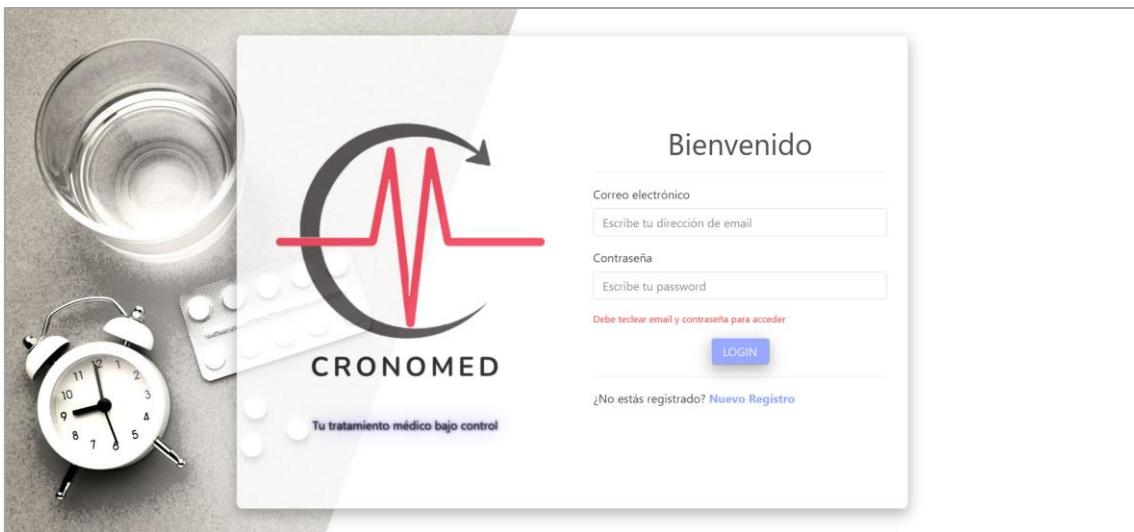


Ilustración 20: R02.F03.P03 - Prueba con datos vacíos



Ilustración 21: R03.F03.P01 - Prueba con nombre de usuario con caracteres no admitidos



Ilustración 22: R03.F03.P02 - Prueba con email incorrecto

**Formulario de Registro**

Nombre de usuario  
prueba

Email  
prueba@email.com

Contraseña  
\*\*\*\*\*

Escribe una clave de 6 caracteres  
El email prueba@email.com ya está registrado. Accede a la página desde Login.

**REGISTRO**

Si ya estás registrado, haz click en [Login](#)

Ilustración 23: R03.F03.P03 - Prueba con email ya registrado

**Formulario de Registro**

Nombre de usuario  
prueba

Email  
prueba@email.co

Contraseña  
\*\*\*\*\*

Escribe una clave de 6 caracteres  
La contraseña debe ser de 6 caracteres.

**REGISTRO**

Si ya estás registrado, haz click en [Login](#)

Ilustración 24: R03.F03.P04 - Prueba con password menor de 6 caracteres

**Formulario de Registro**

Nombre de usuario  
Escribe un nombre de usuario para tu perfil

Email  
Escribe tu dirección de email

Contraseña  
Escribe una password para tu acceso

Escribe una clave de 6 caracteres  
Debe completar todos los campos del formulario.

**REGISTRO**

Si ya estás registrado, haz click en [Login](#)

Ilustración 25: R03.F03.P05 - Prueba con datos vacíos

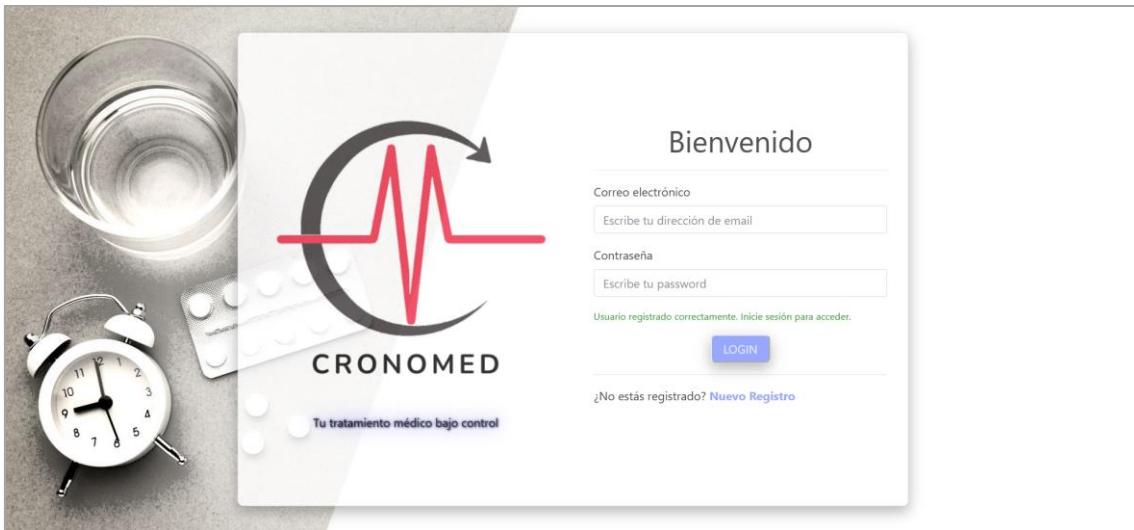


Ilustración 26: R03.F03.P06 - Prueba de registro con datos correctos

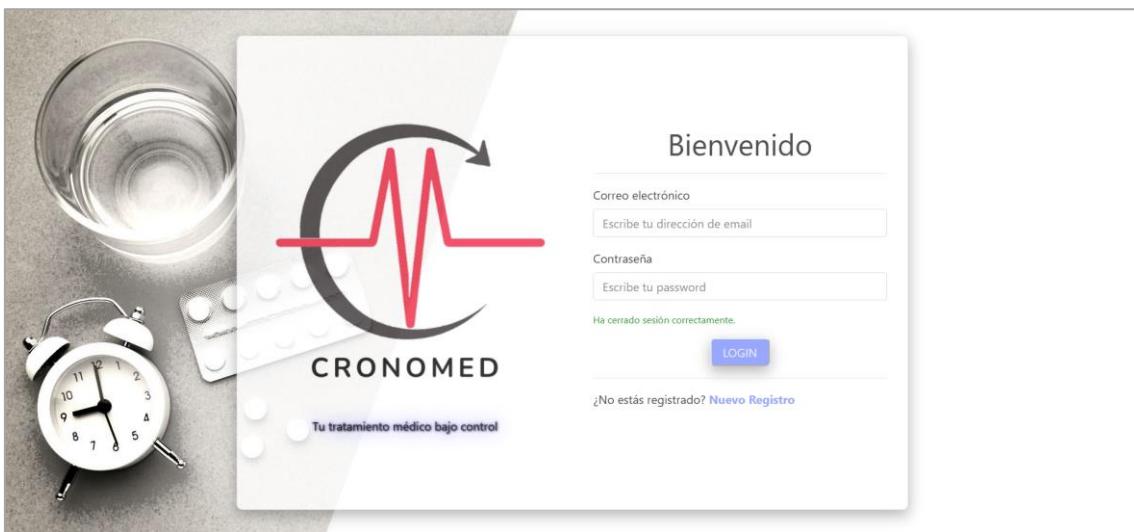


Ilustración 27: R04.F01.P01 - Prueba de cierre de sesión

MEDICAMENTO	
AERIUS 0,5 MG/ML	
ALMAX FORTE 1,5 G	
AMOXICILINA 250 MG/5 ML POLVO	
AMOXICILINA/ACIDO CLAVULANICO 500MG/125MG	
ANTALGIN 550 MG	
ASPIRINA 500 MG	
DESLORATADINA 0,5 MG/ML	
DIAZEPAN 5 MG	
DICLOFENACO 50 MG	
EBASTEL FORTE 20 MG	
ENANTYUM 25 MG	
ESTURONA 12,5 C	

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 28: R06.F01.P01 - Prueba para listar medicamentos

Medicamentos Registro En curso Finalizados PRUEBA ▾

### Registro de Tratamientos

» Registra un nuevo tratamiento: completa el formulario y pulsa en **Agregar**  
 » Modifica un tratamiento existente: seleccionalo en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Finaliza un tratamiento existente: pulsa a la izquierda del tratamiento, en el botón **Finalizar**

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	MODIFICAR	FINALIZAR
AERIUS 0,5 MG/ML	prueba paciente	1	Cada 12 hrs.	2 dias	prueba observaciones		

Selecciona un medicamento  
AERIUS 0,5 MG/ML  
Nombre del paciente  
Escribe el nombre del paciente  
Dosis en unidades (máx. 30)  
¿Cuál es la dosis?  
Pauta en horas  
¿Cada cuántas horas?  
Duración en días (máx. 15)  
¿Durante cuántos días?  
Observaciones  
Añade alguna observación

**AGREGAR** **ACTUALIZAR**

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 29: R07.F02.P01 - Prueba para listar tratamientos del usuario

Medicamentos Registro En curso Finalizados PRUEBA ▾

### Registro de Tratamientos

» Registra un nuevo tratamiento: completa el formulario y pulsa en **Agregar**  
 » Modifica un tratamiento existente: seleccionalo en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Finaliza un tratamiento existente: pulsa a la izquierda del tratamiento, en el botón **Finalizar**

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	MODIFICAR	FINALIZAR
AERIUS 0,5 MG/ML	prueba paciente	1	Cada 12 hrs.	2 dias	prueba observaciones		

Debes llenar todos los campos del formulario para agregar un nuevo tratamiento  
Selecciona un medicamento  
AERIUS 0,5 MG/ML  
Nombre del paciente  
nombre  
Dosis en unidades (máx. 30)  
1  
Pauta en horas  
¿Cada cuántas horas?  
Duración en días (máx. 15)  
¿Durante cuántos días?  
Observaciones  
Añade alguna observación

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 30: R07.F03.P01 - Prueba agregar tratamiento sin completar formulario

Medicamentos Registro En curso Finalizados PRUEBA ▾

### Registro de Tratamientos

» Registra un nuevo tratamiento: completa el formulario y pulsa en **Agregar**  
 » Modifica un tratamiento existente: seleccionalo en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Finaliza un tratamiento existente: pulsa a la izquierda del tratamiento, en el botón **Finalizar**

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	MODIFICAR	FINALIZAR
AERIUS 0,5 MG/ML	prueba paciente	1	Cada 12 hrs.	2 dias	prueba observaciones		

No has seleccionado ningún tratamiento para actualizar  
Selecciona un medicamento  
AERIUS 0,5 MG/ML  
Nombre del paciente  
Escribe el nombre del paciente  
Dosis en unidades (máx. 30)  
¿Cuál es la dosis?  
Pauta en horas  
¿Cada cuántas horas?  
Duración en días (máx. 15)  
¿Durante cuántos días?  
Observaciones  
Añade alguna observación

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 31: R07.F04.P01 - Prueba actualizar sin seleccionar tratamiento previamente

Medicamentos Registro En curso Finalizados PRUEBA +

### Registro de Tratamientos

» Registra un nuevo tratamiento: completa el formulario y pulsa en **Agregar**  
 » Modifica un tratamiento existente: seleccionalo en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Finaliza un tratamiento existente: pulsa a la izquierda del tratamiento, en el botón **Finalizar**

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	MODIFICAR	FINALIZAR
AERIUS 0,5 MG/ML	prueba paciente	1	Cada 12 hrs.	2 días	prueba observaciones		

Tratamiento finalizado correctamente. Puede consultar el historial en tratamientos Finalizados

Selecciona un medicamento  
AERIUS 0,5 MG/ML

Nombre del paciente  
Escribe el nombre del paciente

Dosis en unidades (máx. 30)  
¿Cuál es la dosis?

Pauta en horas  
¿Cada cuántas horas?

Duración en días (máx. 15)  
¿Durante cuántos días?

Observaciones  
Añade alguna observación

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 32: R07.F05.P01 - Prueba finalización de tratamiento

Medicamentos Registro En curso Finalizados PRUEBA +

### Registro de Tratamientos

» Registra un nuevo tratamiento: completa el formulario y pulsa en **Agregar**  
 » Modifica un tratamiento existente: seleccionalo en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Finaliza un tratamiento existente: pulsa a la izquierda del tratamiento, en el botón **Finalizar**

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	MODIFICAR	FINALIZAR
AERIUS 0,5 MG/ML	prueba paciente	1	Cada 12 hrs.	2 días	prueba observaciones		

El nombre no puede contener caracteres especiales.

Selecciona un medicamento  
AERIUS 0,5 MG/ML

Nombre del paciente  
Prueba!!!

Dosis en unidades (máx. 30)  
1

Pauta en horas  
1

Duración en días (máx. 15)  
1

Observaciones  
Añade alguna observación

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 33: R07.F06.P01 - Prueba con nombre de paciente con caracteres especiales

Medicamentos Registro En curso Finalizados PRUEBA +

### Registro de Tratamientos

» Registra un nuevo tratamiento: completa el formulario y pulsa en **Agregar**  
 » Modifica un tratamiento existente: seleccionalo en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Finaliza un tratamiento existente: pulsa a la izquierda del tratamiento, en el botón **Finalizar**

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	MODIFICAR	FINALIZAR
AERIUS 0,5 MG/ML	prueba paciente	1	Cada 12 hrs.	2 días	prueba observaciones		

Selecciona un medicamento  
AERIUS 0,5 MG/ML

Nombre del paciente  
paciente

Dosis en unidades (máx. 30)  
57

Seleccione un valor que no sea mayor de 30.

Duración en días (máx. 15)  
1

Observaciones  
Añade alguna observación

**AGREGAR** **ACTUALIZAR**

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 34: R07.F06.P02 - Prueba con número de dosis incorrecta

 Medicamentos Registro En curso Finalizados PRUEBA \*

### Registro de Tratamientos

» Registra un nuevo tratamiento: completa el formulario y pulsa en **Agregar**  
 » Modifica un tratamiento existente: seleccionalo en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Finaliza un tratamiento existente: pulsa a la izquierda del tratamiento, en el botón **Finalizar**

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	MODIFICAR	FINALIZAR
AERIUS 0,5 MG/ML	prueba paciente	1	Cada 12 hrs.	2 dias	prueba observaciones		

Selecciona un medicamento  
AERIUS 0,5 MG/ML

Nombre del paciente  
paciente

Dosis en unidades (máx. 30)  
1

Pauta en horas  
31 | Seleccione un valor que no sea mayor de 24.

Observaciones  
Añade alguna observación

**AGREGAR** **ACTUALIZAR**

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 35: R07.F06.PO3 - Prueba con pauta horaria incorrecta

 Medicamentos Registro En curso Finalizados PRUEBA \*

### Registro de Tratamientos

» Registra un nuevo tratamiento: completa el formulario y pulsa en **Agregar**  
 » Modifica un tratamiento existente: seleccionalo en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Finaliza un tratamiento existente: pulsa a la izquierda del tratamiento, en el botón **Finalizar**

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	MODIFICAR	FINALIZAR
AERIUS 0,5 MG/ML	prueba paciente	1	Cada 12 hrs.	2 dias	prueba observaciones		

Selecciona un medicamento  
AERIUS 0,5 MG/ML

Nombre del paciente  
paciente

Dosis en unidades (máx. 30)  
1

Pauta en horas  
1

Duración en días (máx. 15)  
31 | Seleccione un valor que no sea mayor de 15.

AGREGAR ACTUALIZAR

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 36: R07.F06.PO4 - Prueba con duración de tratamiento incorrecta

 Medicamentos Registro En curso Finalizados PRUEBA \*

### Registro de Tratamientos

» Registra un nuevo tratamiento: completa el formulario y pulsa en **Agregar**  
 » Modifica un tratamiento existente: seleccionalo en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Finaliza un tratamiento existente: pulsa a la izquierda del tratamiento, en el botón **Finalizar**

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	MODIFICAR	FINALIZAR
AERIUS 0,5 MG/ML	prueba paciente	1	Cada 12 hrs.	2 dias	prueba observaciones		
AERIUS 0,5 MG/ML	prueba paciente	5	Cada 8 hrs.	3 dias	sin observaciones		

Tratamiento agregado correctamente.

Selecciona un medicamento  
AERIUS 0,5 MG/ML

Nombre del paciente  
Escribe el nombre del paciente

Dosis en unidades (máx. 30)  
¿Cuál es la dosis?

Pauta en horas  
¿Cada cuántas horas?

Duración en días (máx. 15)  
¿Durante cuántos días?

Observaciones  
Añade alguna observación

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 37: R07.F06.PO5 - Prueba con datos correctos

Medicamentos Registro En curso Finalizados PRUEBA ▾

### Listado de Tratamientos en curso

A continuación, tienes un listado con la frecuencia horaria de todos los tratamientos que tienes en curso. Si quieres actualizarlos pulsa en la opción de **Registro** del menú de navegación.

MEDICAMENTO	PACIENTE	DOSIS	OBSERVACIONES	HORA
AERIUS 0,5 MG/ML	prueba paciente	1	prueba observaciones	12-05-2023 20:27:49
AERIUS 0,5 MG/ML	prueba paciente	1	prueba observaciones	13-05-2023 08:27:49
AERIUS 0,5 MG/ML	prueba paciente	1	prueba observaciones	13-05-2023 20:27:49
AERIUS 0,5 MG/ML	prueba paciente	1	prueba observaciones	14-05-2023 08:27:49

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 38: R08.F01.P01 - Prueba listar tratamiento con pauta horaria

Medicamentos Registro En curso Finalizados PRUEBA ▾

### Tratamientos Finalizados

Estos son los tratamientos que has finalizado. Puedes eliminarlos pulsando en **Eliminar**, pero recuerda!!! Si eliminas alguno definitivamente, no podrás recuperarlo.

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	ELIMINAR
PARACETAMOL 100 MG/ML	prueba paciente	2	Cada 8 hrs.	3 días	prueba observaciones	Eliminar
ASPIRINA 500 MG	prueba paciente	5	Cada 8 hrs.	3 días	sin observaciones	Eliminar

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 39: R09.F01.P01 - Prueba listar tratamientos finalizados

Medicamentos Registro En curso Finalizados PRUEBA ▾

### Tratamientos Finalizados

Estos son los tratamientos que has finalizado. Puedes eliminarlos pulsando en **Eliminar**, pero recuerda!!! Si eliminas alguno definitivamente, no podrás recuperarlo.

Tratamiento eliminado correctamente.

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	ELIMINAR
PARACETAMOL 100 MG/ML	prueba paciente	2	Cada 8 hrs.	3 días	prueba observaciones	Eliminar

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 40: R09.F02.P01 - Prueba eliminar tratamiento

 Medicamentos CLAUDIA \*

### >Listado de Medicamentos

Administrador, claudia

» Agrega un nuevo medicamento indicando su nombre y composición y pulsando en **Agregar**. Se permiten caracteres especiales: números y .,%  
 » Modifica un medicamento existente en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Borra un medicamento de la lista pulsando en **Eliminar**

NOMBRE Y COMPOSICIÓN DEL MEDICAMENTO	MODIFICAR	ELIMINAR
AERIUS 0,5 MG/ML		
ALMAX FORTE 1,5 G		
AMOXICILINA 250 MG/5 ML POLVO		
AMOXICILINA/ACIDO CLAVULANICO 500MG/125MG		
ANTALGIN 550 MG		

Nombre y composición del medicamento  Filtrar por nombre del medicamento   
   << Listar todos los medicamentos

Medicamento eliminado correctamente.

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 41: R11.F04.P01 - Prueba eliminar medicamento

 Medicamentos CLAUDIA \*

### Listado de Medicamentos

Administrador, claudia

» Agrega un nuevo medicamento indicando su nombre y composición y pulsando en **Agregar**. Se permiten caracteres especiales: números y .,%  
 » Modifica un medicamento existente en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Borra un medicamento de la lista pulsando en **Eliminar**

NOMBRE Y COMPOSICIÓN DEL MEDICAMENTO	MODIFICAR	ELIMINAR
PARACETAMOL 1 GR		
PARACETAMOL 100 MG/ML		
PARACETAMOL 650 MG		

Nombre y composición del medicamento  Filtrar por nombre del medicamento   
   << Listar todos los medicamentos

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 42: R11.F05.P01 - Prueba filtrar medicamento

 Medicamentos CLAUDIA \*

### Listado de Medicamentos

Administrador, claudia

» Agrega un nuevo medicamento indicando su nombre y composición y pulsando en **Agregar**. Se permiten caracteres especiales: números y .,%  
 » Modifica un medicamento existente en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**  
 » Borra un medicamento de la lista pulsando en **Eliminar**

NOMBRE Y COMPOSICIÓN DEL MEDICAMENTO	MODIFICAR	ELIMINAR
AERIUS 0,5 MG/ML		
ALMAX FORTE 1,5 G		
AMOXICILINA 250 MG/5 ML POLVO		
AMOXICILINA/ACIDO CLAVULANICO 500MG/125MG		
ANTALGIN 550 MG		

Nombre y composición del medicamento  Filtrar por nombre del medicamento   
   << Listar todos los medicamentos

El nombre no puede contener caracteres especiales.

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 43: R11.F06.P01 - Prueba con caracteres especiales no admitidos

Ilustración 44: R11.F06.PO2 - Prueba agregar medicamento con datos vacíos

Ilustración 45: R11.F06.PO3 - Prueba actualizar sin seleccionar medicamento previamente

Ilustración 46: R11.F06.PO4 - Prueba agregar medicamento correctamente

Medicamentos

CLAUDIA \*

**Listado de Medicamentos**

Administrador, claudia

» Agrega un nuevo medicamento indicando su nombre y composición y pulsando en **Agregar**. Se permiten caracteres especiales: números y ./%

» Modifica un medicamento existente en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**.

» Borra un medicamento de la lista pulsando en **Eliminar**.

Nombre y composición del medicamento

Filtrar por nombre del medicamento

BUSCAR

AGREGAR ACTUALIZAR LISTAR << Listar todos los medicamentos

Medicamento modificado correctamente.

NOMBRE Y COMPOSICIÓN DEL MEDICAMENTO	MODIFICAR	ELIMINAR
AA PRUEBA MEDICAMENTO 99.9%	Modificar	Eliminar
AERIUS 0,5 MG/ML	Modificar	Eliminar
ALMAX FORTE 1,5 G	Modificar	Eliminar
AMOXICILINA 250 MG/5 ML POLVO	Modificar	Eliminar
AMOXICILINA/ACIDO CLAVULANICO 500MG/125MG	Modificar	Eliminar

© 2023 Claudia Rubio. Todos los derechos reservados

Ilustración 47: R11.F06.P05 - Prueba modificar medicamento correctamente

## Manual de usuario

Se ha realizado un manual de usuario en el que se explican las funcionalidades y como se acceder a ellas en cada caso dependiendo de la acción que se desee realizar.

El manual va dirigido al perfil de usuario, por lo que se explica de manera sencilla sin términos técnicos. Puede consultarse el manual en el Anexo I.

## **CONCLUSIONES**

En este trabajo de fin de grado, se ha logrado desarrollar con éxito una aplicación web que cumple con todos los objetivos del proyecto. A pesar de no tener familiaridad con la tecnología JSP, se han adquirido los conocimientos necesarios para utilizarla de manera certera y eficaz en el desarrollo del proyecto. Se han enfrentado retos aprendiendo una tecnología diferente, realizando modificaciones a partir de una idea inicial, pero logrando mantener el enfoque en la funcionalidad u objetivo principal de calcular la frecuencia de administración de la medicación de una forma visual, práctica y sencilla.

Aunque el tiempo disponible para el desarrollo fue limitado, se ha logrado alcanzar una versión funcional de la aplicación. A lo largo del desarrollo del proyecto se han identificado necesidades que plantean oportunidades de mejora de la aplicación que no han podido ser abordadas en este período. Estas mejoras se han documentado en el apartado de trabajos futuros del proyecto. Entre las posibles mejoras se encuentran la incorporación de funcionalidades adicionales, como recordatorios de dosis, una interfaz de usuario más intuitiva y la posibilidad de generar informes de los tratamientos.

A pesar de las limitaciones de tiempo, el proyecto ha sido un éxito en términos de alcanzar el objetivo principal y adquirir experiencia en el desarrollo de aplicaciones web utilizando JSP. Los resultados obtenidos sientan las bases para posibles desarrollos futuros que permitan añadir las mejoras identificadas y expandir la funcionalidad de la aplicación.

## TRABAJOS FUTUROS

Con el objetivo de maximizar la sencillez de utilización de la aplicación por el usuario, se han identificado áreas de mejora. Estos trabajos se enfocan en optimizar la funcionalidad existente, mejorar la usabilidad y la experiencia del usuario, así como incorporar nuevas características.

La principal mejora es la optimización, llevando a cabo una revisión exhaustiva del código e identificar opciones de escalabilidad para garantizar un rendimiento óptimo.

Mejoras en la experiencia del usuario revisando el diseño de la interfaz y el flujo de navegación. Actualmente, en ventana reducida, el usuario debe volver al menú home de CronoMed para seleccionar una opción:

- Inclusión de menú “hamburguesa” para simplificar la interacción del usuario con la aplicación, de esta manera las funcionalidades estarían en un menú desplegable lateral.

Además, se plantea la incorporación de nuevas características y tecnologías que enriquezcan la experiencia del usuario:

- Una opción para programar la fecha y hora de inicio del tratamiento, o modificarlo para permitir el recálculo del tratamiento.
- Ordenar el listado de tratamientos en curso por hora.
- Implementación de notificaciones de administración de medicamento.
- Opción para imprimir la información de los tratamientos en curso.

## BIBLIOGRAFÍA

Para el desarrollo del proyecto, se han utilizado diversas fuentes de información, bibliografía y referencias relevantes que han sido fundamental para asegurar un enfoque sólido y actualizado en la implementación del proyecto web.

### Videos

Juan Díaz (Productor). (2017) Java EE. Módulo 1 y Módulo2 [Píldoras Informáticas]. De <https://www.pildorasinformaticas.es>

Juan Díaz (Productor). (2021) Curso Git [Píldoras Informáticas]. De [https://www.youtube.com/watch?v=ANF1X42\\_ae4&list=PLU8oAIHdN5BlyaPFiNQcVOxDqyOeR35aU&index=2](https://www.youtube.com/watch?v=ANF1X42_ae4&list=PLU8oAIHdN5BlyaPFiNQcVOxDqyOeR35aU&index=2)

sfw Libre (2015). Encriptar contraseñas [Tutorial Java]. De <https://www.youtube.com/watch?v=gdvGzP5TFw4>

SinFloo (2019). Sistema de Ventas en Java Web — Patrón MVC y My SQL [SinFloo]. De <https://www.youtube.com/watch?v=RPLrKKgnabc&t=6s>

### Blogs

Foros del web (2009). Sumar horas a una fecha [Mensaje en un blog]. Recuperado de <https://www.forosdelweb.com/f45/sumar-horas-fecha-684335/>

Parzibyte's blog (2018). Sumar y restar fechas en Java [Mensaje en un blog]. Recuperado por <https://parzibyte.me/blog/2018/10/16/sumar-restar-fechas-java/>

### Referencias de páginas

Agencia Española de Medicamentos y Productos Sanitarios (2017). Buscador de medicamentos. Recuperado por <https://cima.aemps.es/cima/publico/buscadoravanzado.html>

Agencia Española de Medicamentos y Productos Sanitarios (2019). CIMA REST API. Recuperado por [https://sede.aemps.gob.es/docs/CIMA-REST-API\\_1\\_19.pdf](https://sede.aemps.gob.es/docs/CIMA-REST-API_1_19.pdf)

Algoritmos y Algo más. (2023). *Login en Java web y MySQL*. Recuperado por  
<https://algoritmosyalgomas.com/login-javascript-usuario-password-conexion-base-datos-mysql-n-capas-parte2/>

Alura Latam (2020). Cómo convertir de String para Date en Java. Recuperado por  
<https://www.aluracursos.com/blog/como-convertir-string-para-date-en-java>

Oracle. (1993, 2020). *Java Platform, Standard Edition 7 API Specification*. Recuperado por  
<https://docs.oracle.com/javase/7/docs/api/>

Oracle. (1993, 2023). *Class DateTimeFormatter*. Recuperado por  
<https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html>  
!

Oracle. (1993, 2023). *Class ZonedDateTime*. Recuperado por  
<https://docs.oracle.com/javase/8/docs/api/java/time/ZonedDateTime.html>

Techie Delight (2021). *Obtenga la fecha y hora actual en Java*. Recuperado por  
<https://www.techiedelight.com/es/get-current-date-and-time-java/>

Universidad de Alicante. (2012). *Introducción a las aplicaciones web Java*. Recuperado por  
<http://www.jtech.ua.es/j2ee/restringido/cw/sesion01-apuntes.html>

## **Tesis y trabajos de grado**

Oscar Ruiz Rodríguez (2020). *Aplicación web para la gestión completa de los TFG* (trabajo de fin de grado). Universidad Autónoma de Madrid, Madrid

## GLOSARIO

**Apache Tomcat:** es un servidor web y contenedor de servlets para ejecutar aplicaciones web Java.

**Checklist:** es una lista de verificación utilizada para comprobar que se han realizado todas las tareas y evitar omisiones o errores.

**Clases DAO (Data Access Object):** clases Java que se encargan de interactuar con la base de datos, a través de métodos para realizar operaciones de lectura, escritura, actualización y eliminación de datos.

**JDBC (Java Database Connectivity):** es una librería que actúa como puente entre una aplicación Java y una base de datos relacional, permitiendo la comunicación y el intercambio de información.

**Java JSP (JavaServer Pages):** es una tecnología que mezcla código HTML estático con código Java para generar contenido web dinámico.

**JSTL (JavaServer Pages Standard Tag Library):** es una librería de etiquetas de JSP utilizada para la iteración de datos, entre otros.

**Modularización:** consiste en dividir un sistema complejo en partes más pequeñas y manejables con una funcionalidad específica.

**Servlets:** son programas escritos en lenguaje Java en entorno servidor para hacer consultas, inserciones, eliminaciones y actualizaciones de forma dinámica.

**Sprints:** es una unidad pequeña de tiempo en la que el trabajador (o equipo) se enfoca en el desarrollo de un conjunto de funcionalidades.

**WAR (Web Application Archive):** es un archivo comprimido utilizado para empaquetar y distribuir aplicaciones web en entornos Java.

**ANEXO I**



---

**Manual de Usuario**

---

## Índice

1.	Acceso a la aplicación .....	2
2.	Menú principal CronoMed .....	2
3.	Medicamentos.....	3
4.	Registro.....	3
4.1.	Agregar nuevo tratamiento.....	3
4.2.	Actualizar tratamiento existente.....	4
4.3.	Finalizar tratamiento.....	4
5.	En curso.....	4
6.	Finalizados.....	5
7.	Salir de la aplicación.....	5

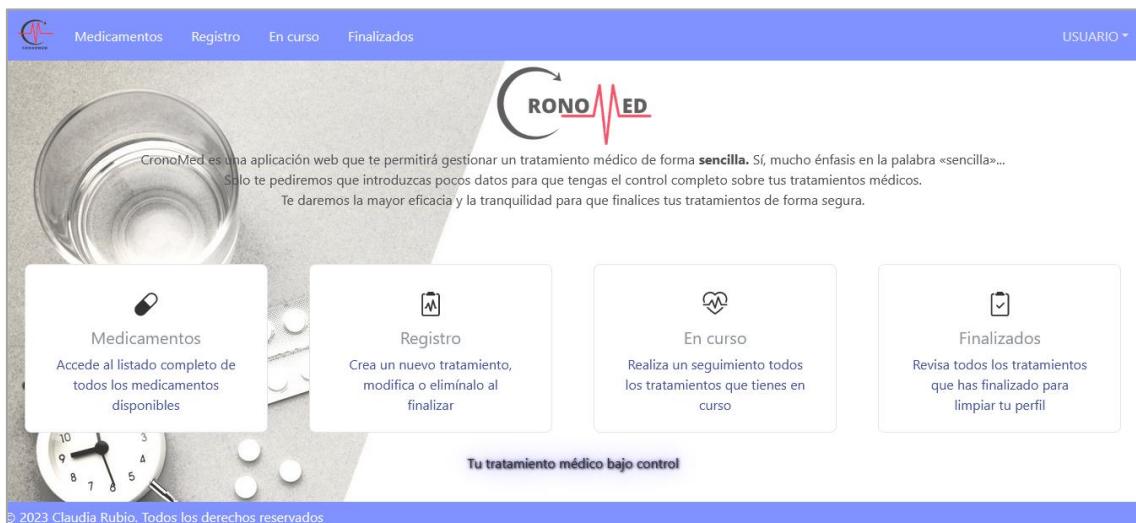
## 1. Acceso a la aplicación

Para acceder a la aplicación:

1. Si ya está registrado, complete el formulario de Bienvenida escribiendo dirección de email, contraseña de acceso y pulse el botón **LOGIN**
2. Si no estás registrado, desde la ventana de Bienvenida pulsa sobre la opción de **Nuevo Registro** y accede al formulario.
  - 2.1. En el formulario de registro, escriba un Nombre de usuario sin espacios ni caracteres especiales, una cuenta de correo email válido y una contraseña personal de 6 caracteres.
  - 2.2. Despues de haber completado los campos del formulario, pulse sobre el botón de **REGISTRO**. Esta opción le reenviará a la ventana de Login por lo que deberá seguir los pasos del punto 1.

## 2. Menú principal CronoMed

Al iniciar sesión, se tiene acceso al menú principal de la aplicación, con las opciones disponibles. Estas opciones están disponibles pinchando sobre cada ficha o desde el menú principal superior.



Para volver desde cualquier ventana a este menú se debe pinchar sobre el logo de CronoMed en el menú de navegación superior, imagen izquierda →

### 3. Medicamentos

Desde esta opción se tiene acceso a todo el listado de los medicamentos disponibles en la aplicación.

### 4. Registro

Desde esta ventana se puede gestionar un tratamiento médico, con las opciones de agregado, modificación y finalización. En todo momento se puede visualizar a la derecha de la ventana un listado con todos los tratamientos asociados a su perfil de usuario.

MEDICAMENTO	PACIENTE	DOSIS	PAUTA	DURACION	OBSERVACIONES	MODIFICAR	FINALIZAR
AERIUS 0,5 MG/ML	prueba paciente	1	Cada 12 hrs.	2 días	prueba observaciones	Modificar	Finalizar

Cada una de las siguientes opciones están descritas brevemente en la ventana de registro:

#### • Registro de Tratamientos

- » Registra un nuevo tratamiento: completa el formulario y pulsa en **Agregar**
- » Modifica un tratamiento existente: seleccionalo en la lista pulsando **Modificar**, modifica sus datos en el formulario y pulsa **Actualizar**
- » Finaliza un tratamiento existente: pulsa a la izquierda del tratamiento, en el botón **Finalizar**

#### 4.1. Agregar nuevo tratamiento

Para agregar un nuevo tratamiento a su perfil:

1. Completar los campos requeridos del formulario a la izquierda de la ventana: seleccione un medicamento de la lista desplegable, escriba nombre del paciente e indique la dosis con un máximo de 30 unidades, cada cuántas horas debe ser administrado el medicamento, y la duración del tratamiento con un máximo de 15 días.
2. No es lo habitual, pero si se necesita suministrar una dosis superior a 30 unidades o iniciar un tratamiento superior a 15 días, puede añadir otro “nuevo” tratamiento, o agregar un comentario en el campo de observaciones.

- Finalmente pulse en el botón  al final del formulario para añadir el tratamiento que verá actualizado en el listado de la derecha de la ventana.

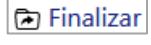
#### 4.2. Actualizar tratamiento existente

Para actualizar un tratamiento asociado a su perfil:

- Localice el tratamiento en el listado y pulse la opción  . Al realizar esta acción, el formulario de la izquierda se autocompleta con los datos del tratamiento.
- Modifique los datos del tratamiento en el formulario, tal y como se indica en el punto 1 del apartado anterior.
- Finalmente pulse el botón  al final del formulario para reemplazar el tratamiento del listado con los nuevos datos que ha tecleado.

#### 4.3. Finalizar tratamiento

Cuando haya finalizado un tratamiento:

- Localice el tratamiento en el listado y pulse la opción  . Al realizar esta acción, el tratamiento desaparecerá del listado de tratamientos asociados a su perfil.
- Todavía puede acceder a él desde la opción  del menú principal de navegación.

### 5. En curso

Desde esta opción tendrá acceso a un listado todos los tratamientos asociados a su perfil con la frecuencia horaria de administración de medicamentos, en el que podrá visualizar fecha y hora de administración, nombre del paciente, nombre de la medicación y dosis a administrar.

MEDICAMENTO	PACIENTE	DOSIS	OBSERVACIONES	HORA
AERIUS 0,5 MG/ML	prueba paciente	1	prueba observaciones	12-05-2023 20:27:49
AERIUS 0,5 MG/ML	prueba paciente	1	prueba observaciones	13-05-2023 08:27:49
AERIUS 0,5 MG/ML	prueba paciente	1	prueba observaciones	13-05-2023 20:27:49
AERIUS 0,5 MG/ML	prueba paciente	1	prueba observaciones	14-05-2023 08:27:49

## 6. Finalizados

Para eliminar un tratamiento asociado a su perfil de usuario:

1. Localice el tratamiento en el listado y pulse la opción Eliminar . Al realizar esta acción, el tratamiento desaparecerá del listado de tratamientos asociados a su perfil.
2. **NOTA IMPORTANTE.** Si elimina algún tratamiento no podrá recuperarlo, por lo que asegúrese que realmente quiere borrarlo de su historial.

## 7. Salir de la aplicación

Para salir de la aplicación:

1. Pulsar sobre su nombre de usuario, ubicado en la esquina superior derecha de la ventana de la aplicación USUARIO ▾
2. En el desplegable, pulse sobre el botón Salir para cerrar la aplicación.