*PROYECTO***:**

**MeetHub**

|  |
| --- |
| *Imagen relacionada*  *con el proyecto* |

*Realizado por:* ***Rubén Descalzo Rodríguez***



|  |  |
| --- | --- |
|  | **Desarrollo de Aplicaciones Web (DAW)** |
| **Valladolid, a dé de 2024** |

Contenido

[1. Introducción 2](#_Toc166426848)

[2. Lenguajes empleados 3](#_Toc166426849)

[3. Distribución 4](#_Toc166426850)

[4. Requisitos de los clientes 8](#_Toc166426851)

[5. Licenciamiento 9](#_Toc166426852)

[6. Recursos 9](#_Toc166426853)

[6.1. Hardware 9](#_Toc166426854)

[6.2. Software 9](#_Toc166426855)

[6.3. Humanos 10](#_Toc166426856)

[6.4. Previsión económica del coste del proyecto 11](#_Toc166426857)

[7. Descripción de la aplicación 12](#_Toc166426858)

[7.1. Funcionamiento General 12](#_Toc166426859)

[7.2. Arquitectura 13](#_Toc166426860)

[7.3. Interfaz 22](#_Toc166426861)

[7.3.1. Características Generales 22](#_Toc166426862)

[7.3.2. Adaptación a dispositivos móviles 28](#_Toc166426863)

[7.3.3. Usabilidad/accesibilidad 28](#_Toc166426864)

[7.4. Autoevaluación y conclusiones 28](#_Toc166426865)

[7.4.1. Valoración del Trabajo y dificultades 28](#_Toc166426866)

[7.4.2. Valoración de la herramienta o aplicación desarrollada 28](#_Toc166426867)

[7.4.3. Conclusiones 28](#_Toc166426868)

# Introducción

En el tiempo actual que nos encontramos, la eficiencia en la organización de reuniones es clave, ya sea para empresas como para estudiantes…

Aquí es donde entra Meethub, un proyecto innovador diseñado para cambiar la manera en la que gestionamos nuestros encuentros con los demás. A diferencia de las plataformas más usadas como Zoom, Temas, MeetHub ofrece una experiencia de usuario simplificada y accesible para todos.

MeetHub se integra perfectamente con el calendario de Google, permitiendo a los usuarios no solo programar sus reuniones, si no también mantener todos sus citas sincronizadas. Con esta aplicación, puedes crear reuniones especificando fecha, hora y lugar, ya sea de manera remota o presencial. Cada reunión creada genera automáticamente un enlace a Google Meet, facilitando la conexión entre los diferentes participantes.

Además, con MeetHub tendrás la flexibilidad de elegir entre los diferentes calendarios de Google disponibles en tu cuenta, ofreciendo una personalización alta. Invitar a amigos, compañeros de trabajo o de clase es un proceso rápido e intuitivo.

La aplicación no solo permite crear y editar reuniones, sino que también permite visualizar y gestionar todas tus reuniones ya previamente programadas.

# Lenguajes empleados

Java - Utilizo Java para el desarrollo del backend, por su robustez y portabilidad. Con el BackEnd gestiono la lógica y la interacción con los datos.

Spring Boot - Un Framework potente de Java para simplificar la configuración y el despliegue del backend. Este framework facilita la creación de microservicios independientes lo que facilita el desarrollo.

MySQL - Para gestionar el uso de las bases de datos relacionales. Es un sistema de gestión que permite almacenar y recuperar todos los datos de usuario de manera segura y eficaz, incluyendo detalles de las reuniones y usuarios registrados.

HTML - El HTML es empleado para estructurar el contenido de la página web de MeetHub. A través de HTML, definir los elementos de la interfaz del usuario, como los formularios, listado de reuniones y controles de navegación.

CSS - Se utiliza CSS para estilizar los elementos HTML. CSS me permite personalizar la apariencia de la plataforma, incluyendo colores, tipografías y layout, para garantizar la correcta experiencia visual.

Bootstrap - Para la maquetación de los elementos en la página web. Este framework de CSS facilita el diseño responsivo para que sea adaptativo a todos los tamaños de pantalla garantizando que MeetHub sea accesible desde cualquier dispositivo.

JavaScript -> Emplea JavaScript para añadir interactividad en la página web. Con JavaScript implementó las validaciones de cliente, la manipulación dinámica del DOM.

# Distribución

La aplicación está distribuida en GITHUB, y solo se podrá usar en Local  
Enlace

<https://www.github.com/rubendesrod/MeetHub-Project>

Errores que se han encontrado al poder resubir los commit al proyecto.

El acceso a las llamadas de los EndPoints y a la API te exige el tener unas credenciales “GOOGLE\_CLIENT\_ID” y “GOOGLE\_CIENT\_SECRET”, al tener estas credenciales escritas en el código, GITHUB no deja que haya claves secretas ni de autorización en el código de la aplicación.

Se han creado como variables de entorno en el sistema y se pudo solucionar el error, pero al haber commit anteriormente con las credenciales en el código, he tenido que ejecutar unos comandos de github que son los siguientes:

**```**

git filter-branch --force --tree-filter \

"find . -type f -exec sed -i ...'s/499287062338-uoc7i3fnf68t0m835ig2qj2l4evrmigi.apps.googleusercontent.com/System.getenv(\"GOOGLE\_CLIENT\_ID\")/g' {} +" \

--tag-name-filter cat -- --all

**```**

**```**

git filter-branch --force --tree-filter \

"find . -type f -exec sed -i 's/GOCSPX-ETsgC\_uiFvdgZOUQXgjYS3eeeIUd/System.getenv(\"GOOGLE\_CLIENT\_SECRET\")/g' {} +" \

--tag-name-filter cat -- --all

**```**

Estos comandos han corregido el error del texto plano en todos los commit que había del repositorio.

La aplicación no ha podido ser subida a un HOST por los siguientes problemas que presta Google a la hora de querer montar una aplicación con su licenciamiento.

El primer punto que presenta Google es que para poder hacer una aplicación y que pueda entrar cualquier usuario a ella, es poder subir la aplicación y no dejarla como una prueba este es un proceso en el que Google, verifica tu aplicación

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En este caso MeetHub se encuentra en un entorno de prueba con el cual para poder acceder con un usuario hay que darle de alta

Tabla

Descripción generada automáticamente

Estos usuarios son los únicos que podrán realizar el registro en la aplicación web con las cuentas de Google, en el caso de que se hiciera un login con otro usuario que no correspondiese con esos saldría un error

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Y el último punto sería un entorno en el que yo pudiese crear unas variables de entorno para poder guardar en el sistema las credenciales del GOOGLE\_CLIENT\_ID y las credenciales del GOOGLE\_CLIENT\_SECRET, ya que en el código se recogen las variables de esta manera

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



# Requisitos de los clientes

Comprobación de los diferentes navegadores más usados

|  |  |
| --- | --- |
| **NAVEGADOR** | **OK/KO** |
| Microsoft Edge  Versión 124.0.2478.80 (Compilación oficial) (64 bits) | OK |
| Google Chrome Versión 124.0.6367.158 (Build oficial) (64 bits) | OK |
| Opera One(versión: 109.0.5097.68)  Fuente de actualización:Stable  Sistema:Windows 10 64-bit  Versión de Chromium:123.0.6312.124 | OK |

(No se ha podido realizar pruebas de SAFARI, ya que no se ha podido desplegar y no se dispone de alcance a dispositivos IOS)

Requisitos para los Usuarios de Aplicación Web:

* Disponer de un ordenador/Teléfono móvil
* Disponer de acceso a internet
* Tener una cuenta de Google
* Disponer de una cámara en el dispositivo (Este es opcional)
* Preferiblemente disponer del Navegador de Google Chrome (Ver la tabla para otros Navegadores)

# Licenciamiento

**Cumplimiento de la Licencia a Terceros:**

> **Java y Spring Boot, MySQL**: Ambos componentes se distribuyen bajo la Licencia Pública General de GNU (GPL).

> **Bootstrap, HTML, CSS y JavaScript:** Usados bajo licencias Permisivas (Licencia MIT), que permite el uso y modificación y redistribución del software.

> **Google Calendar API:** se utiliza bajo los términos de servicio específicos de Google para API’s.

# Recursos

## Hardware

Un ordenador con acceso internet (ya sea por cable o innalámbrico), un móvil con acceso a internet, una webcam, un micrófono

## Software

Como software utilizado para el desarrollo de esta aplicación está el IDE de las Spring Tools Suite (Es un IDE el cual extiende del IDE Eclipse, está preparado para trabajar con Spring Boot, Gradle o Maven), En mi caso he utilizado la paquetización de Maven. También se ha requerido una cuenta en Google Cloud para la cual ha sido necesaria añadir la aplicación MeetHub y que tipo de API’s iban a ser utilizado en MeetHub a la hora de hacer llamadas.

## Humanos

Aquí dejo una tabla con todas las fases que ha habido durante el desarrollo de la aplicación desde que se realiza un boceto (En FIGMA) hasta que se acaba por completo y se realiza la paquetización con MAVEN-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***COMPUTO DE HORAS POR FASE*** | ***1 minuto = 0.01 1 hora = 1.00*** | | |
| ***Fecha*** | ***Fase*** | ***Lenguaje*** | ***Tiempo*** | ***ACABADO*** |
| **14/03/2024** | **Diseño** | **Figma** | **10** |  |
| **14/03/2024** | **Base de datos** | **MySQL** | **8** |  |
| **20/03/2024** | **Login** | **HTML/CSS** | **2** |  |
| **20/03/2024** | **Login** | **HTML/CSS** | **2** |  |
| **20/03/2024** | **Registro** | **HTML/CSS** | **1** |  |
| **20/03/2024** | **Registro** | **HTML/CSS** | **0.5** |  |
| **12/04/2024** | **Login** | **HTML/CSS** | **1.06** |  |
| **14/04/2024** | **Registro** | **HTML/CSS** | **1.45** |  |
| **15/04/2024** | **Registro** | **HTML/CSS** | **1.07** |  |
| **15/04/2024** | **Poner Contraseña** | **HTML/CSS** | **0.8** |  |
| **16/04/2024** | **Poner Contraseña** | **HTML/CSS** | **0.5** |  |
| **17/04/2024** | **Poner Contraseña** | **HTML/CSS** | **0.5** |  |
| **21/04/2024** | **Perfil** | **HTML/CSS** | **4.2** |  |
| **22/04/2024** | **Perfil** | **HTML/CSS** | **2** |  |
| **23/04/2024** | **Perfil** | **HTML/CSS** | **5** |  |
| **24/04/2024** | **Perfil** | **HTML/CSS** | **2.04** |  |
| **25/04/2024** | **Perfil** | **HTML/CSS** | **1.1** |  |
| **26/04/2024** | **Crear reunión** | **HTML/CSS** | **5.3** |  |
| **27/04/2024** | **Editar reunión** | **HTML/CSS** | **0.5** |  |
| **28/04/2024** | **Editar reunión** | **HTML/CSS** | **2.06** |  |
| **29/04/2024** | **Calendario** | **HTML/CSS** | **2** |  |
| **30/04/2024** | **Calendario** | **HTML/CSS** | **2** |  |
| **01/05/2024** | **Poner Contraseña** | **HTML/CSS** | **1** |  |
| **01/05/2024** | **Diseño** |  | **1** |  |
| **01/05/2024** | **Diseño** |  | **4.4** |  |
| **02/05/2024** | **Registro** | **JAVA** | **3.35** |  |
| **03/05/2024** | **Registro** | **JAVA** | **3** |  |
| **03/05/2024** | **Login** | **JAVA** | **5.13** |  |
| **04/05/2024** | **Calendario** | **JAVA** | **7.4** |  |
| **05/05/2024** | **Calendario** | **JAVA** | **4** |  |
| **05/05/2024** | **Perfil** | **JAVA** | **1** |  |
| **06/05/2024** | **Perfil** | **JAVA** | **6** |  |
| **07/05/2024** | **Crear reunión** | **JAVA** | **7** |  |
| **08/05/2024** | **Editar reunión** | **JAVA** | **7.2** |  |
| **10/05/2024** | **Paquetización** |  | **2** |  |
| **10/05/2024** | **GitHub** | **MarkDown** | **1** |  |
| **10/05/2024** | **GitHub** |  | **2** |  |

Aquí dejo un total de horas, sumando la columna de **Tiempo** completa

|  |
| --- |
| **TOTAL DE HORAS** |
| ***108.57*** |

## Previsión económica del coste del proyecto

En la previsión de coste del proyecto haciendo los cálculos apropiados y según como anda el precio de un programador, es de 10,77€ La hora

Si el **total** de tiempo de la aplicación ha sido de **108 horas y 57 minutos**

* 108,57 x 10,77 = **1.169,30 €**

Esto es un total sobre las horas que se han tardado en hacer la aplicación completa con diseños...

# Descripción de la aplicación

## Funcionamiento General

La aplicación nos permite hacer Login, Registrarnos, ver nuestros calendarios, modificar nuestro perfil, crear una reunión con invitados y enlace al meet y editar nuestras propias reuniones.

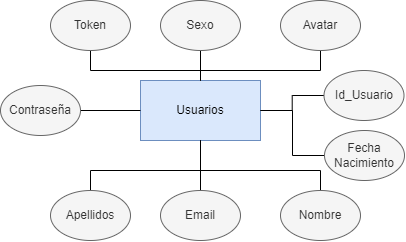
* Login -> Una vez nos hayamos registrado en la aplicación previamente, se permite hacer el login con la cuenta de correo que nos hemos registrado.
* Registro -> En el registro es obligatorio usar una cuenta de Google ya que MeetHub hace uso de la API Google Calendar
* Perfil -> Permite modificar el perfil del usuario, nombre, apellidos, sexo, fecha de nacimiento, cambiar contraseña y borrar la cuenta
* Crear Reunión -> Implementa la creación de una reunión poniendo su Nombre, el modo presencial u online, fecha de la reunión, hora de empiece y hora de finalización y los correos de los invitados
* Editar reunión -> Permite editar las reuniones en la que el usuario es propietario de ellas, es decir, el creador, y se pueden modificar la fecha las horas y el nombre o la descripción.
* Calendario -> Se muestra un calendario al usuario, de con un calendario predeterminado de Google. Se mostrará en la interfaz un elemento para que el usuario pueda seleccionar que calendario quiere mostrar a través de este. El calendario tiene vistas diferentes como la vista semanal, diaria y la mensual, dentro del calendario se puede hacer clic en el día para que se muestre un modal y el usuario pueda crear un evento (no una reunión), al cual le puede poner nombre y descripción, a la vez, que el calendario muestra reuniones se puede hacer clic en ellas y que nos lleve al enlace meet de esta

## Arquitectura

* + 1. **Diseño de la base de datos**

Solo se ha utilizado una tabla para la base de datos, ya que lo único que necesitaba guardar eran datos de los usuarios y lo mas importante el token de actualización que genera Google al autenticarse un usuario con el OAuth2.0 de Google(Ese token me permite generar tokens de acceso y hacer llamadas a la API con solo una autenticación)

**MODELO ENTIDAD-RELACIÓN**

****

**MODELOS LÓGICO**

**Tabla: Usuarios**

* **ID\_Usuario (INT, clave primaria, autoincrementable):** Un identificador único para cada usuario.
* **Email (VARCHAR, único, no nulo):** El correo electrónico del usuario, utilizado como identificador para el inicio de sesión**.**
* **Contraseña (VARCHAR, no nulo):** La contraseña del usuario, almacenada de forma segura usando hashing.
* **Nombre (VARCHAR, no nulo):** Nombre del usuario que se obtiene de la API de Google cuando se registra el usuario
* **Token (VARCHAR):** Un token que permite mantener la sesión del usuario activa con la API de Google Calendar sin necesidad de re-autenticación frecuente**.**
* **Avatar (VARCHAR):** URL del avatar del usuario que se ha registrado en la aplicación
* **Apellidos (VARCHAR):** Apellidos del usuario, ya que en Google puedes o no tener apellidos puede ser nulo.
* **FechaNacimiento (VARCHAR):** Fecha de nacimiento del usuario que se ha registrado en la aplicación
* **Sexo (ENUM(‘Hombre’,’Mujer’)):** Solo puede obtener dos valores Hombre o Mujer y representa la sexo del usuario

**NORMALIZACIÓN**

**Primera Forma Normal (1FN):**

* La tabla **Usuarios** ya está en 1FN, ya que no tiene atributos multivaluados.

**Segunda Forma Normal (2FN):**

* La tabla **Usuarios** ya está en 2FN, ya que no tiene dependencias parciales en sus atributos.

**Tercera Forma Normal (3FN):**

* La tabla **Usuarios** no tienen dependencias transitivas, por lo que ya están en 3FN.

**MODELO FÍSICO**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**SCRIPT SQL – BASE DE DATOS**

**Texto

Descripción generada automáticamente con confianza media**

* + 1. **Arquitectura del sistema**

Mi aplicación contiene una arquitectura MVC – Modelo, Vista, Controlador

Texto

Descripción generada automáticamente

Se destacan las carpetas, static, templates, proyect, controllers, models, repositorys, services y target.

También se hacen destacar algunos archivos como, application.properties y pom.xml.

* La carpeta **static** contiene recursos estáticos, los cuales pueden ser los CSS, JavaScript, imágenes…
* Imagen que contiene Texto

  Descripción generada automáticamente
* En **templates** encontraremos las plantillas/Vistas HTML que usa Thymeleaf para poder mostrar al usuario la pantalla que necesite
* Texto

  Descripción generada automáticamente con confianza media
* **project** contiene todas las demás carpetas que conforman el código de la aplicación y el fichero main.
* Texto

  Descripción generada automáticamente
* **controllers** contiene todos los ficheros controladores, es decir, donde se encuentran los métodos que obtienen la ruta del navegador, y en cuestión del tipo de ruta que se envíe desde el navegador responderá un método u otro.
* Texto

  Descripción generada automáticamente
* Métodos para destacar es del fichero **OAuth2Controller**, ya que aquí se encuentran los métodos para autorizar al usuario con Google

En este método se construye una de las URL de redirección que se deben de configurar en la Consola de Google Cloud una vez que sacamos las credenciales, este método es llamada por el principal de la autenticación.

* Texto

  Descripción generada automáticamente

Este método es el principal de la autorización, ya que es el que se encarga de redireccionar al usuario a la autorización una vez es llamado

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Y este método redirige al siguiente método que es captado por la ruta del navegador, el cual obtiene el token de actualización del usuario para poder generar tokens de acceso y hacer llamadas a la API

Texto

Descripción generada automáticamenteEstas rutas tanto /authrize como /callback hay que tenerlas en Google console como que pueden ser usadas para la validación OAuth2.0, cuando sacamos las credenciales se nos permite añadir URI de redireccionamiento

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* También caben a destacar los ficheros controladores **GlobalExceptionHandler** y el fichero **CustomErrorController**

Ambos ficheros contienen un manejo de errores, pero su uso en este caso es para cuando el usuario quiere acceder a una ruta que no existe en el navegador y redirigir a una página 404 propia de la aplicación.

Texto

Descripción generada automáticamente

Recoge esta excepción, la cual indica que se ha podido encontrar un elemento de la WEB, creo un objeto ModelAndView, al cual le puedo pasar directamente un Objeto o Atributo como se hace en las vistas con Model y el nombre de la vista a la que se quiere redirigir al usuario, en este caso es a la vista error404, y el navegador redirige al usuario a la vista con la que se ha controlado el error.

Imagen que contiene Icono

Descripción generada automáticamente

* **models** contiene los DTO y los DAO, en este caso solo hay un DAO ya que la base de datos solo tiene una tabla.
* Texto

  Descripción generada automáticamente
* **repositorys** están los ficheros que extienden de JPARepository, clase la cual permite poder hacer consultas sin tener que escribir una consulta SQL, por si sola trae consultas como save, update, delete.
* 
* **services** están los ficheros que se conectan con los repositorios para agrupar su funcionalidad y poder añadir lógica antes de hacer una consulta, También se encuentra un servicio de Google para poder gestionar los detalles de la cuenta del usuario que se registra(nombre, apellidos, avatar…)
* Texto

  Descripción generada automáticamente
* **target** como el proyecto está mavenizado, para ejecutar la aplicación sin tener que entrar al código, se puede hacer un install con alguna depencia en el pom, para poder crear un fichero jar el cual esta en esta carpeta, para poder ejecutar la aplicación sin tener que ejecutar el main. Abrimos una CMD en la carpeta target y ponemos **java -jar MeetHub-2.5.0.jar** cuando le ejecutemos saldrá en la consola el Logo de Spring Boot, y cuando acabe mostrará un mensaje de [Started MeetHubApplication in 5.635 seconds (process running for 6.028], y el puerto local por el que se abre la aplicación está en un mensaje anterior a este mostrado.
* Texto

  Descripción generada automáticamente
* **application.properties** en este archivo están las propiedades con las que se ejecuta el servidor apache que lanza spring boot automáticamente, como por ejemplo la configuración del puerto en el que se abre el servidor, la ruta de la base de datos con el usuario y contraseña de esta.
* **Texto

  Descripción generada automáticamente**
* **Pom.xml** este es el archivo principal con el que MAVEN organiza e instala las dependencias que necesita el proyecto. También se encuentra el plugin de paquetización de Maven, para poder extraer el proyecto a un .jar y ejecutarle desde consola
* Esta es la configuración principal del proyecto en el POM
* **Texto

  Descripción generada automáticamente**
* Así se organizan las dependencias en MAVEN
* **Texto

  Descripción generada automáticamente**
* Y los plugin para poder empaquetar la aplicación
* **Texto

  Descripción generada automáticamente**

## Interfaz

### Características Generales

Mi interfaz se ha diseño de una forma en la que no es habitual visualizar las páginas webs, ya que esta contiene un diseño horizontal y no vertical. He utilizado un diseño horizontal ya que este tipo de diseños web, suelen estar más enfocados a entorno de administración.

Ya que MeetHub se trata de una aplicación de administración de reuniones, se ha optado por este tipo de diseño web

Se han realizado previamente unos bocetos en FIGMA, para ir teniendo una idea de la aplicación

*Si este enlace se copia en el navegador se podrá acceder al FIGMA del proyecto empleado para el desarrollo*

https://www.figma.com/file/n1X0Exr4yz8bGLd4li8Ia1/TFG?type=design&mode=design&t=id1v6eKSsuT7kzea-1

Este es un boceto de la idea principal de la ventana “perfil”

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Los mockups preparados en Figma han sido utilizados para seguir un orden y el como se quiere que quede un diseño.

Para la navegación como se puede comprobar en la siguiente imagen, Se dispone de un menú lateral y de un menú arriba a la derecha el cuál, nos permite acceder a nuestro perfil y a cerrar sesión

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En este menú pequeño es donde se puede acceder al perfil y a cerrar la sesión en la aplicación

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Los lenguajes que han sido empleados para realizar esta interfaz han sido **HTML, CSS**, **JavaScript, Boostrap5** (Biblioteca de estilo y posicionamiento con HTML, CSS y JavaScript) **y FullCalendar** (biblioteca de JavaScript que permite crear un calendario con muchos estilos y personalizaciones)

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

<https://fullcalendar.io/docs>

La validación empleada en la página se realiza con los propios métodos en Spring Boot pasando los como atributos a la vista, la cual con Thymeleaf lee si ese atributo existe y muestra el contenido, Un Ejemplo:

Texto

Descripción generada automáticamente con confianza baja

Cuando el nombre no cumple con las validaciones se añade un atributo al model (Objeto Model – para mandar atributos desde el código a la vista, ya sean Objetos DTO, arrays…)



Se comprueba con un if de Thymeleaf si el atributo contiene información o si existe y dentro del span con el th:text se setea el texto que se añade al modelo como segundo parámetro en el model.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Esto es un ejemplo de cómo realiza las validaciones MeetHub hay mas validaciones como por ejemplo pueden ser, en el caso de crear una reunión.

Imagen que contiene Tabla

Descripción generada automáticamente

Estás misma validaciones se aplican a la hora de editar una reunión, ya que ambas partes llevan los mismos datos, aunque lo participantes no se puedan editar

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Hay validaciones a la hora de añadir evento a nuestro calendario, que no deja poner un nombre con menos de 3 letras o vacío, en este caso es una validación dinámica ya que se realiza con JavaScript, nada más entrar al modal saldrá el botón de agregar desactivado, y el input nombre cambiará de color a rojo si no vale la validación o a verde si sí que vale

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Las validaciones faltantes están en la vista “Perfil”, en la cual en el área información básica, se validan los campos

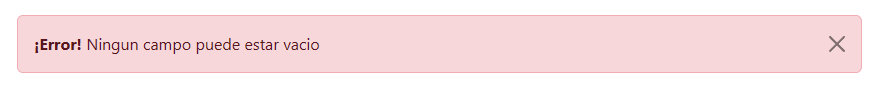
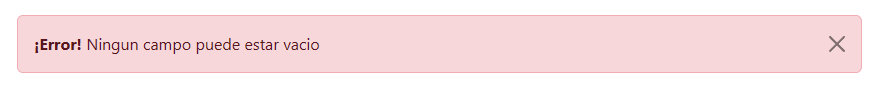
* Nombre -> Solo letras, minimo 3 de longitud y no puede estar en blanco
* Apellidos -> Solo letras, puede haber un espacio en blanco, puede estar en blanco

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Lo demás campos no son validados ya que pueden estar nulos en la Base de datos, una vez se ha validado los campo correctamente sin errores aparecerá el siguiente mensaje.

En el área de cambiar de contraseña se validan los siguientes aspectos

* Ninguno de los campos puede estar vacío
* 
* La contraseña actual tiene que coincidir con la que el usuario puso al registrarse en la aplicación
* 
* Nueva contraseña y la repetida tiene que coincidir
* 
* Y que contenga los requerimientos que se especifican abajo
* 

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En el caso de una correcta actualización de los datos muestra el siguiente mensaje



### Adaptación a dispositivos móviles

Para la mayor experiencia en la navegación de la web en móviles se ha realizado unos diseños responsivos para la aplicación, los cuales permitirán acceso a la aplicación con comodidad a todos los tipos de pantallas.

Como se habla en apartados anteriores MeetHub posee un diseño administrativo el cual tiene un enfoque mas horizontal que vertical, por el hecho de tener una barra de navegación en un lateral, para poder mejorar la experiencia de usuario en pantallas pequeñas-medianas se ha realizado la siguiente función.

Como se muestra en la imagen en el menú dropdown que solo tenía perfil y cerrar sesión, una vez la barra lateral desaparece, sus opciones aparecen en este menú para poder navegar en la web.

1. Vista del calendario

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

1. Vista de crear reunión

Interfaz de usuario gráfica

Descripción generada automáticamente

Como se aprecia para móvil había que optar por un diseño donde predominase la verticalidad, y se ha logrado gracias a que se ha guardado la barra de navegación en el menú del usuario, y se ha podido dejar las áreas como formato de la aplicación

### Usabilidad/accesibilidad

La interfaz cuenta con unos colores en los que el texto resalta sobre el color de fondo por eso se ha elegido fondos claritos y texto negro para dar el contraste, personas con dificultad visual, podrían leer correctamente lo que les quiere mostrar la aplicación.

Se ha realizado alguna prueba con un programa para personas con esa dificultad de lectura con un programa llamada JAWS (Lee los textos alternativos del código para que el usuario capte el mensaje), cuando una persona pulsa sobre un botón este programa analiza el botón y dice al usuario en donde ha pulsado y a donde se ha redirigido.

Texto

Descripción generada automáticamente

La interfaz responsiva también es un termino de accesibilidad a la aplicación de MeetHub ya que permite a cualquier usuario con un dispositivo de cualquier tamaño de pantalla hacer uso de la aplicación con comodidad

.

## Autoevaluación y conclusiones

### Valoración del Trabajo y dificultades

Este proyecto me ha permitido afianzar mis conocimiento con java, y sobre aprender más sobre el Framework de Spring Boot WEB, y sobre todo sobre su larga lista de dependencias, las cuales te ayudan a mejorar la eficacia del código

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Valoración de la herramienta o aplicación desarrollada

### Conclusiones