
SanitApp

Axel Pérez Gaspar

I.E.S. Serpis

2º Curso – 2020/2021

Desarrollo de Aplicaciones Multiplataforma

Tutora: Manuela Ángeles Ripollés Orovitg

Este documento es la memoria del proyecto de final de ciclo de Desarrollo de Aplicaciones Multiplataforma.

En él se mostrará el detalle del mismo, así como la implementación y desarrollo de la aplicación.

También se analizará el Framework utilizado, en este caso Spring Boot-MVC junto con las tecnologías que han ayudado al desarrollo del proyecto.

Por último, se dará un repaso a las dificultades obtenidas y a los aprendizajes recibidos en el desarrollo del proyecto SanitApp, así como las posibles ampliaciones para futuras versiones.

Índice

- 1- Introducción
- 2- Tecnologías utilizadas en el proyecto
 - 2.1 Desarrollo del proyecto
 - 2.2 Software
 - 2.3 Hardware
- 3- Temporalización
- 4- Descripción del proyecto
 - 4.1 Detalle del proyecto
 - 4.2 Funcionamiento de la aplicación
 - 4.2.1 Acceso
 - 4.2.2 Primeros pasos
 - 4.2.3 Estructura de la aplicación
 - 4.2.4 Gestión de roles
 - 4.2.5 Tareas específicas para cada rol
 - 4.2.6 Proceso de negocio
 - 4.3 Datos técnicos de la aplicación
 - 4.3.1 Base de datos
 - 4.3.2 Lógica de negocio
 - 4.3.3 Frontend. Thymeleaf y Bootstrap
- 5- Dificultades
- 6- Contenidos relacionados con el ciclo
- 7- Aprendizajes alcanzados
- 8- Posibles ampliaciones del proyecto
- 9- Conclusiones finales
- 10- Bibliografía
- 11- Agradecimientos

1. Introducción

Para muchas empresas, tanto del sector sanitario como de muchos otros sectores, el hecho de no tener informatizado todo su proceso de negocio, puede suponer pérdidas tanto a nivel de tiempo (documentos e informes mal organizados en papel, lo que conlleva mucha dificultad a la hora de encontrar un determinado fichero) como a nivel económico.

El objetivo principal de este proyecto es dar una solución a dicho problema. Con una interfaz amigable, la aplicación permite al usuario llevar a cabo todas las operaciones necesarias para el buen funcionamiento de un negocio del ámbito de la salud, tanto para la gestión de pacientes (organización de sus fichas médicas) como para la organización de las citas y sesiones, así como el control de la facturación de las líneas de venta de los productos que puede tener dicho negocio.

2. Tecnologías utilizadas

2.1 Desarrollo del proyecto

Para este proyecto se ha decidido utilizar el Framework Java Spring-MVC, ya que es una tecnología muy potente, que permite desarrollar aplicaciones muy completas y con el código bien organizado y muy sencillo de desarrollar.

El proyecto consiste en una aplicación web, cuyo Frontend está basado en las tecnologías HTML + CSS.

Para el desarrollo de las plantillas HTML se ha utilizado como motor de plantillas Thymeleaf.

Thymeleaf es una biblioteca Java que sustituye al 100% las plantillas JSP, y es capaz de procesar todo tipo de lenguajes de marcas, tales como XML, XHTML o HTML5. Es capaz de trabajar junto a frameworks como Spring-MVC de una forma casi transparente. Sus principales características son las siguientes:

- Los @Controller de Spring pueden redirigirse directamente a las plantillas Thymeleaf
- Permite el uso de Spring Expression Language en las plantillas.
- Los formularios se integran perfectamente con los Beans y los Result Binding de java para el manejo de errores.
- Permite la internacionalización a través de los message.properties de Spring.

Para el desarrollo de los estilos, se han utilizado las librerías Bootstrap y jQuery, que combinan Javascript+CSS y ofrecen un resultado muy satisfactorio.

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para el diseño de sitios y aplicaciones web, capaz de adaptarse a multitud de frameworks, y proporciona una serie de elementos pre-formateados para los formularios, botones, menús de navegación, etc, así como extensiones de JavaScript adicionales.

Para el backend, se ha optado por usar Java Spring para la lógica de negocio y MySQL como base de datos relacional.

2.2 Software

El software utilizado para la realización del proyecto ha sido el siguiente:

- El IDE Spring Tools Suite 4 (basado en Eclipse).
- MySQL Workbench para la gestión de la base de datos.
- Exploradores de internet para el testeo (Google Chrome, Firefox e Internet Explorer)
- Sistema operativo Windows 10

2.3 Hardware

Para la realización del proyecto, se han querido utilizar diferentes dispositivos conectados en red para probar la aplicación con diferente hardware.

- Ordenador portátil con Linux
- Dispositivo móvil android.

3. Temporalización

En un principio se estimó que el proyecto tendría un coste de desarrollo de 60 horas, pero finalmente han resultado unas 70, ya que el horario ha sido de 2 horas diarias de media, y se ha realizado más o menos en 35 días.

Para el desarrollo de la aplicación, se han empleado unas 60 horas, y para la memoria unas 10 horas.

Los primeros días se emplearon principalmente en el aprendizaje del framework seleccionado, y a partir de la segunda semana de marzo se empezó con el desarrollo del proyecto.

4. Descripción del proyecto

4.1- Detalle del proyecto

La aplicación consta de los siguientes bloques:

- Gestión de pacientes: Organización de sus fichas médicas, de sus datos personales y las compras realizadas a la clínica.
- Gestión de citas: Dar de alta nuevas citas, así como gestión de las mismas.
- Gestión de ventas: Venta de productos sanitarios, tales como cremas, medicamentos, etc. Por parte de los médicos especialistas a los pacientes que los requieran para su tratamiento.

El rol de usuario corresponde al encargado de apuntar las citas de los pacientes, así como de cancelarlas si se diera el caso, y de asignárselas al especialista correspondiente.

El administrador, como es evidente, puede acceder a todos los campos de la aplicación.

4.2- Funcionamiento de la aplicación.

4.2.1. Acceso

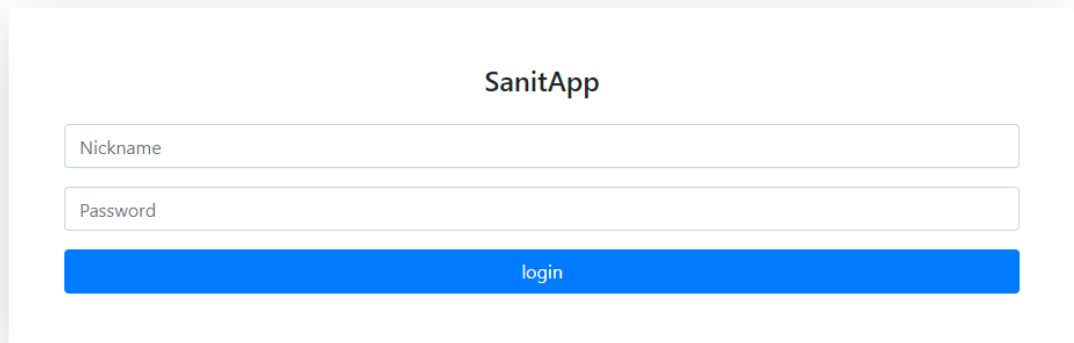
Al ser una aplicación web, el primer paso después de compilar el proyecto es acceder a un explorador de internet, e introducir en la barra de búsqueda la dirección IP junto con el puerto de donde se sitúe la aplicación.

Si se trabaja en local, se tendría que introducir la dirección <http://localhost:xxxx/>, (ej. <http://localhost:9000>) donde XXXX es el puerto del servidor que estemos usando (en este caso Tomcat). Este puerto se configura en el fichero application.properties del proyecto.

Si el proyecto estuviera almacenado en otro PC de una misma red local, se podría acceder, una vez compilada la aplicación, en la ruta <http://yyy.yyy.yyy.yyy:XXXX> (ej. <http://192.168.1.34:9000>)

4.2.2 Primeros pasos

Al inicio del programa, se abre la pantalla de login:

The image shows a login screen for an application named "SanitApp". At the top center, the text "SanitApp" is displayed. Below it, there are two input fields: the first is labeled "Nickname" and the second is labeled "Password". Both fields are rectangular with a light gray border. Below these fields is a solid blue button with the word "login" written in white text in the center.

Desde dicha ventana se podrá acceder a la aplicación utilizando cualquiera de los usuarios registrados.

Después se pueden dar 3 casos, dependiendo del rol que tenga el usuario que haya entrado.

- Si es un médico especialista: Su pantalla principal, con todos sus detalles, pacientes, citas etc.
- Si es un secretario/secretaria: La pantalla con todas las citas, tanto las programadas como las terminadas.
- Si es el administrador: Una pantalla principal. Desde donde podrá acceder a las distintas secciones de la aplicación.

4.2.3 Estructura de la aplicación

La aplicación se compone de un menú lateral desplegable con todas las secciones del programa:



- En la primera sección, aparece el nickname del usuario que ha entrado en la aplicación. Si es el usuario administrador, éste será “admin”. En el caso del secretario, será “secretario”, y en el caso de ser un especialista, será su DNI.
Al desplegarla hay dos opciones, “Mi perfil” y “Salir”.
 - La de mi perfil, lleva a una página diferente dependiendo del usuario que ha entrado. Si es un especialista, llevará a su perfil personal, si es el secretario llevará a la lista de citas, y si es el administrador a su página inicial.
 - La de salir, hará Logout de la aplicación.
- Luego llegaría la sección de pacientes. En ella se encuentran dos opciones:

SanitApp

- Ver pacientes: Mostrará la lista de pacientes. Desde esta lista se podrá acceder al detalle de cada paciente, y también, si se necesitara, se podrían actualizar sus datos.

Dentro del detalle del paciente se pueden observar las siguientes secciones:

- Datos personales: Aquí aparecen los datos personales del paciente, el diagnóstico si lo hubiera, y un enlace para actualizar sus datos.
- Listado de citas: Desde aquí se observan el listado de citas que ha tenido el paciente en la clínica, y se puede ver su estado. Si están pendientes, terminadas o canceladas.
- Listado de compras: Las compras que ha realizado el paciente, detallando los productos que ha comprado, el importe y el especialista que se la ha realizado.
- Nuevo paciente: Desde esta sección, se abrirá un formulario donde se podrá añadir a un nuevo paciente.
- A continuación vendría la sección de productos. En ella se encuentran dos opciones:
 - Ver productos: Desde aquí se pueden observar todos los productos que se ofrece en la tienda de la clínica.
 - Nuevo producto: Al pulsar en esta opción, se abre un formulario donde se pueden introducir nuevos productos.
- La siguiente sección es la de proveedores. Al desplegar el menú, se observan dos opciones:
 - Ver proveedores: muestra el listado de todos los proveedores con los que la clínica tiene algún contrato. En la línea de cada proveedor, se encuentran dos enlaces, uno para ir a sus detalles, y otro para actualizar los datos.

En el detalle del proveedor, además de sus datos personales, se puede observar un listado con los productos que ofrece dicho proveedor.

- Nuevo proveedor: Al pulsar en esta opción, se despliega un formulario para introducir un nuevo proveedor en la base de datos.

- A continuación se encuentra la sección de compras. En ella se encuentran las siguientes opciones:
 - Ver facturas: Muestra un listado con todas las ventas realizadas, desde donde se podrá acceder al detalle de cada una. Si se entra en el detalle, se observan las siguientes secciones:
 - Datos factura: los datos básicos que tiene dicha factura, como el cliente, el especialista que ha realizado la venta o el importe.
 - Productos: La lista de productos que contiene esa factura
 - Nueva venta: Desde aquí se podrá formalizar una venta a un paciente.

Se debe seleccionar el especialista que hace la venta, el paciente y los productos que se quieren vender.

Un aspecto importante es que si a la aplicación entra el administrador, cuando va a realizar alguna venta puede hacerla en nombre de cualquier especialista, en cambio si el que está logueado es un médico especialista, solo podrá realizar ventas en su nombre, no dejará elegir otro especialista, como es obvio.
- En la siguiente sección, se encuentra lo relativo a los especialistas. Esto es:
 - Ver especialistas: Muestra el listado de los especialistas. Desde este listado se podrá acceder al detalle de cada especialista, y también actualizar sus datos personales. Si accedemos al detalle, podemos observar las siguientes secciones:
 - Datos personales: los datos personales del especialista, y un enlace para actualizar sus datos.
 - Lista de pacientes que está tratando o ha tratado alguna vez
 - Las ventas realizadas
 - Las citas que tiene asociadas, ya estén pendientes o terminadas.

En este punto, el especialista puede acceder a la sesión correspondiente a esa cita, revisar como ha ido, hacer las anotaciones pertinentes, y plasmar el diagnóstico del paciente.
 - Nuevo especialista: Muestra un formulario para introducir un nuevo especialista en la base de datos.

- Por último, se encuentra la sección de las citas. En ella, tenemos las siguientes opciones:
 - Ver citas: Se muestra el listado de las citas. En este listado se puede observar un enlace en el que podemos actualizar la cita. En concreto de esta forma:
 - Cuando acabamos de registrar una cita, aparece como “pendiente”. Si esta en este punto, al pulsar en el enlace pueden pasar dos cosas. Si la cita va correctamente, en el momento en el que el especialista va a realizar la cita, el secretario/secretaria cambia el estado a “pendiente de revisar” y se abre un formulario donde puede asociar un especialista. Si el paciente decide cancelar la cita, desde ese formulario el secretario puede cancelarla. A partir de ese momento, dicha cita aparece en la página personal del especialista. En cuanto el especialista revisa la cita y toma las notas pertinentes, la cita pasa al estado de “revisada”.
 - Nueva cita: Desde esta sección se dará de alta una nueva cita, especificando el paciente, la fecha y la especialidad.

4.2.4 Gestión de roles

En base a los diferentes bloques de la aplicación, los usuarios que utilicen esta aplicación tendrán uno de los siguientes roles:

Rol	Gestión pacientes	Gestión citas	Gestión ventas
Administrador	Total	Total	Total
Usuario/Usuaría	No	Lectura/Escritura	No
Especialista*	Lectura*	Lectura/Actualización*	Lectura/Escritura*

*En cuanto a los médicos especialistas, tienen un sub-rol, que es su especialidad. Esto significa que solo podrá acceder a los bloques citados anteriormente, siempre y cuando correspondan a su especialidad.

Por ejemplo, un especialista con la especialidad de Acupuntor, solo podrá ver los pacientes relacionados con su especialidad, consultar sus citas asignadas, y lo mismo para las ventas, con la intención de que un especialista no pueda vender productos en nombre de otro, o modificar los datos de una sesión que no es la suya.

4.2.5 Tareas específicas para cada Rol

La idea de tener varios roles es que cada usuario pueda realizar una serie de acciones, para así cubrir todas las operaciones del negocio. Dichas tareas se repartirían de la siguiente forma:

- Administrador:

Este usuario es el encargado del mantenimiento de la aplicación, y es capaz de acceder a todas las secciones de la misma. Para cualquier operación específica que un usuario no pudiera realizar, debería pedir permisos a dicho usuario.

- Secretario/secretaria:

Este usuario es el encargado, principalmente, de tomar nota de las citas, o cancelarlas si el paciente no pudiera acudir a una determinada cita.

También se encargaría de dar de alta a los pacientes, productos, proveedores o especialistas nuevos.

En cualquier caso, no tiene potestad para realizar ventas de productos, ya que, al ser productos especializados de diferentes categorías médicas, es el especialista el que debe recetar dichos productos.

- Médico especialista:

Este usuario es el encargado de realizar las visitas médicas y realizar el seguimiento de los pacientes que tiene a su cargo. También puede vender productos a los mismos, así como actualizar sus datos personales.

Una cosa a destacar es que cada especialista tiene su especialidad, y esto permite que en todas las secciones a las que accede dicho usuario solo se muestren los datos relacionados con su propia especialidad.

Esta característica es muy potente, ya que con esto se consigue que los especialistas no puedan acceder a los datos de los otros especialistas, y el nivel de confidencialidad con el paciente es mucho mayor.

4.2.6 Proceso de negocio:

El modo de trabajo sería el siguiente:

- El secretario o secretaria atiende al teléfono, y programa las citas según la especialidad que requiera el paciente. Si el paciente llama para pedir cita en odontología, por ejemplo, el secretario la anotará para poder asignársela a los y las dentistas de la clínica. Si el paciente es la primera vez que llama a la clínica, registrará sus datos personales para poder llevar su ficha de forma más detallada.

- A partir de aquí pueden pasar dos cosas, que el paciente llame para cancelar la cita, o que la cita se realice sin problemas.

En el primer caso, es el secretario el encargado de marcar la cita como cancelada. En el segundo caso, se pasa a la siguiente fase.

- Una vez llega el día de la cita, y el secretario/secretaria sabe quién va a hacerle la visita al paciente, se la asignaría al especialista indicado.

- Cuando la cita ya se ha realizado correctamente, pasa al estado de “pediente de revisar”. Esto significa que el especialista debe valorar el resultado de la cita, anotar los datos que considere necesarios para el seguimiento del paciente, realizar un diagnóstico y describir el tratamiento a seguir.

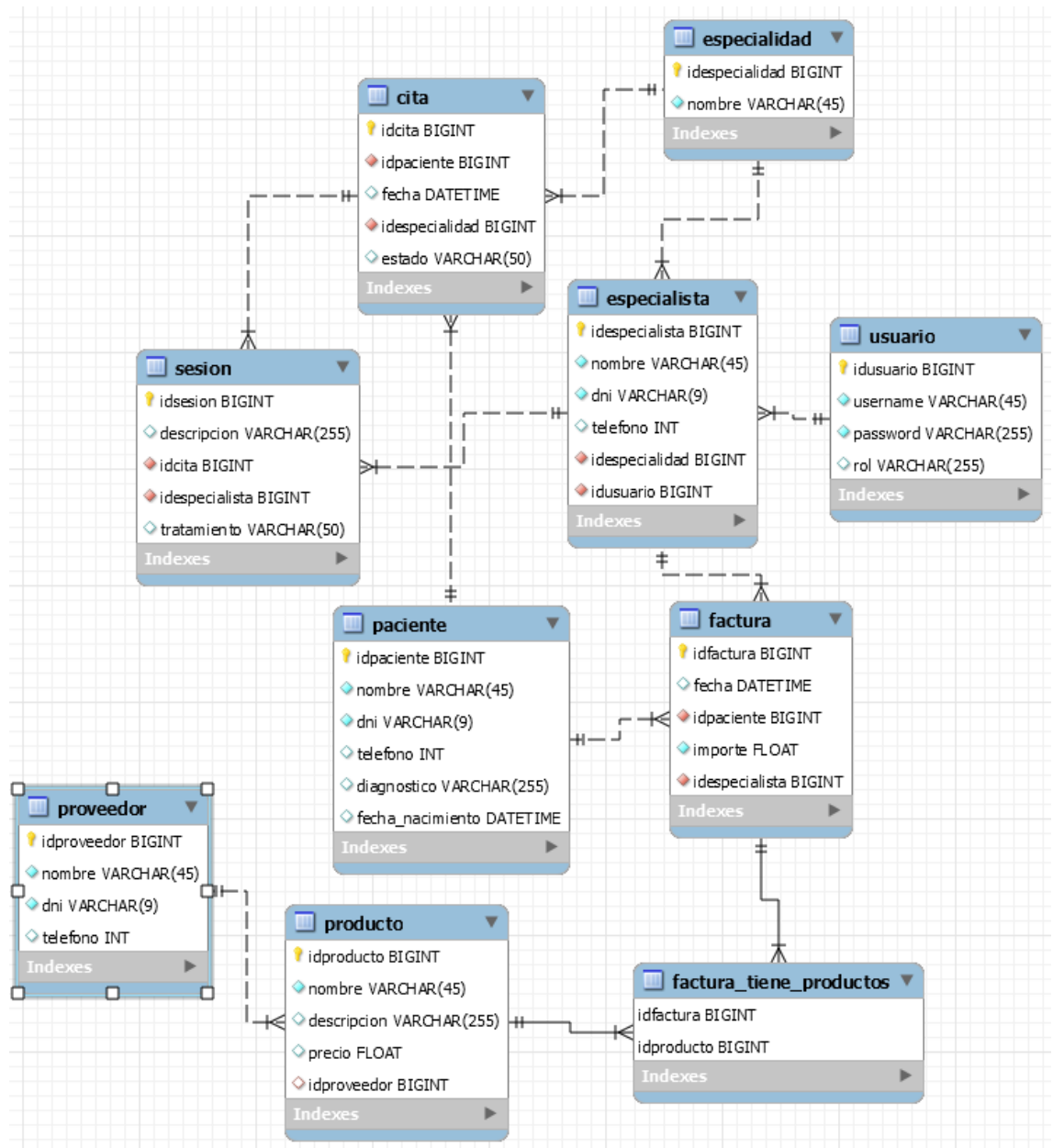
Si el especialista lo considera necesario, en este punto puede venderle productos al paciente que le ayuden a sobrellevar el tratamiento.

Además de esto, el secretario es el encargado de dar de alta en la tienda los productos que ofrecen los proveedores, así como, en el caso de hacer un contrato con un nuevo proveedor, poder dar de alta al mismo con sus datos personales, etc.

4.3- Datos técnicos de la aplicación

4.3.1. Base de datos

El esquema de la base de datos MySQL es el siguiente:



En el esquema se observan todas las tablas necesarias para el funcionamiento de la aplicación.

- Proveedor: Los datos personales de los proveedores registrados en la clínica.
- Producto: Los datos del producto. Tiene una clave ajena a la tabla Proveedor, que representa el vendedor que proporciona ese producto a la clínica.

SanitApp

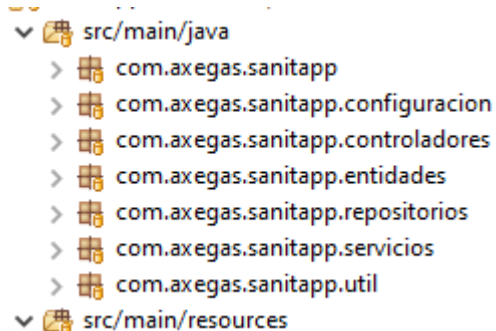
- Factura: Los datos de la factura, así como la clave ajena del especialista que ha realizado la venta, y la del paciente que ha realizado la compra.
- Factura y producto tienen una relación Many2Many que se plasma en la tabla “factura_tiene_productos”, ya que una factura puede contener muchos productos.
- Paciente: Datos personales del paciente, así como el diagnóstico que le realiza el profesional.
- Usuario: Es la tabla con todos los usuarios del sistema. Contiene, además del username, el password codificado y el rol que tiene en el sistema. Esta tabla contiene dos usuarios fijos: admin (administrador) y user (secretario). Por otro lado, contiene un usuario por cada especialista que tenga registrada la clínica.
- Especialista: Esta tabla contiene sus datos personales, así como una clave ajena que la relaciona con su usuario, y la clave ajena que describe la especialidad que ejerce dicho especialista.
- Especialidad: Esta tabla contiene las especialidades que ofrezca la clínica, tales como fisioterapia, quiromasaje, etc.
- Cita: Esta tabla tiene el estado de la cita (pendiente/cancelada/pendiente de revisar/revisada), la fecha prevista, la clave ajena del paciente que ha pedido la cita, y la especialidad de la misma.
- Sesión: Esta tabla corresponde al registro de las visitas realizadas por cada especialista, y contiene el informe detallado de lo que se ha realizado, el tratamiento a aplicar, y la clave ajena a la cita correspondiente a esa sesión, así como otra clave ajena que corresponde al especialista que ha realizado la cita.

Hay que remarcar ese matiz entre estas dos últimas tablas. La tabla sesión está directamente relacionada con la tabla cita, no se entiende una sesión sin su cita, es decir, todas las sesiones tienen que tener una cita asignada, y este campo tiene que ser único en la tabla, pero a su vez no todas las citas tienen que tener una sesión asignada, ya que pueden ser canceladas.

4.3.2. Lógica de negocio

En cuanto al código se puede destacar la arquitectura MVC que utiliza Spring Boot.

La lógica se distribuye en paquetes, para una mayor comprensión de la codificación de la aplicación.



Cabe destacar los siguientes paquetes:

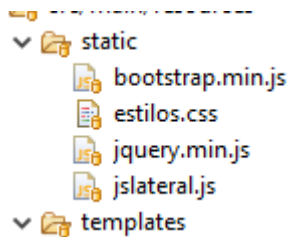
- Configuración: Aquí se almacenan las clases necesarias para la configuración de seguridad y los permisos de los diferentes roles.
- Controladores: Aquí se encuentran las clases que se encargan de gestionar las peticiones que hace el usuario, y en el caso que proceda llamar a los servicios para que pidan lo necesario a la base de datos.
- Entidades: Los objetos encargados de representar de forma lógica las tablas de la base de datos, para poder mapearlos desde Java, están en este paquete.
- Repositorios: Este paquete contiene las interfaces que hacen de enlace entre el código java y la base de datos MySQL. Hibernate es el encargado de mapear esas entidades y hacer que se puedan realizar todas las operaciones CRUD de forma automática.
- Servicios: Los servicios son las clases que implementan las operaciones que vienen de los repositorios, para que los controladores instancien directamente el servicio necesario para la ejecución de una determinada operación.
- Util: es un paquete donde se almacenan diversas clases de ayuda para hacer más sencilla la codificación.

4.3.3. Frontend. Thymeleaf y Bootstrap

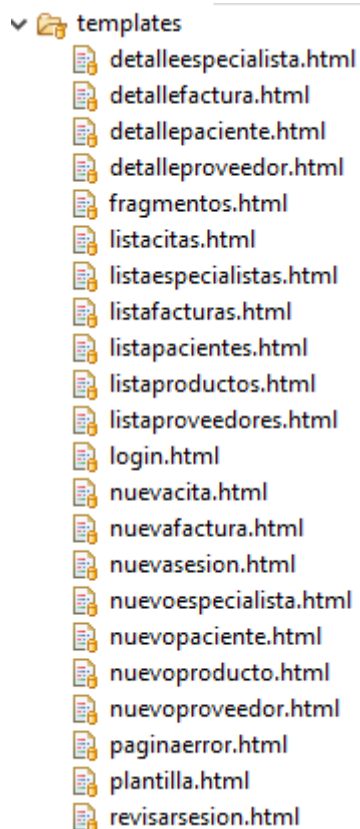
Thymeleaf y bootstrap ofrecen todo lo necesario para poder realizar las templates de una forma sencilla y con mucha calidad.

El esquema de ficheros que se ha seguido es el siguiente:

- Estilos y JavaScript:



- Plantillas:



Para el nombre de las plantillas, siempre se ha seguido el mismo patrón, una palabra que describe la acción que se realizará en la plantilla, seguida de la entidad a la que hace referencia, o a una determinada sección. Por ejemplo, listacitas

muestra el listado de las citas, detalleespecialista: el detalle de un determinado especialista, etc.

Además de las plantillas principales, tenemos unas plantillas específicas:

- Paginaerror: se muestra cuando el servidor devuelve un error, ya sea porque el usuario intenta acceder a una sección de la que no tiene permisos, o un error del propio servidor.
- Login: la página de login de la aplicación.
- Fragmentos: Thymeleaf ofrece una utilidad que se llama “fragments”, que permite acceder a una determinada porción de código que esté anotada con una determinada etiqueta desde todas las plantillas que apunten a esa etiqueta. Teniendo en cuenta esta potente utilidad, en esta plantilla se almacenan diversos fragmentos utilizados en el resto de la aplicación.

5. Dificultades

Teniendo en cuenta que se ha optado por utilizar un framework que no se ha trabajado en el ciclo, se han tenido algunas dificultades en el desarrollo del proyecto.

Para poder sobrellevarlas se ha tenido que recurrir a diversos manuales, tutoriales y cursos.

Entre ellas, podemos destacar las siguientes:

- Gestión de seguridad y roles.

Este aspecto es un sistema bastante importante en cualquier aplicación informática, sobre todo en los sistemas de gestión, y como tal en éste proyecto se ha querido incidir en una buena gestión en ese sentido. El problema viene a que en los frameworks estudiados en el ciclo nunca se han desarrollado este tipo de aspectos, y ha sido todo un reto poder solventarlo. Para ello, Spring Boot ofrece una solución tremendamente sencilla, y con unas pocas líneas de código se pueden dar los permisos necesarios.

Implementando una clase con la etiqueta `@Configuration` que extienda de la clase `WebSecurityConfigurerAdapter`, con el siguiente sencillo método podemos configurar los permisos que tendrá cada rol en nuestra aplicación:

SanitApp

```

@Override
protected void configure(HttpSecurity http) throws Exception {

    http
        .authorizeRequests()
            .antMatchers("/webjars/**", "/static/**", "/css/**").permitAll()
            .antMatchers("/perfil/**", "/facturas/**", "/pacientes/detalle/**").hasAnyRole("JOB", "ADMIN")
            .antMatchers("/especialistas/**", "/pacientes/**", "/citas/**",
                "/proveedores/**", "/productos/**").hasAnyRole("USER", "ADMIN")
            .anyRequest().authenticated()
        .and()
        .formLogin()
            .loginPage("/login")
            .permitAll()
        .and()
        .logout()
            .logoutUrl("/logout");
}

```

En este método anotamos, mediante el método “antMatchers”, las ubicaciones de nuestra aplicación web, y a continuación se añade la etiqueta que describe los roles que pueden acceder a esas ubicaciones.

- Gestión de estilos CSS.

Uno de los aspectos que más han costado de aplicar son los estilos, ya que pese a que Bootstrap es una herramienta que facilita mucho la vida de un programador en ese aspecto, sí que es cierto que es necesario tener unos conocimientos importantes de CSS.

No obstante, estudiando un poco sobre Bootstrap se pueden ir aplicando estilos muy interesantes sin tener que leer muchos manuales.

- Errores en la resolución gráfica del proyecto.

Uno de los principales problemas de las aplicaciones, tanto de escritorio como para dispositivos móviles, es la resolución. No es lo mismo ejecutar una aplicación en una pantalla de 15 pulgadas que en una de 20 o de 24.

Por este motivo, se han tenido dificultades en el desarrollo de la aplicación para poder adaptar el programa a diferentes pantallas.

Finalmente se ha intentado, mediante estilos CSS, adaptar de forma proporcional a cada pantalla la vista del proyecto.

- Aprendizaje del framework.

Para el desarrollo del proyecto, se ha tenido que invertir un tiempo en familiarizarse con la arquitectura Spring boot-MVC, leyendo manuales, viendo tutoriales, y realizando una serie de aprendizajes para solventar el problema de utilizar una tecnología nunca vista por el desarrollador.

- Funcionamiento de una clínica a nivel conceptual.

Es difícil realizar un proyecto sobre un determinado aspecto ajeno al campo de trabajo o estudio del desarrollador informático, ya que en esta aplicación en concreto, no se tienen conocimientos previos del funcionamiento de una clínica médica, y eso puede llevar a algunos errores a nivel conceptual muy grandes.

Para solventarlos, se ha recurrido a terceras personas trabajadoras en el campo de la salud para informarse y consultar determinados aspectos y dudas en el desarrollo del proyecto, para no caer en sinsentidos.

6. Contenidos relacionados con el ciclo

Para la realización de este proyecto, se han requerido unos conocimientos básicos aprendidos durante el ciclo. Entre ellos, se destacan los siguientes:

- ADA. Toda la parte de programación, acceso a base de datos y conocimiento del lenguaje java, ha sido indispensable para la realización del proyecto. También fue la profesora de ADA la que nos introdujo el framework Spring, aunque no lo llegamos a desarrollar.
- DI. Para la parte del front, el aprendizaje del framework Electron, aunque no es exactamente la misma tecnología, sí que ha servido para conocer un poco mejor el lenguaje HTML, CSS y Javascript, que han sido indispensables para el desarrollo del proyecto.
- SGE. A nivel conceptual, el hecho de conocer diferentes ERP y los módulos que contienen, ha servido para tener una visión clara de lo que puede necesitar una empresa y qué estructura debe seguir.

7. Aprendizajes alcanzados

A nivel técnico, se han obtenido sobretodo conocimientos del framework Spring Boot-MVC, del motor de plantillas Thymeleaf y del gestor de estilos Bootstrap.

Al trabajar con estas tecnologías, se ha desarrollado un dominio en dicho framework, que es uno de los más potentes en el mundo de la programación, y para un futuro programador es muy relevante este tipo de conocimientos.

Pese a la aparición de otros frameworks también muy potentes como Angular, Django o React, Spring boot no se queda atrás en cuanto a potencia y versatilidad.

8. Posibles ampliaciones

Este proyecto tiene el objetivo de ofrecer al negocio las herramientas necesarias para hacer que su trabajo sea más efectivo. Pero tanto a nivel técnico como a nivel operativo hay que tener en mente, para futuras versiones, nuevas funcionalidades y herramientas que hagan el programa más eficiente o que le den una mayor satisfacción al usuario.

Para nuevas actualizaciones del programa, se han contemplado las siguientes ampliaciones:

- Borrado de información

Dado que el tiempo de desarrollo de la aplicación era limitado, y que borrar información de la base de datos puede llevar a problemas del sistema, se ha optado por dejar para una futura versión una implementación bien hecha que pueda ofrecer con toda seguridad la función de borrar datos, tanto de pacientes, facturas, productos etc.

- Aviso por e-mail:

Se ha planteado que sería muy útil el hecho que la aplicación avisara al paciente, vía e-mail, de que tiene una cita programada. Tanto a nivel de recordatorio para el paciente como a nivel administrativo para la empresa, ya que tener un registro de las citas en el correo puede ser muy interesante.

- Recordar contraseña:

Pese a que el control de usuarios está muy avanzado, dado que es un prototipo no se ha incidido en el control de las contraseñas. En una versión más estable de la aplicación, habría que añadir un sistema para recordar la contraseña en caso que el usuario la haya olvidado, poder modificarla, etc.

- Calendario:

Otra utilidad muy interesante sería la implementación de un calendario para que el secretario pudiera ver, de forma más visual e interactiva, las citas pendientes, los huecos libres de cada especialista, etc.

- Generar informes en PDF:

Sería interesante que se pudieran generar ficheros PDF tanto de los informes de los pacientes, como de su ficha personal, de las facturas realizadas, para poder, de ser necesario, llevarse la información en un pendrive, o por ejemplo enviar una factura por e-mail.

- Gestionar albaranes de los proveedores.

Un aspecto muy importante en un negocio que ofrezca una línea de venta de determinados productos, es el poder gestionar los albaranes de los proveedores.

- Control de stock:

Otra cosa interesante que se podría incluir en futuras versiones es el control de stock. Es importante saber, en todo momento, los productos disponibles en la tienda, para realizar pedidos en caso de ser necesario.

- Añadir fotos al detalle de las sesiones:

Una cosa que sería interesante es integrar la funcionalidad de insertar fotos, radiografías, etc. en el historial de los pacientes. Es muy útil y muy visual para el especialista tener esa herramienta.

- Aplicación móvil

Sería muy interesante poder acceder a la agenda, a los detalles de los pacientes, a las facturas realizadas, desde cualquier sitio con una sencilla aplicación de móvil.

Sí que es cierto que, como se ha probado durante el desarrollo del proyecto, mediante el explorador web nativo del móvil (en mi caso lo he testeado con un móvil Android) se puede acceder a la aplicación, pero sería muy interesante poder acceder a la misma a través de internet.

- Mejorar la búsqueda de información.

A nivel operativo, es muy útil poder incluir filtros en las búsquedas, ya que en el caso de haber mucha información en la base de datos, es muy incómodo buscar en un listado que contenga muchos campos sin ningún tipo de orden ni filtro. Para ello, se contemplan las siguientes utilidades:

- Filtrar las búsquedas por un criterio determinado

- Ordenar las tablas según el criterio que el usuario elija, ya sea por precio en el caso de los productos, por fecha, alfabéticamente o de cualquier otra forma.
- Pagar los resultados. Marcar previamente un número de elementos por página, para que no muestre todos los resultados de golpe, ya que eso implicaría, en el caso que en la base de datos haya mucha información, una sobrecarga del sistema, a la vez que para el usuario sería más incómodo encontrar la información requerida.

9. Conclusiones finales

En general el proyecto ha sido difícil de realizar, con muchas complicaciones debido a la complejidad de la aplicación, el poco tiempo debido a las FCT y muchos otros aspectos, como el hecho del estudio de una arquitectura totalmente nueva como es Spring Boot-MVC, tener un conocimiento básico del motor de plantillas Thymeleaf o el gestor de estilos Bootstrap.

Al tiempo dedicado al propio desarrollo de la aplicación, se le añaden esas horas que en cuanto a la producción del propio programa no son productivas, pero que cumplen una función fundamental en el desarrollo del programa: Aprender.

Al final, el objetivo tanto del proyecto como del curso en sí es obtener conocimiento para poder aplicarlo en la vida profesional como programador, por eso se valora enormemente el usar unas tecnologías nuevas, que no se han ofrecido en el ciclo, pero que mediante la investigación y la búsqueda personal se pueden aprender de forma autodidacta y trabajar con ellas de una forma autónoma.

Por otro lado, el hecho de que este framework le saque todo el partido a la potencia que tiene el lenguaje de programación Java, hace que desarrollar aplicaciones sea mucha más cómoda, y el código quede más legible y la interfaz mucho más amigable.

También cabe destacar el aprendizaje obtenido respecto al mundo sanitario, del funcionamiento de una clínica, las distintas especialidades médicas, tratamientos y diagnósticos que existen, ya que pese a no ser unos conocimientos relacionados con el mundo de la programación, no dejan de ser conocimientos remarcables y que en un futuro tal vez puedan ser útiles para la vida personal o profesional.

10. Bibliografía.

11. Agradecimientos.

No se podría haber desarrollado este proyecto sin la ayuda de mi pareja, que como osteópata y masajista me ha aconsejado sobre como plantear ciertos aspectos referentes a su campo, a su forma de trabajar, y ha sido la que me ha dado la inspiración para realizar este proyecto.