



APLICACIÓN WEB PARA EL SECTOR FARMACÉUTICO SIRONA

M.ª Rosa Martínez Paredes

Grado superior en desarrollo de aplicaciones web

Trabajo fin de grado

Tutor de TFG: Enrique Tortajada González

Tutor de prácticas: Xavier Venancio Rosillo Guerrero

Convocatoria Junio

Ficha técnica del trabajo final

Título del trabajo: Sirona

Nombre del autor: M.^a Rosa Martínez Paredes

Nombre del consultor: Enrique Tortajada González

Fecha de entrega: 30/05/2022

Titulación: Desarrollo de aplicaciones web

Idioma del trabajo: Castellano

Resumen

La aplicación propuesta consiste en controlar y gestionar tanto el personal como los productos que tiene una farmacéutica.

Además, se controlará el acceso por diferentes roles con distintos privilegios para no tener acceso íntegramente a la aplicación, eso supone la limitación en algunas partes.

Esta aplicación podrá ser utilizada por cualquier empresa farmacéutica y se adaptaría a las necesidades de cada una de ellas personalizando cualquier tipo de petición por el cliente en cuestión.

ABSTRACT

The proposed application consists of controlling and managing both the products that a pharmaceutical company has.

In addition, access will be controlled by different privileges so that the entire application is not accessed, which means that access to different parts of the application will be limited.

This application can be used by any pharmaceutical company and would adapt to the needs of each of them, customizing any type of request by the client in question.

CONTENIDO

CAPITULO 1: INTRODUCCIÓN

1.1	INTRODUCCIÓN	5
1.2	MOTIVACIÓN Y OBJETIVOS	6

CAPÍTULO 2: VIABILIDAD DEL PROYECTO

2.1	DESCRIPCIÓN DE LA EMPRESA DESARROLLADORA	7-8
2.1.1	CLIENTE FINAL	7
2.1.2	NECESIDADES Y REQUERIMIENTOS DEL CLIENTE	7
2.1.3	IMAGEN CORPORATIVA Y NOMBRE	8
2.2	VIABILIDAD DE MERCADO	9
2.3	VALORACIÓN ECONÓMICA DEL PROYECTO	10
2.4	VIABILIDAD ECONÓMICA FINANCIERA	10-11
2.5	RECURSOS NECESARIOS	12-13
2.6	MARKETING	13

CAPÍTULO 3: ANÁLISIS TÉCNICO DE LOS ENTORNOS SERVIDOR Y CLIENTE

3.1	CASOS DE USO	14
3.2	DESCRIPCIÓN DE LA ESTRUCTURA DEL PROYECTO	15-16
3.2.1	METODOLOGÍA ÁGIL SCRUM	16-23
3.3	ESPECIFICACIÓN DEL MODELO DEL DATOS	23-27
3.3.1	TABLAS EN BASE DE DATOS	23-25
3.3.2	DIAGRAMA ENTIDAD-RELACION	23-27

CAPITULO 4: CONCLUSIONES Y POSIBLES ACTUALIZACIONES

4.1	CONCLUSIÓN	28
4.2	POSIBLES ACTUALIZACIONES	28

CAPITULO 5: BIBLIOGRAFÍA	29-31
--------------------------------	-------

CAPÍTULO 1: INTRODUCCIÓN

1.1 INTRODUCCIÓN

En estos últimos años, con el inicio de la crisis de la COVID-19, la farmacia comunitaria ha demostrado su compromiso social y sanitario con esfuerzos constantes para garantizar el acceso a los tratamientos, asistir a los pacientes vulnerables y responder a las dudas de la ciudadanía.

Esta crisis ha conllevado a un cambio en el sistema sanitario, las farmacias se han mantenido en primera línea, implementando nuevos servicios asistenciales para contribuir a la salud pública y haciendo frente a la situación sanitaria compleja provocada por el coronavirus.

Hace tiempo que la farmacia vive un proceso de transformación digital con el fin de adaptarse a los nuevos tiempos y a los cambios en los hábitos de los pacientes. Sin embargo, la situación de la COVID-19 ha acelerado este proceso y ha forzado la implementación de cambios que todavía no estaban previstos. Para hacer frente al problema, se han instaurado medidas como la receta electrónica privada. Además, muchas farmacias emplearon la consulta telemática para atender a los pacientes. Por todos estos motivos, aprovechamos la oportunidad de crear una aplicación que ofrezca a las farmacéuticas trabajar de una manera ágil y adaptándonos a las necesidades de cada una de ellas.

1.2 MOTIVACIÓN Y OBJETIVOS

La motivación principal por el cual se lanza esta aplicación web es debido a que para un país es uno de los sectores fundamentales para la economía, por ello, se ve la necesidad de adaptar las tecnologías para que el servicio sea lo más rápido posible y actualizado, sobre todo desde la pandemia por Covid-19, ya que nos vimos en una situación límite debido a que las farmacéuticas se vieron obligadas a verse desabastecidas por el terror que tenía la ciudadanía y la preocupación de quedarse sin lo necesario para salir de sus casas.

El principal objetivo de Sirona es automatizar el control de todo lo que requiere una farmacéutica integrando las nuevas tecnologías adaptándose a cada empresa, es decir, dependiendo de la farmacéutica se adaptará la web a sus propias necesidades. Sirona puede **gestionar** todo tipo de productos y empleados, **controlar** la situación de cada uno de ellos y por último **administrar** el almacén y a sus empleados.

El objetivo final de esta web sería estudiar a gran escala la comunicación entre farmacéuticas para que haya una integración entre ellas y que puedan disponer de una app lo más actual posible siempre con la finalidad de adaptarnos a sus propias características.

CAPÍTULO 2: VIABILIDAD DEL PROYECTO

2.1 DESCRIPCIÓN DE LA EMPRESA DESARROLLADORA

2.1.1 CLIENTE FINAL

La aplicación web está basada en un público de habla hispana, además de esto Sirona es una aplicación que ha sido desarrollada para ser utilizada por cualquier empresa del sector farmacéutico, es decir, que el cliente final son todos los trabajadores que conforman una empresa.

2.1.2 NECESIDADES Y REQUERIMIENTOS DEL CLIENTE

La aplicación ofrece una serie de servicios que el cliente necesita para poder gestionar una empresa de una manera sencilla y llevar el control con exactitud tanto de sus empleados como de sus existencias a parte de otros servicios que se podría cumplir si el cliente lo solicitara.

La pregunta es: ¿Qué necesitan los clientes finales?

Si se contesta a esta pregunta se podría destacar lo siguiente:

Al ser una empresa con ánimo de lucro necesita saber con exactitud todas las entradas y salidas del inventario para poder llevar un control exhaustivo.

Además, necesitan saber los empleados que tiene la empresa con todos los datos correspondientes para poder consultar en cualquier momento la información de sus trabajadores.

Por último, es muy importante la comunicación entre trabajador y administrador y por ello se ha hecho un canal para intercambiar mensajes.

2.1.3 IMAGEN CORPORATIVA Y NOMBRE

El nombre de la empresa ha sido elegido por la diosa Sirona, diosa de las sanaciones de origen celta adorada en el este de Galicia central y a lo largo del limes danubiano, así pues, como la actividad principal y la especialización va dirigida al sector farmacéutico se ha elegido dicho nombre.

En cuanto a la imagen corporativa se han elegido colores principales el gris oscuro y blanco.

La imagen se conforma de tres elementos:

- Circulo que abraza la marca y el dibujo
- La mano como símbolo de curación
- La marca de la empresa.



Figura 1: Logotipo de la empresa

2.2 VIABILIDAD DE MERCADO

Analizando la competencia en este sector se ha investigado vía internet las aplicaciones que hay disponibles ahora mismo en el mercado y se han encontrado diferentes plataformas, aunque no hacen exactamente lo mismo.

Consoft es un software de gestión para la oficina de farmacia, convirtiéndose en líderes del mercado en 1990.

El reconocimiento del sector farmacéutico ha permitido a Consoft el desarrollo de soluciones y servicios para este sector: Gestión de colegios, módulos de comunicaciones, desarrollos e implantación de Tarjetas Sanitarias, implementación de Recetas Electrónicas, interfaces para robótica.

Esta plataforma sería nuestra competencia más directa porque ofrece muchos servicios a las farmacias y la aplicación estaría ahora mismo en vías de desarrollo para ampliar los servicios a los clientes finales.

También existen aplicaciones gratuitas dirigidas a las farmacias:

1. **Accesible plus:** es una aplicación gratuita que permite la consulta de información actualizada sobre los medicamentos, buscando por el nombre o mediante la captura del código de barras o código datamatrix.
2. **FarmaApp:** es una aplicación que ofrece a los usuarios conocer las farmacias que hay a su alrededor y evita el desplazamiento y filas innecesarias.

Lo que nuestra aplicación quiere ofrecer a largo plazo es una combinación de todas ellas, de esa manera tendrán todo lo necesario en una misma plataforma sin la necesidad de tener varias aplicaciones.

2.3 VALORACIÓN ECONÓMICA DEL PROYECTO

La valoración económica puede variar según los requerimientos de un cliente, aunque se va a hacer una estimación sobre la demo de este proyecto:

Hay que tener en cuenta varios factores para estimar el coste, como el diseño de la web, tiempo de trabajo, funcionalidades del software y lenguaje de programación, por todo ello se ha hecho una estimación mensual según el tipo de licencia:

Administrador (otorga todos los derechos al usuario): 100€/mes

Limitada (solamente puede consultar información): 30€/mes.

2.4 VIABILIDAD ECONÓMICA FINANCIERA

La inversión inicial para este proyecto sería el siguiente:

- Coste del nombre del dominio: en este caso va a ser un dominio que principalmente es para España, por lo cual es una extensión .es (su coste viene incluido en el siguiente punto).
- Coste del hosting: este apartado es muy importante ya que no vale cualquier hosting, si no uno que te de la seguridad de que tu página web vaya a tener una velocidad óptima y de carga rápida. Hostinger es una web de hostings donde ofrece unos precios competitivos, nuestra web usaría el “*plan empresarial*”. Este plan ofrece:

<i>100 sitios web</i>	<i>200 GB de Almacenamiento SSD</i>
<i>100000 visitas Mensuales</i>	<i>Email Gratis 100 SSL</i>

<i>Acceso GIT</i>	<i>Acceso SSH y dominio</i>
<i>Copias de seguridad diarias</i>	<i>CDN Cloudflare</i>
<i>Bases de datos ilimitadas</i>	<i>Herramienta Staging de wordPress</i>

Todo esto implica un coste de 3,99€/mes lo que implica al año **47,88€**.

- Coste de certificado: aunque en el plan nos viene un certificado de seguridad SSH, interesa el SSL ya que dispone de las más fuertes.

Para ello, se usará SSL Comodo ya que es compatible con el 99,3% de los navegadores actuales y su coste es de **39,95€** al año.

- Coste de marketing: en principio en cuestión de marketing se utilizará un plan para obtener los mejores resultados y la publicidad se obtendría de ir personalmente a las farmacéuticas a hacerles una demo de la aplicación y se publicaría en redes sociales. También se ofrecería una prueba gratuita de un mes para que la prueben.

En resumen, el coste para iniciar la actividad se desglosaría:

- Gastos de alojamiento: 87,83€/año.
- Gastos varios: 260€/año (luz, internet...)

La suma total es de **439,15€/año**.

2.4 RECURSOS NECESARIOS

Los medios materiales que se va a emplear son los siguientes

Ordenador sobremesa
Ratón
Teclado
Dos pantallas
Cascos
Silla ergonómica
Local propio

Los medios inmateriales son:

Internet
Luz

Los medios de software:

Base de datos	Phpmyadmin
Entorno de desarrollo	VSC Plugins: Bracket Pair Colour Auto Close Tag Javascript Debugger JS-CSS.HTML Formatter Live server Material icon theme

	Php formatter PHP getters & setters Twig pack
Framework	Symfony 5

2.6 MARKETING

Para hacer publicidad de Sirona se usará las redes sociales principalmente para hacer promoción de la aplicación web, también se personificará en las farmacias publicitando la web ofreciendo una prueba durante un mes gratuitamente para que prueben la aplicación, además se asistirá a las convenciones del sector sanitario para dar a conocer la aplicación y poder conocer en profundidad el sector.

Para el marketing del proyecto se utilizará las siguientes herramientas y estrategias, se haría una web promocionando la aplicación donde se usaría:

- Google Adwords como posicionamiento SEM: es un servicio de Google para hacer publicidad, tiene unas ventajas muy remarcadas:

1. Aumentar las ventas online
2. Planificador de Palabras Clave
3. Inmediatez (a través del posicionamiento SEO)
4. Tráfico cualificado (aumentar el número de visitas)

- Posicionamiento SEO: para que la web salga entre las primeras.

CAPÍTULO 3: ANÁLISIS TÉCNICO DE LOS ENTORNOS

SERVIDOR Y CLIENTE

3.1 CASOS DE USO

El usuario solamente tiene acceso a la lectura de los datos a través del perfil donde se obtienen los datos del usuario logueado y a la hora de la búsqueda de medicamentos donde se muestra un listado con el contenido buscado.

También podrá enviar mensajes vía email a través de la aplicación web.

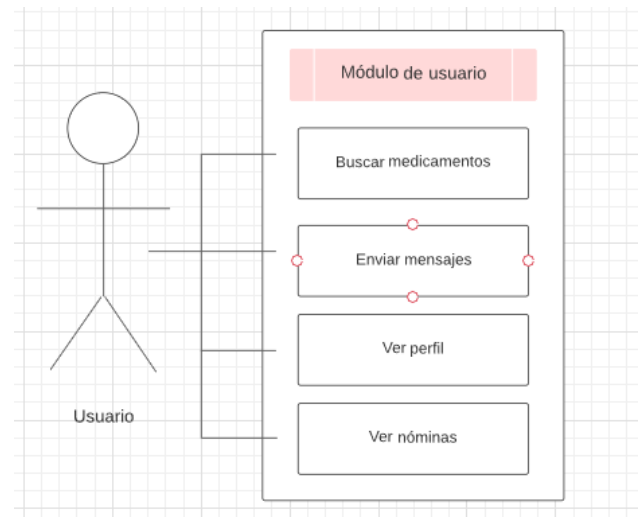


Figura 2: Caso de uso del usuario

El administrador tiene todos los privilegios:

- Búsqueda de medicamentos
- Ver, editar y eliminar usuarios
- Ver, editar y eliminar proveedores
- Ver, editar y eliminar medicamentos
- Agregar nóminas

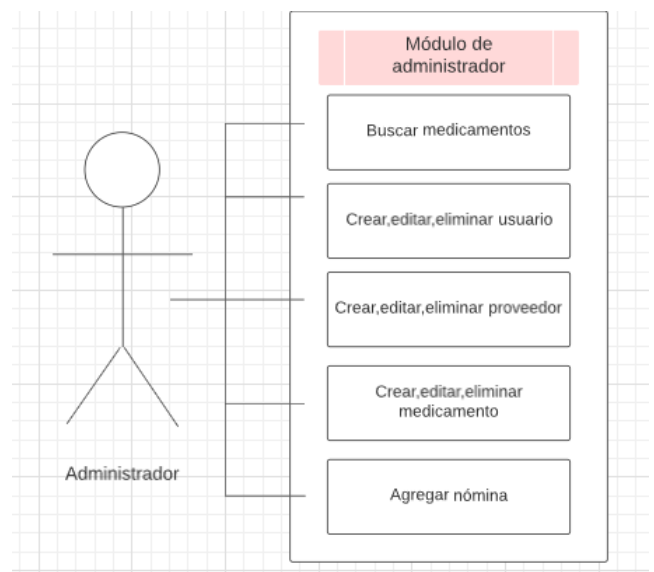


Figura 3: Caso de uso del administrador

3.2 DESCRIPCIÓN DE LA ESTRUCTURA DEL PROYECTO

La aplicación esta realizada con el framework de Symfony, este framework es un entorno de trabajo que se utiliza para el desarrollo de aplicaciones web utilizando PHP como lenguaje de programación.

Symfony está basado en el patrón modelo vista controlador que separa la lógica de la presentación, la lógica del servidor y la lógica de negocio. Además, este framework nos permite la escalabilidad, esto quiere decir, que una vez creada se puede ampliar y crecer a pesar de tener mucho código.

Por todos estos motivos, la idea de la estructura de la web se adapta plenamente en lo que Symfony ofrece a los programadores.

Hay que hacer mención a las plantillas de symfony, Twig, es un motor de plantillas desarrollado para el lenguaje de programación PHP y que nace con el objetivo de facilitar a los desarrolladores de aplicaciones web que

utilizan la arquitectura MVC el trabajo con la parte de las vistas, gracias a que se trata de un sistema que

resulta muy sencillo de aprender y capaz de generar plantillas con un código preciso y fácil de leer.

Actualmente el código se distribuye bajo licencia BSD y es utilizado por el framework Symfony, aunque

puede ser utilizado directamente con proyectos desarrollados en PHP en el que no interviene ese framework.

Como base de datos se ha utilizado **MySQL Database**, MySQL es un componente importante de una pila empresarial de código abierto llamada LAMP. LAMP es una plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de bases de datos relacionales y PHP como lenguaje de scripting orientado a objetos.

También se ha usado **Xampp** ya que es una herramienta de desarrollo que permite probar el desarrollo web basado en PHP en local sin necesidad de tener acceso a internet, además se instala tanto mysql como apache (servidor).

Además, se ha usado **javascript** para añadir acciones interactivas con el cliente, este lenguaje es muy apropiado para todo ello ya que permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás.

Para dar estilo a los templates se ha usado **css** como lenguaje principal ya que se puede moldear la vista fácilmente.

Por último, se ha usado **Jquery**, es una librería que permite añadir una capa de interacción AJAX entre la web y las aplicaciones que se desarrolla controlando eventos, creando animaciones y diferentes efectos para enriquecer la experiencia de usuario.

Para la subida se ha trabajado con Github ya que es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador. La plataforma está creada para **que** los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargar la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.

3.2.1 METODOLOGIA AGIL SCRUM

En este proyecto se ha elegido método ágil “Scrum” por dos razones, por un lado, es que en las prácticas los proyectos se realizan con esta metodología.

Por otro lado, el framework se basa en la idea de la mejora continua a través de un enfoque orientado al proceso. Los problemas se pueden resolver más

eficientemente si se dividen en piezas más pequeñas editando con los problemas o no que han surgido.

Se ha estructurado el proyecto de la siguiente manera:

En las primeras semanas se dedican a investigación y análisis del mercado.

▼ Semana 1

Buscar información
Análisis de objetivos y funcionalidades
Diseño de la web y logotipo

▼ Semana 2

Elección de tecnologías y lenguajes
Elección de frameworks
Creación del ERD

▼ Semana 3

Creación del proyecto con symfony
Migraciones para crear las entidades en la base de datos
Estudiar las casuísticas

▼ Semana 4

Elección de tipos de rol
Creación de login (dar estilo)
Creación del home del usuario

A partir de la semana 3 se crea el proyecto en Symfony configurando el "env" para conectar el proyecto con la base de datos, asimismo se integran las tablas en la base de datos creando en el proyecto simultáneamente las entidades con las que se van a trabajar con el siguiente comando:

```
php bin/console make:entity
```

Además, se crea el login con seguridad integrada validando la entrada del usuario, en esta validación se comprueba el tipo de rol que tiene cada usuario y verifica si el usuario existe en la base de datos, si no existe redirecciona al login inicial y si existe dependiendo del rol redirige a un home distinto.

```
public function onAuthenticationSuccess(Request $request, TokenInterface $token, string $firewallName): ?Response
{
    if ($targetPath = $this->getTargetPath($request->getSession(), $firewallName)) {
        return new RedirectResponse($targetPath);
    }

    if($token->getUser()->getRoles()[0] === 'ROLE_USER'){
        return new RedirectResponse($this->urlGenerator->generate('user'));
    }elseif($token->getUser()->getRoles()[0] === 'ROLE_ADMIN'){
        return new RedirectResponse($this->urlGenerator->generate('admin'));
    }
}
```

Figura 4: Redirección al home por el rol asignado

▼ Semana 5

Creación del navbar (dar estilos)
Creación del buscador de medicamentos

▼ Semana 6

Creación del Footer (dar estilo)
Diseñar home (dar estilo)
Creación enviar mensajes

En estas semanas se crea el buscador de medicamentos, se realiza con código javascript mediante una petición ajax al back mediante un evento de keyup, cuando se presiona una tecla se hace una petición al back y devuelve una lista de medicamentos que contenga el valor enviado buscado por nombre de medicamento.

```
function ajax_post() {  
  $("#input_text").keyup(function(){  
    let search = $("#input_text").val();  
    $.post('/user/ajax_post', {info:search}, function(data){  
      $('p').remove();  
      $('table').remove();  
      let table = document.createElement('table');  
      table.setAttribute('class', 'table');  
      data.forEach(task => {  
        if($("#input_text").val() != ''){  
          let tr = document.createElement('tr');  
          for(let i = 0; i < task.length; i++){  
            let td = document.createElement('td');  
            td.innerHTML = task[i];  
            tr.appendChild(td);  
          }  
          table.appendChild(tr);  
          document.getElementById('result').appendChild(table);  
          $('#result').append(content);  
        }  
      });  
    });  
  });  
}
```

Figura 5: Petición a back para filtrar los medicamentos que contengan el valor enviado

También se realizó la sección enviar mensajes, esta sección se hizo con una librería que se llama “Google mailer”, al instalarlo se añade al .env unos parámetros para poder validar los mensajes, vía email, así de esta manera se enviarán los mensajes a un correo validado por google, si no está validado no llegarán los emails.

```

/**
 * @Route("/user/messages", name="user_messages")
 */
public function sendMessage(Request $request, MailerInterface $mailer): Response{

    $form = $this->createForm(SendEmailType::class);

    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $email = (new Email())
            ->from('sirona.rmp@gmail.com')
            ->to('sirona.rmp@gmail.com')
            ->subject($form->getData()['name'])
            ->html('<p>'.$form->getData()['text'].'</p>');

        $mailer->send($email);

        return $this->render('user/main.html.twig', [
            'controller_name' => 'UserController',
            'user' => $this->getUser(),
        ]);
    }

    return $this->render('user/messages.html.twig', [
        'form' => $form->createView(),
    ]);
}

```

Figura 6: Envío de emails a través de google mailer

▼ Semana 7

Creación de la vista de perfil (sin foto)
Creación de la vista de nóminas (sin archivo)

▼ Semana 8

Comprobación de errores del usuario
Mejoras en la vista

En estas semanas lo que se hizo fue traer los datos del usuario a través de la petición get para mostrar al usuario los datos del perfil, además se crea un apartado para mostrar las nóminas que tiene guardadas el usuario.

▼ Semana 9

Creación del home del admin
Creación del filtro

▼ Semana 10

CRUD usuarios, proveedores y medicamentos
Comprobación de errores
Dar estilos

En estas dos semanas se comienza con el perfil del administrador, se crea el home igual que el del usuario y se trae las peticiones del filtrado de medicamentos por nombre.

Se crea todo lo que pertenece a los trabajadores, petición get para obtener el listado de trabajadores, petición put con el id del usuario para eliminarlo y petición post con el id del usuario para editarlo.

```
/**
 * @Route("/admin/new_worker", name="add_worker", methods={"GET", "POST"})
 */
public function new(Request $request, UserRepository $userRepository, UserPasswordEncoderInterface $upi): Response
{
    $user = new User();
    $form = $this->createForm(UserType::class, $user);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        //busca si el usuario existe en la base de datos
        $userExist = $userRepository->findOneBy(['email' => $user->getEmail()]);
        if($userExist){
            $this->addFlash('error', 'El usuario ya existe');
            return $this->redirectToRoute('add_worker');
        }else{
            $brochureFile = $form['photo']->getData();
            if ($brochureFile) {
                $originalFilename = pathinfo($brochureFile->getClientOriginalName(), PATHINFO_FILENAME);
                $safeFilename = iconv('UTF-8', 'ASCII//TRANSLIT', $originalFilename);
                $newFilename = $safeFilename.'-'.uniqid().'.'.$brochureFile->guessExtension();

                try {
                    $brochureFile->move(
                        $this->getParameter('files_directory'),
                        $newFilename
                    );
                } catch (FileException $e) {
                    throw new \Exception('Error al subir el archivo');
                }
            }
            $user->setPhoto($newFilename);
            $user->setPassword($upi->encodePassword($user, $form['password']->getData()));
            $userRepository->add($user);

            return $this->redirectToRoute('show_workers', [], Response::HTTP_SEE_OTHER);
        }
    }
}
```

Figura 7: Creación de un nuevo trabajador

De igual manera se realizan las peticiones de medicamentos y proveedores.

📌 Semana 12

Creación de nueva sección (nóminas)

Comprobación de errores

user/ agregar foto de perfil

Se comprueban todos los errores que puede generar los crud como campos vacíos, fechas futuras, larguras de campo, campos nulos entre otros y se capturan para mostrarlos al usuario.

```
/**
 * @Route("/admin/add_payroll", name="add_payroll")
 */
public function add_payroll(Request $request, EntityManagerInterface $em): Response
{
    $payroll = new Payroll();
    $form = $this->createForm(PayrollType::class, $payroll);
    $form->handleRequest($request);
    if ($form->isSubmitted() && $form->isValid()) {
        $brochureFile = $form['file']->getData();
        if ($brochureFile) {
            $originalFilename = pathinfo($brochureFile->getClientOriginalName(), PATHINFO_FILENAME);
            $safeFilename = iconv('UTF-8', 'ASCII//TRANSLIT', $originalFilename);
            $newFilename = $safeFilename.'-'.uniqid().'.'.$brochureFile->guessExtension();

            try {
                $brochureFile->move(
                    $this->getParameter('files_directory'),
                    $newFilename
                );
            } catch (FileNotFoundException $e) {
                throw new \Exception('Error al subir el archivo');
            }
            $payroll->setFile($newFilename);
        }
        //buscar el id del usuario seleccionado
        $user = $em->getRepository(User::class)->find($form['Trabajador']->getData());
        $payroll->setUser($user);
        $em->persist($payroll);
        $em->flush();
        return $this->redirectToRoute('add_payroll');
    }
    return $this->render('admin/add_payroll.html.twig', [
        'form' => $form->createView(),
    ]);
}
```

Figura 8: Agregar una nómina a un usuario determinado

Se genera nueva sección para agregar nóminas a un usuario con una petición post donde se le pasa el id del usuario y todos los campos para que se guarde en la base de datos.

3.3 ESPECIFICACIÓN DEL MODELO DE DATOS

3.3.1 TABLAS EN BASE DE DATOS

1	id	int(11)		No	Ninguna	AUTO_INCREMENT
2	email	varchar(50)	utf8mb4_unicode_ci	No	Ninguna	
3	roles	longtext	utf8mb4_unicode_ci	No	Ninguna	(DC2Type:json)
4	password	varchar(255)	utf8mb4_unicode_ci	No	Ninguna	
5	name	varchar(20)	utf8mb4_unicode_ci	No	Ninguna	
6	surname	varchar(50)	utf8mb4_unicode_ci	No	Ninguna	
7	adress	varchar(50)	utf8mb4_unicode_ci	Sí	NULL	
8	phone	varchar(20)	utf8mb4_unicode_ci	No	Ninguna	

Figura 9: Estructura base de datos usuario

Primary key: es el id, el cual se incrementa automáticamente y no se puede repetir.

Varchar: email, password, name, surname, adress y phone son de tipo string ya que los datos que vamos a recibir son textos. El password tiene una longitud de 255 ya que la contraseña estará encriptada y por ese motivo se recibirá muchos caracteres. La dirección puede ser null por si no se precisa de esta información.

Longtext: el rol se recibe como un json, en este caso solo podremos recibir ["ROLE_USER"] o ["ROLE_ADMIN"].

1	id	int(11)		No	Ninguna	AUTO_INCREMENT
2	user_id	int(11)		Sí	NULL	
3	date	date		No	Ninguna	
4	salary	double		No	Ninguna	
5	file	varchar(500)	utf8mb4_unicode_ci	Sí	NULL	

Figura10: Estructura base de datos nómina

Primary key: es el id, el cual se incrementa automáticamente y no se puede repetir.

Date: se recoge la fecha solamente con YYYY/MM/DD.

Double: el salario contiene decimales.

Varchar: el file es el documento que se adjunta como justificante de nómina, se ha puesto con esa largura porque se guarda el nombre del archivo. Puede ser nulo porque un usuario puede tener o no nóminas (se contempla el caso de haber entrado recientemente en la empresa).

Int: se relaciona con el id del usuario este también puede ser null por el mismo caso del anterior.


1	id 	int(11)	No	Ninguna	AUTO_INCREMENT
2	cif	varchar(30) utf8mb4_unicode_ci	No	Ninguna	
3	name_enterprise	varchar(30) utf8mb4_unicode_ci	No	Ninguna	
4	person_contact	varchar(20) utf8mb4_unicode_ci	No	Ninguna	
5	email	varchar(30) utf8mb4_unicode_ci	No	Ninguna	
6	adress	varchar(50) utf8mb4_unicode_ci	No	Ninguna	
7	phone	varchar(30) utf8mb4_unicode_ci	No	Ninguna	

Figura 11: Estructura base de datos proveedor

Primary key: es el id, el cual se incrementa automáticamente y no se puede repetir.

Varchar: todos los atributos son de tipo texto y ninguno de ellos puede ser nulo.

1	id 	int(11)	No	Ninguna	AUTO_INCREMENT
2	description	varchar(20) utf8mb4_unicode_ci	No	Ninguna	

Figura 12: Estructura base de datos proveedor

Primary key: es el id, el cual se incrementa automáticamente y no se puede repetir.

Varchar: todos los atributos son de tipo texto y ninguno de ellos puede ser nulo.

1	id	int(11)	No	Ninguna	AUTO_INCREMENT
2	provider_id	int(11)	Sí	NULL	
3	name	varchar(20) utf8mb4_unicode_ci	No	Ninguna	
4	reference	int(11)	Sí	NULL	
5	quantity	int(11)	No	Ninguna	
6	description	varchar(100) utf8mb4_unicode_ci	Sí	NULL	
7	price	double	No	Ninguna	

Figura 13: Estructura base de datos producto

Primary key: es el id, el cual se incrementa automáticamente y no se puede repetir.

Varchar: el nombre y la descripción son de tipo texto a diferencia de que la descripción puede ser nula por si no necesita ese producto una descripción.

Int: la referencia puede ser nula por si el producto no tiene una referencia, la cantidad no puede ser nula porque es necesario para el control de las existencias y el id del proveedor también es nula por si ese producto no pertenece a un proveedor en concreto.

Double: el precio contiene decimales.

3.3.2 DIAGRAMA ENTIDAD-RELACION

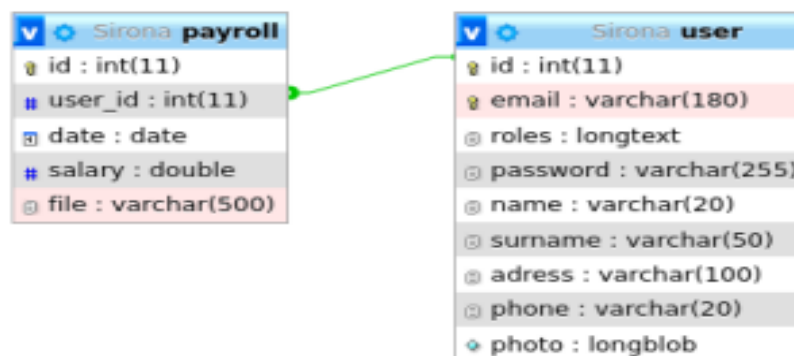


Figura 14: Estructura base de datos nómina/usuario

Las tablas están relacionadas con la primary key de usuario, un usuario puede tener muchas nóminas, pero una nómina no puede pertenecer a varios usuarios.

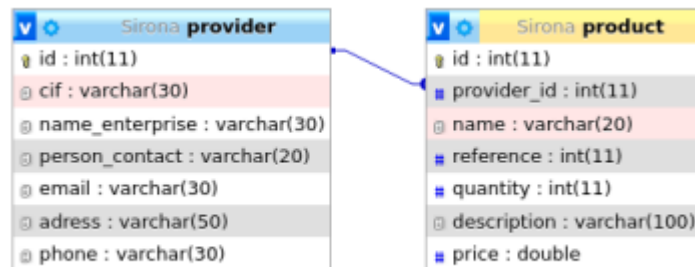


Figura 15: Estructura base de datos proveedor/producto

Las tablas están relacionadas con la primary key del proveedor ya que un producto puede ser de un proveedor, pero un proveedor puede tener varios productos.

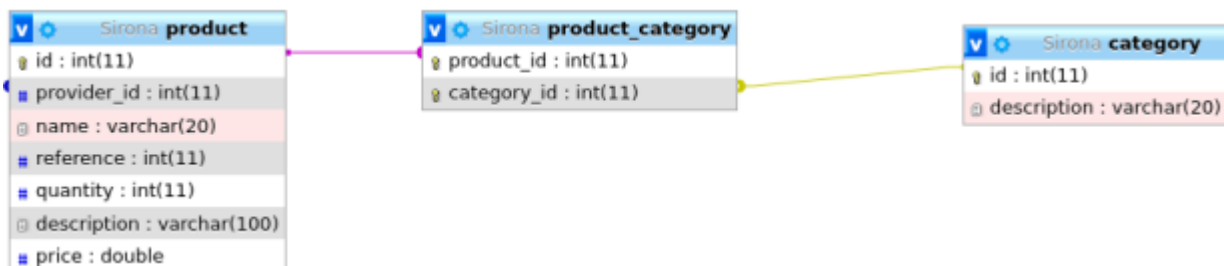
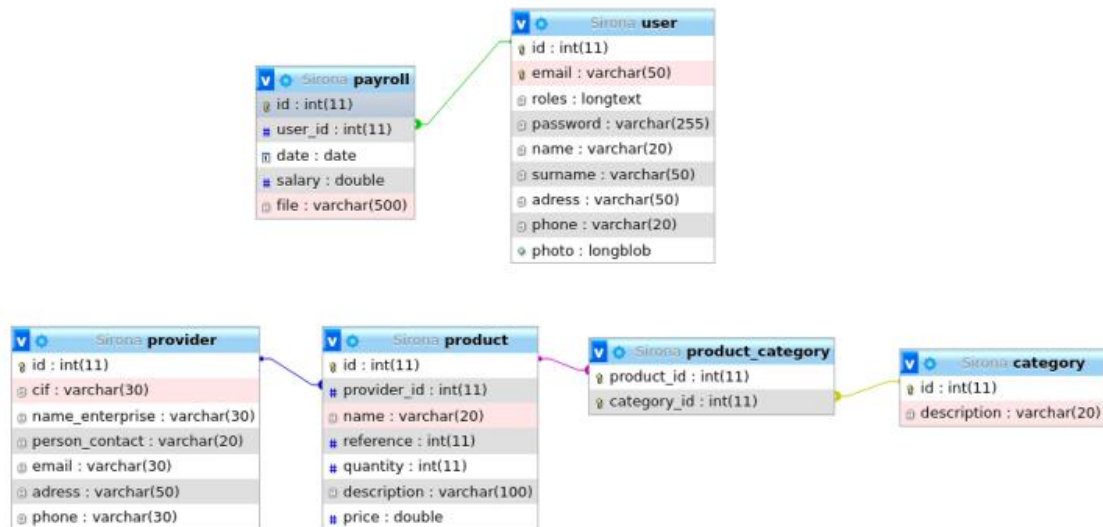


Figura 16: Estructura base de datos categoría/producto

En este caso se ha generado una tabla de muchos a muchos relacionándose las primary key de cada una de las tablas ya que un producto puede tener varias categorías y una categoría puede tener varios productos.



De esta manera se quedaría la base de datos con todas las relaciones entre tablas.

CAPÍTULO 4: CONCLUSIONES Y POSIBLES ACTUALIZACIONES

En este capítulo se analizará las conclusiones a las que se ha llegado al realizar el proyecto y el futuro del proyecto para las posibles mejoras que podría poder llegar a tener.

4.1 CONCLUSIÓN

Gracias al desarrollo del proyecto se ha logrado alcanzar que el código sea fácil de mantener y cómodo de modificar para posibles futuras mejoras ya que está basado en diferentes controladores y es visiblemente cómodo a la hora de navegar por el proyecto.

Además, se ha utilizado herramientas para la gestión del proyecto que ha ayudado a gestionar el proyecto de una manera ágil facilitando el propio desarrollo con la metodología scrum.

4.2 POSIBLES ACTUALIZACIONES

A pesar de que los objetivos se han cumplido han quedado ideas para integrar en el proyecto y mejorar la experiencia del usuario.

- Localización de farmacias cercanas
- Foro de farmacias que usen la misma aplicación
- Incluir integración continua con un sistema de testing unitario mejorado
- Mejorar el filtrado para que se pueda buscar por más campos
- Poder editar las fotografías y añadir más de una

CAPÍTULO 5: BIBLIOGRAFÍA

el Economista. (2021, 30 abril). *La industria farmacéutica como motor económico*.

elEconomista.es. <https://www.eleconomista.es/salud-innovacion/noticias/11171603/04/21/La-industria-farmaceutica-como-motor-economico.html>

B., G. (2022, 3 mayo). *¿Cuál es el precio de crear una página web en 2022?* Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/precio-pagina-web#:~:text=En%20resumen%2C%20crear%20un%20sitio,los%201000%20euros%20por%20a%C3%B1o.>

B., G. (2022, 3 mayo). *¿Cuál es el precio de crear una página web en 2022?* Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/precio-pagina-web#:~:text=En%20resumen%2C%20crear%20un%20sitio,los%201000%20euros%20por%20a%C3%B1o.>

Ribas, E. (2018, 8 noviembre). *Las mejores herramientas y aplicaciones para el sector Farmacéutico*. Thinking for Innovation. <https://www.iebschool.com/blog/mejores-herramientas-aplicaciones-sector-farmaceutico-marketing-digital/>

Documentación sobre Symfony. (2022). symfony.es. Recuperado 2 de junio de 2022, <https://symfony.es/documentacion/>

JavaScript | MDN. (2022, 30 mayo). Javascript. Recuperado 2022, de <https://developer.mozilla.org/es/docs/Web/JavaScript>

CSS | MDN. (2021, 7 julio). Css. <https://developer.mozilla.org/es/docs/Web/CSS>

Home - Twig - The flexible, fast, and secure PHP template engine. (2022). Twig. Recuperado 2 de junio de 2022, de <https://twig.symfony.com/>

colaboradores de Wikipedia. (2022, 6 mayo). *Industria farmacéutica*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Industria_farmac%C3%A9utica

JS Foundation - js.foundation. (2022). *jQuery*. Jquery. Recuperado 2 de junio de 2022, de <https://jquery.com/>

Verificación humana. (2022). Zack Overflow en español. Recuperado 2 de junio de 2022, de <https://es.stackoverflow.com/search?q=symfony>

Fernández, R. (2022, 14 marzo). *La industria farmacéutica en España - Datos estadísticos*. Statista. <https://es.statista.com/temas/5603/la-industria-farmaceutica-en-espana/>

INE - Instituto Nacional de Estadística. (2022). *INE / La organización estadística en España / Inventario de Operaciones Estadísticas / Inventario actual / Ficha IOE actual: 54042*. INE. Recuperado 2 de junio de 2022, de <https://www.ine.es/dyngs/IOE/es/operacion.htm?numinv=54042>

Lucid visual collaboration suite: Log in. (2022). Lucid. Recuperado 2 de junio de 2022, de <https://lucid.app/users/login>

Eguiluz, J., & Eguiluz, J. (2007, 11 septiembre). *El framework Symfony, una introducción práctica (I parte)*. Maestros del Web. <http://www.maestrosdelweb.com/el-framework-symfony-una-introduccion-practica-i-parte/>

monday.com. (2022). *monday.com / Una nueva forma de trabajar*. Recuperado 2 de junio de 2022, de https://monday.com/lang/es/?utm_source=adwordsbrand&utm_campaign=latam_es-s-brand-b-desk-monday&utm_language=es&aw_keyword=%2Bmonday&aw_match_type=b&gclid=CjwKCAjwv-GUBhAzEiwASUMm4sx_RYWODJ9VUppzAKfQ8DyIRj9nNX6IMBqkWmj81OjaxtdLqtyJXhoCf1gQAvD_BwE