

# **DRAWTATOO**

CFGS DAW

CURSO 2023/2024

JAVIER CASTILLA GARRIDO

JORGE GARCÍA ULLÁN

COLEGIO CALASANZ SALAMANCA

1. Introducción .....	3
2. Descripción de la aplicación. ....	3
3. Plan de empresa .....	4
4. Tecnologías escogidas y justificación .....	5
5. Diseño de la aplicación .....	6
5.1. Diagramas y definición de casos de uso.....	6
5.2. Diagramas de clase .....	9
5.3. Modelo entidad relación.....	10
6. Arquitectura de la aplicación .....	11
6.1 Estructura del proyecto .....	11
6.2. Librerías externas utilizadas .....	11
7. Manual de despliegue .....	12

## 1. Introducción

Nuestro grupo ha pensado en realizar una aplicación web para la gestión del negocio de una tienda de tatuajes. En ella vamos a realizar una presentación de la empresa “DRAWTATOO”, de sus trabajos anteriores y también un sistema de registro de citas, apta tanto para usuarios registrados como para los no registrados, que dará la información correspondiente al usuario sobre el tatuador seleccionado, el tipo de tatuaje, un margen de precio... todo acompañado de una galería de imágenes que ilustraran las posibles opciones a seleccionar.

## 2. Descripción de la aplicación.

La aplicación se iniciará en una página principal que constará de tres partes: En primer lugar, una cabecera compuesta de un menú para la navegabilidad por la página web. En segundo lugar, tendremos el cuerpo de la página, donde se mostrarán las distintas secciones que se ofrecen, como la información de los tatuadores, una galería, etc. Para terminar, tendremos un pie de página que incluirá la información típica en el desarrollo de este tipo de páginas y los enlaces a las redes sociales. Tanto la cabecera como el pie de página serán constantes dentro del diseño de las páginas sucesivas. Dentro del menú tendremos una sección de inicio (que nos devolverá a la página principal), una de tatuadores (información sobre cada uno de nuestros artistas y una explicación sobre su especialidad), una galería general de los últimos trabajos realizados en la tienda, una página de información sobre la empresa que ofrecerá la solicitud de citas y un formulario de login/registro. La página sobre nuestros tatuadores ofrece la posibilidad de realizar una cita con el profesional escogido, mientras que el formulario de solicitud de cita de la página “Sobre nosotros” no indicará al tatuador en primera instancia. Ambos formularios tienen la funcionalidad de darle a nuestros clientes el costo del tatuaje basándose en su tamaño y si se realizará con tinta de color o negra. Si una persona no registrada quiere solicitar una cita, el formulario de solicitud le requerirá una contraseña, ya que también funcionará como formulario de registro. Todo usuario registrado tendrá la posibilidad de acceder a una zona personal, estando logeado en primer lugar (una vez se logue es necesario recargar la página), donde podrá comprobar sus citas pasadas y futuras y donde podrá editar las futuras citas si fuese necesario.

Para la parte de la gestión de la tienda se generará un PDF de todas las citas, clientes y tatuajes que tengamos registrados en nuestra base de datos. Estos se actualizarán automáticamente al añadir cualquier cambio, es decir, si un cliente nuevo se registra en la aplicación se actualizará el pdf de clientes, o si un cliente registrado edita una de sus citas estos cambios se verán reflejados en el pdf de citas si edita algún dato relacionado con nuestra tabla de citas y el pdf de tatuajes si modifica cualquier dato del tatuaje que hubiera seleccionado en primera instancia.

### 3. Plan de empresa

Al tratarse de la digitalización de un negocio físico y existiendo la posibilidad de obtener una subvención para la realización de este proyecto, nos vamos a referir a la Orden ETD/1498/2021, de 29 de diciembre, por la que se aprueban las bases reguladoras de la concesión de ayudas para la digitalización de pequeñas empresas, microempresas y personas en situación de autoempleo, en el marco de la Agenda España Digital 2025, el Plan de Digitalización PYMEs 2021-2025 y el Plan de Recuperación, Transformación y Resiliencia de España -Financiado por la Unión Europea- Next Generation EU (Programa Kit Digital). En ella se nos da una tabla presupuestaria de muestra que estimaría que, para un proyecto de cuatro meses incluyendo desarrollo y mantenimiento, el presupuesto oscilaría entre los 6000 y los 9000 euros.

Segmentos de beneficiarios	Importe del «bono digital»
Segmento I. Pequeñas empresas entre 10 menos de 50 empleados.	12.000 euros
Segmento II. Pequeñas empresas o Microempresas entre 3 y menos de 10 empleados.	6.000 euros
Segmento III. Pequeñas empresas o Microempresas de entre 0 y menos de 3 empleados.	2.000 euros

Asimismo, los importes máximos de ayuda por Categoría de Soluciones de Digitalización y segmento son los siguientes:

Categorías de Soluciones de Digitalización	Meses de prestación del servicio	Ayudas por segmento para las Categorías de Soluciones de Digitalización		
		Segmento III 0 -<3 empleados	Segmento II 3 -<10 empleados	Segmento I 10 -< 50 empleados
Sitio Web y Presencia en Internet.	12	2.000 €	2.000 €	2.000 €
Comercio Electrónico.	12	2.000 €	2.000 €	2.000 €
Gestión de Redes Sociales.	12	2.000 €	2.500 €	2.500 €
Gestión de Clientes.	12	2.000 € (incluye 1 usuario)	2.000 € (incluye 1 usuario)	4.000 € (incluye 3 usuarios)
Business Intelligence y Analítica.	12	1.500 € (incluye 1 usuario)	2.000 € (incluye 1 usuario)	4.000 € (incluye 3 usuarios)
Gestión de Procesos.	12	500 € (incluye 1 usuario)	2.000 € (incluye 3 usuarios)	6.000 € (incluye 10 usuarios)
Factura Electrónica.	12	500 € (incluye 1 usuario)	1.000 € (incluye 3 usuarios)	1.000 € (incluye 3 usuarios)
Servicios y herramientas de Oficina Virtual.	12	250 € / usuario		
		(hasta 2 usuarios)	(hasta 9 usuarios)	(hasta 48 usuarios)
Comunicaciones Seguras.	12	125 € / usuario		
		(hasta 2 usuarios)	(hasta 9 usuarios)	(hasta 48 usuarios)
Ciberseguridad.	12	125 € / dispositivo		
		(hasta 2 dispositivos)	(hasta 9 dispositivos)	(hasta 48 dispositivos)

A continuación, vamos a mostrar un cuadro con el desglose del presupuesto para cuatro meses más una aproximación del mantenimiento a lo largo del tiempo.

DRAWTATOO	MARZO	ABRIL	MAYO	JUNIO	TOTAL
<b>LICENCIAS</b>					
Office 365 Empresa Estándar (2)	23,4	23,4	23,4	23,4	93,6
Windows 11 Pro (2)	80	80	80	80	320
<b>SERVIDOR WEB</b>					
Ionos Essential	8,47	8,47	8,47	8,47	33,88
<b>EQUIPOS</b>					
Lenovo IdeaPad 3 (2)	1.797,98				1.797,98
<b>SALARIOS</b>					
Programador Junior (2)	2400	2400	2400	2400	9600
<b>LOGOTIPO</b>					
Artista Freelance	120				120
<b>OFICINA</b>					
Avenida de Portugal, Salamanca	110	110	110	110	440
<b>INTERNET</b>					
Movistar Empresas	24,9	24,9	24,9	24,9	99,6
<b>PUBLICIDAD</b>					
Publicidad digital	150	150	150	150	600
					13.105,06
<b>MANTENIMIENTO</b>					+
En Euros/mes					35€/mes

Como se puede observar, el proyecto requeriría de una inversión inicial ya que la subvención no cubriría todos los gastos incurridos en este caso, tratándose de contratos normalizados (no de prácticas o becados) y con gastos iniciales de empresa como son la adquisición de ordenadores y licencias.

## 4. Tecnologías escogidas y justificación

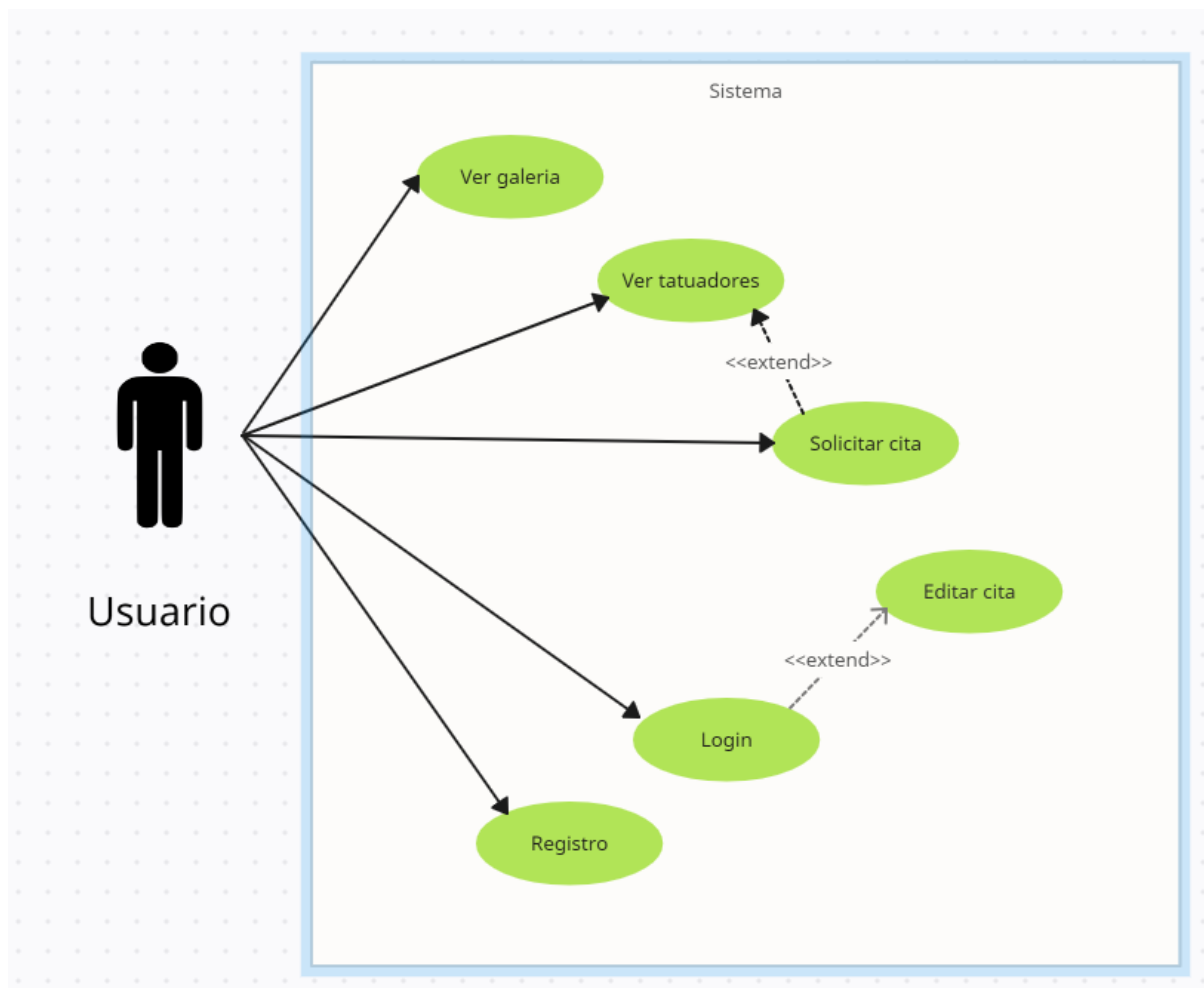
A la hora de plantearse la realización de este proyecto se realizan las siguientes elecciones:

- En cuanto a la elaboración de la base de datos se va a utilizar MySQL, ya que es el lenguaje que hemos aprendido y conocemos. Por el mismo motivo se ha utilizado MySQL Workbench 8.0 para el trabajo con esta.
- En cuanto a la parte de front de la aplicación hemos elegido trabajar con VUE Router por la facilidad que ofrece a la hora de elaborar las distintas vistas que se quieren mostrar en la página web. Esto se ha realizado desde Visual Studio Code. En esta parte, específicamente en la galería de imágenes, decidimos alojarlas en la base de datos para no tener que almacenarlas en la carpeta assets del proyecto. Sin embargo, debido a la longitud de las cadenas de texto para imágenes en base64 se nos hacía difícil incluir imágenes de mayor calidad, por lo que la galería mostrará dos imágenes que tenemos alojadas en base64 en la base de datos, mientras que el resto estarán en la carpeta anteriormente mencionada.

- Finalmente, en cuanto a la parte de back hemos escogido utilizar Java, específicamente su framework Spring, debido a nuestros conocimientos acerca de ella y la mayor facilidad para su desarrollo, desde la herramienta Eclipse IDE. Con ello se ha realizado una API Rest para realizar el contacto entre front y back, al ser una forma sencilla de unir ambas partes.
- Para la elaboración de los elementos de UML se ha utilizado creately.com, que en su versión gratuita ha sido suficiente para la realización de estos, y canva.com por su facilidad de uso.

## 5. Diseño de la aplicación

### 5.1. Diagramas y definición de casos de uso



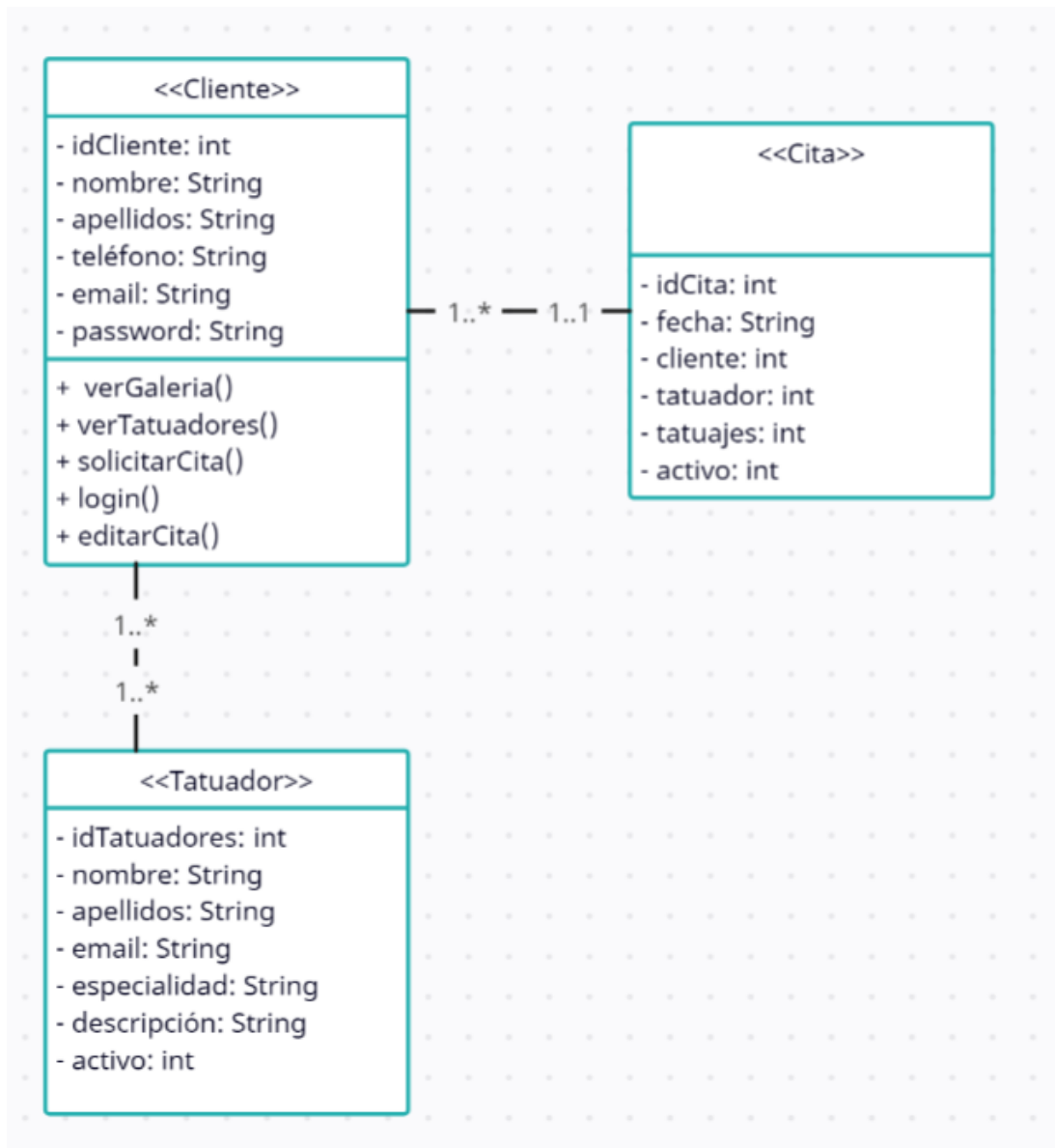
## Casos de uso generales

Caso de uso	Registrar al usuario
Objetivo	Dar de alta al nuevo usuario
Actores	Usuario
Precondiciones	El usuario no tiene cuenta en el sistema
Postcondiciones	Datos del nuevo usuario introducidos en la BBDD
Pasos o Flujo básico	1.A El usuario solicita darse de alta. 2.S Se muestra el formulario para introducir un nuevo usuario. 3.A El usuario introduce los datos del nuevo perfil. (Nombre, apellidos, teléfono, correo electrónico y contraseña). 4.S Se da de alta un nuevo usuario. 5.S Se muestra un mensaje de éxito.
Variaciones	4.a El correo ya existe en la BBDD 1.S. El sistema muestra un mensaje “usuario ya existe” 2.S Se vuelve al paso 2. 4.b El correo no tiene un formato valido. 1.S. El sistema muestra un mensaje “correo no valido”. 2.S. Se vuelve al paso 2. 4.c No ha seleccionado el tipo de usuario 1.S El sistema muestra un mensaje “seleccione el tipo de usuario a registrar”. 2.S. Se vuelve al paso 2. 4.d Alguno de los campos del formulario está vacío 1.S El sistema muestra un mensaje “rellene el formulario completo”. 2.S Se vuelve al paso 2.

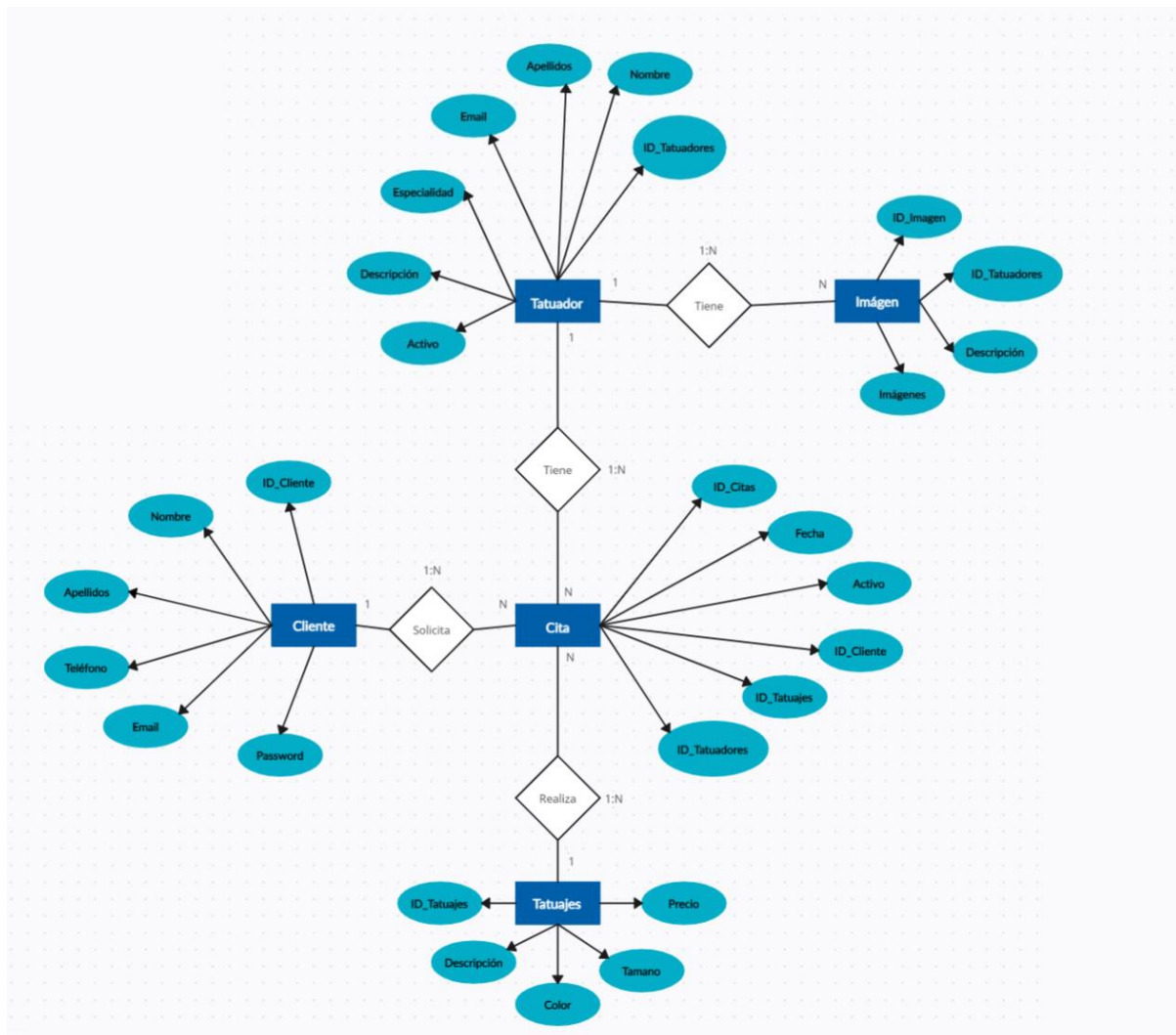
Caso de uso	Iniciar sesión
Objetivo	Un usuario registrado accede al sistema
Actores	Usuario
Precondiciones	El usuario se ha registrado
Postcondiciones	Usuario accede al sistema
Pasos o Flujo básico	1.A El usuario solicita el inicio de sesión 2.S Se muestra el formulario para introducir usuario y contraseña. 3.A El usuario introduce los datos de usuario y contraseña 4.S Se valida contra la BD. 5.S Se inicia la sesión. 6.S Se accede a la aplicación.
Variaciones	4.a El campo usuario no existe en la BD 1.S. El sistema muestra un mensaje “Credenciales incorrectas”. 2.S Se vuelve al paso 2. 4.b La contraseña no coincide con el usuario 1.S. El sistema muestra un mensaje “Credenciales incorrectas”. 2.S. Se vuelve al paso 2. 4.c Alguno de los campos del formulario está vacío 1.S El sistema muestra un mensaje “rellene el formulario completo”. 2.S Se vuelve al paso 2. 4.d El usuario no se ha validado 1.S El sistema muestra un mensaje “valide su cuenta antes de continuar”. 2.S Se vuelve al paso 2.



## 5.2. Diagramas de clase



### 5.3. Modelo entidad relación



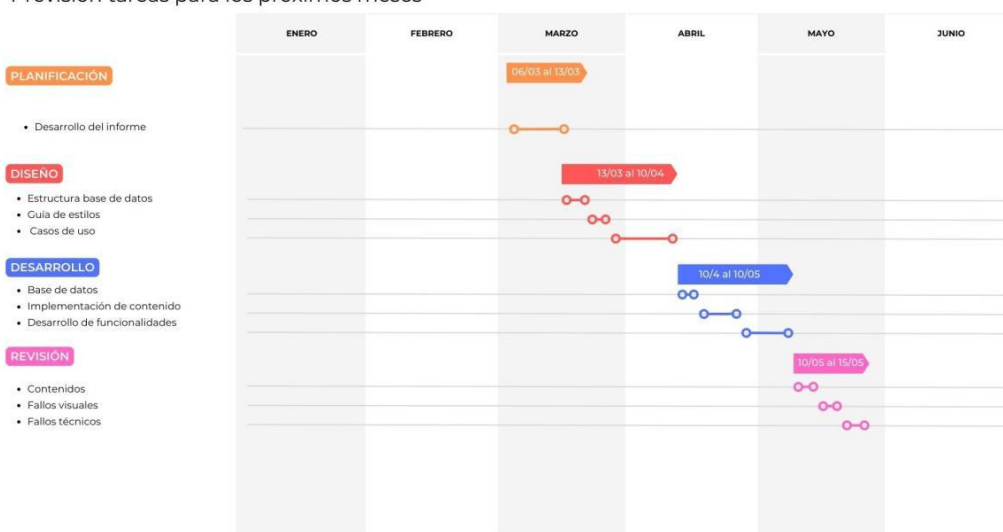
## 6. Arquitectura de la aplicación

### 6.1 Estructura del proyecto

La planificación se hará siguiendo un planteamiento con un diagrama de Gantt, ya que son una herramienta muy útil para la gestión de proyectos porque permiten planificar, programar y monitorear el trabajo de manera efectiva, lo que ayuda a garantizar la entrega de los proyectos a tiempo, si no surgen imprevistos, así como detectar tareas críticas y posibles cuellos de botella que frenan el trabajo

#### CALENDARIO PROYECTO

Previsión tareas para los próximos meses



### 6.2. Librerías externas utilizadas

Las librerías externas que usaremos para este proyecto son las que usamos durante el curso y además de estas una para poder realizar una generación de PDF. Las librerías utilizadas son:

#### 1- Jakarta Servlet JSP JSTL:

Biblioteca: jakarta.servlet.jsp.jstl y jakarta.servlet.jsp.jstl-api

Propósito: Proporciona la implementación y la API para la JSTL (JavaServer Pages Standard Tag Library), que facilita el desarrollo de JSPs.

#### 2- Spring Boot Starter Test:

Biblioteca: spring-boot-starter-test

Propósito: Incluye dependencias para realizar pruebas unitarias y de integración en aplicaciones Spring Boot. Incluye JUnit, Hamcrest, Mockito, y otras herramientas de prueba.

### **3- MySQL Connector:**

Biblioteca: mysql-connector-java

Propósito: Controlador JDBC para conectarse a bases de datos MySQL desde aplicaciones Java.

### **4- Logback Classic:**

Biblioteca: logback-classic

Propósito: Implementación del framework de logging, utilizado por defecto en aplicaciones Spring Boot para gestionar los logs de la aplicación.

### **5- JAXB (Java Architecture for XML Binding):**

Bibliotecas: jaxb-api, jaxb-runtime, jaxb-core, jaxb-impl

Propósito: Proporciona herramientas para mapear objetos Java a XML y viceversa. Es útil para trabajar con servicios web y otros sistemas que requieren intercambio de datos en formato XML.

### **6- iText 7 (para manejo de PDFs):**

Bibliotecas: kernel, layout

Propósito: iText es una biblioteca para crear y manipular documentos PDF. kernel proporciona las funcionalidades básicas de manejo de PDFs y layout facilita la creación y disposición de contenidos en los documentos.

## **7. Manual de despliegue**

Lo primero que tendríamos que hacer sería descargarnos el repositorio o clonarlo. Una vez hecho este paso vamos a necesitar MySQL Workbench con el nombre “root” y su contraseña “PracticaRoot”, si no quisiéramos cambiar nuestro usuario de Workbench más adelante explicaremos otra forma de conectar a la base de datos. Una vez estemos dentro del programa importaremos el archivo BBDD.sql que se encuentra junto con la memoria, y una vez importado lo ejecutaremos para generar la base de datos.

Para la parte del servicio / back vamos a necesitar Eclipse, siendo necesario tener instalada la extensión Spring Tools 4 que se podrá obtener a través del Marketplace del propio Eclipse. Una vez abierto y con Spring instalado, importaremos la carpeta de EstudioTattoo a nuestra área de trabajo, haremos clic derecho sobre ella y dentro del menú que aparecerá seleccionaremos las opciones Run as > Spring Boot App. Hacemos click en ella y dejamos que el servidor se inicie.

Si anteriormente no entramos a MySQL Workbench con el usuario dado el servidor no funcionará. Para poder solucionar esto accederemos a la carpeta src/main/java, y dentro de com.estudio.utils estará el archivo DBUtils.java. En este se podrá modificar los campos “String user” y “String pass” por los del login personal del propio usuario y conseguir que el servidor inicie.

Ahora es el momento de desplegar la parte visual del proyecto / front. Para esto necesitaremos Visual Studio Code, donde abriremos la carpeta de frontstudio dentro de su área de trabajo. Ahora tenemos que hacer clic derecho sobre la carpeta y seleccionar la opción de Abrir en terminal integrado. Esto nos abrirá una consola en la parte inferior de la zona de trabajo, donde en primer lugar tendremos que introducir y ejecutar el comando `npm install`. Una vez termine de ejecutarse, pondremos el comando `npm run serve`. Este comando puede dar error, ya que la parte de back de la aplicación estará funcionando en el puerto 8080. Si Visual Studio Code no asigna por defecto el puerto 8081 al ejecutar el comando, deberemos cambiarlo por `npm run serve -- --port 8081`. Una vez termine de iniciarse, nos dará la ruta <http://localhost:8081>, donde podremos hacer ctrl + clic, abriendo la aplicación completamente desplegada.