**Documentación Proyecto**

**Plataforma de Mensajería Electrónica**

**NRC:** 2093

**Equipo No.** 1

**Integrantes:**

*Ana María Santofimio Garatejo*

*Carlos Arturo Triana Rueda*

*Everlides Rebolledo Alvear*

*Fred de Jesús De Arco Romero*

*Sandra Paola Pérez Balen*

**Descripción de roles del equipo**

|  |  |  |  |
| --- | --- | --- | --- |
| Rol | Integrante | Descripción | Tareas |
| SCRUM Master | **Yurley (Tutora)** | Persona encargada de llevar el seguimiento del proyecto. | Seguimiento del correcto desarrollo de las tareas del proyecto. |
| Desarrollador Front-End | **Ana M. Santofimio**  **Everlides Rebolledo** | Persona encargada de realizar la parte visual de la aplicación. | Desarrollo de las vistas y estilos de la aplicación. |
| Desarrollador Back-End | **Carlos A. Triana** | Persona encargada de realizar la lógica de negocios de la aplicación. | Desarrollo de los controladores de la aplicación y diseño e integración de la base de datos. |
| Desarrollador Back-End | **Sandra P. Pérez**  **Fred De Arco** | Persona encargada de realizar la lógica de negocios de la aplicación. | Desarrollo de los controladores de la aplicación y diseño e integración de la base de datos. |
| Desarrollador Full-Stack | **Everlides Rebolledo** | Persona encargada de realizar la parte visual y la lógica de negocios de la aplicación. | Desarrollo de las vistas y estilos de la aplicación, de los controladores de la aplicación y diseño e integración de la base de datos. |

**Definición de artefactos**

**Backlog Sprint 1**

|  |  |  |  |
| --- | --- | --- | --- |
| User story | Descripción | Estimación | Responsable |
| Definición de roles | Definición de los roles de los integrantes del equipo de trabajo del proyecto. | 2 horas | Carlos/Fred/Ana/ Sandra/Everlides |
| Definición de artefactos | Definición de los artefactos de la metodología SCRUM para el desarrollo del proyecto. | 4 horas | Carlos/Fred/Ana/ Sandra/Everlides |
| Diseño del diagrama de clases | Diseño del diagrama de clases de la aplicación a desarrollar. | 2 horas | Carlos/Fred/Ana/ Sandra/Everlides |
| Definición del cronograma | Definición del cronograma de actividades del proyecto. | 2 horas | Carlos/Fred/Ana/ Sandra/Everlides |

**Backlog Sprint 2**

|  |  |  |  |
| --- | --- | --- | --- |
| User story | Descripción | Estimación | Responsable |
| Definición del mapa de navegabilidad | Definición del mapa de navegabilidad que mostrará la distribución de las vistas de la aplicación | 3 horas | Fred / Sandra |
| Selección de la plantilla de estilos | Selección de la librería CSS a usar para los estilos de las vistas de la aplicación. | 1 hora | Ana María / Everlides |
| Diseño e implementación de las vistas | Diseño e implementación de las vistas en HTML y CSS. | 15 horas | Ana María / Everlides |
| Creación del proyecto en GIT | Creación del proyecto en GIT y posterior cargue a la nube en la plataforma Github. | 1 hora | Carlos |

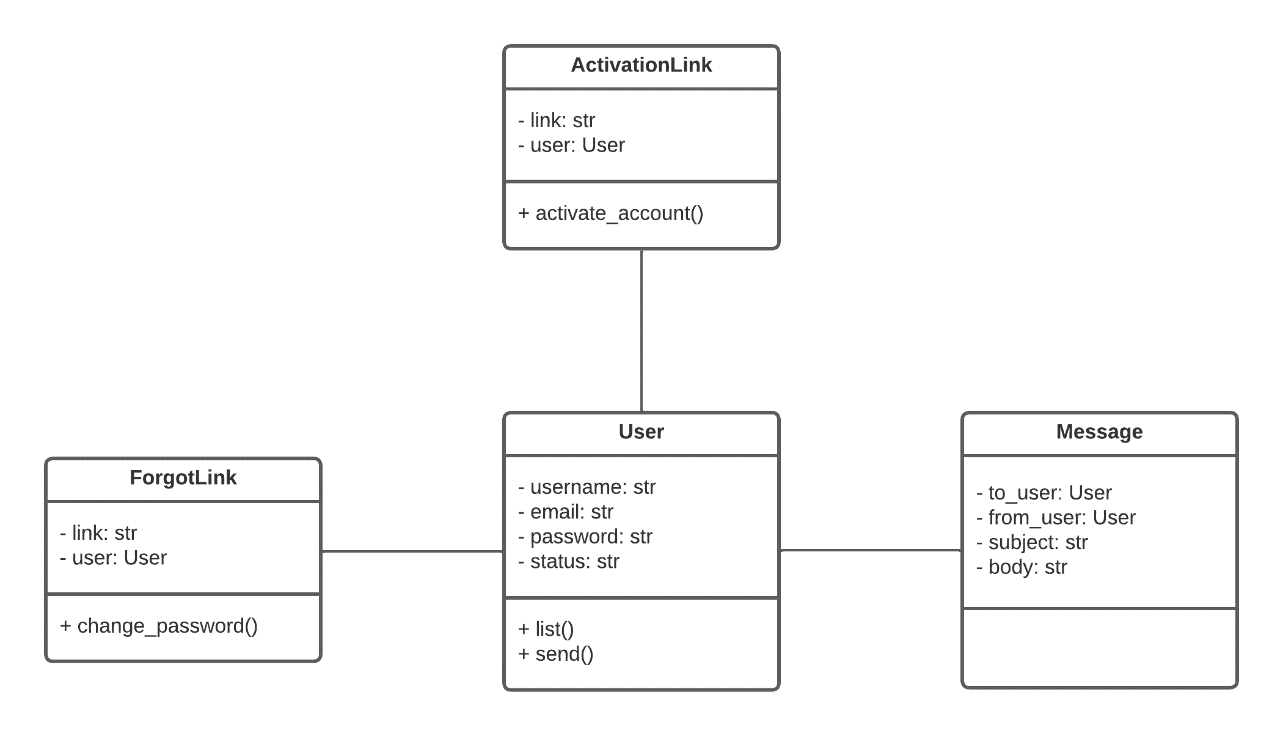
**Backlog Sprint 3**

|  |  |  |  |
| --- | --- | --- | --- |
| User story | Descripción | Estimación | Responsable |
| Diseño e implementación de los controladores para formularios y otras funcionalidades | Diseño e implementación de los controladores que representan la lógica de negocios de la aplicación. | 40 horas | Carlos / Fred |
| Diseño e implementación de base de datos | Diseño de la base de datos relacional que alojará los datos de la aplicación e implementación de la misma usando el motor SQLite. | 10 horas | Carlos / Fred |
| Desarrollo de integración de controladores y bases de datos | Integración de la base de datos con los controladores para la búsqueda y almacenamiento de información persistente de manera segura. | 30 horas | Carlos / Fred |
| Diseño e implementación de portal de acceso usando método de autenticación basado en usuario y contraseña | Diseño e implementación del método de autenticación de los usuarios y manejo de sesiones. | 20 horas | Carlos / Fred |

**Backlog Sprint 4**

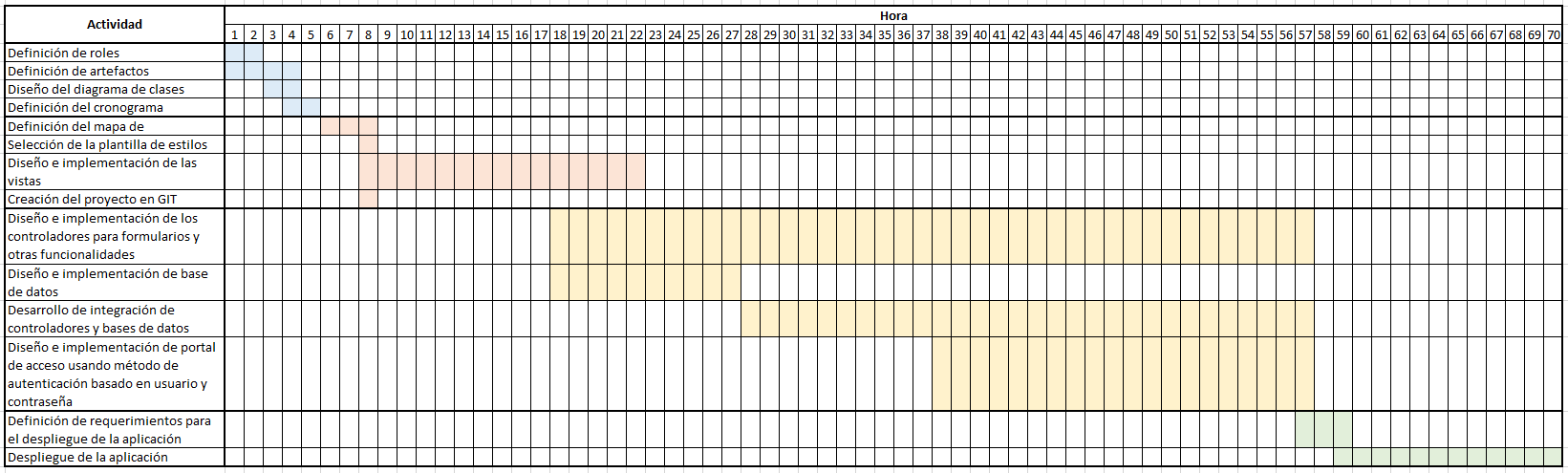
|  |  |  |  |
| --- | --- | --- | --- |
| User story | Descripción | Estimación | Responsable |
| Definición de requerimientos para el despliegue de la aplicación | Definición de los requerimientos necesarios para realizar el despliegue de la aplicación en la plataforma PythonAnywhere | 3 horas | Carlos/Fred/Ana/ Sandra/Everlides |
| Despliegue de la aplicación | Configuración, despliegue y verificación del funcionamiento de la aplicación en la plataforma PythonAnywhere | 12 horas | Carlos/Fred/Ana/ Sandra/Everlides |

**Diagrama de clases**

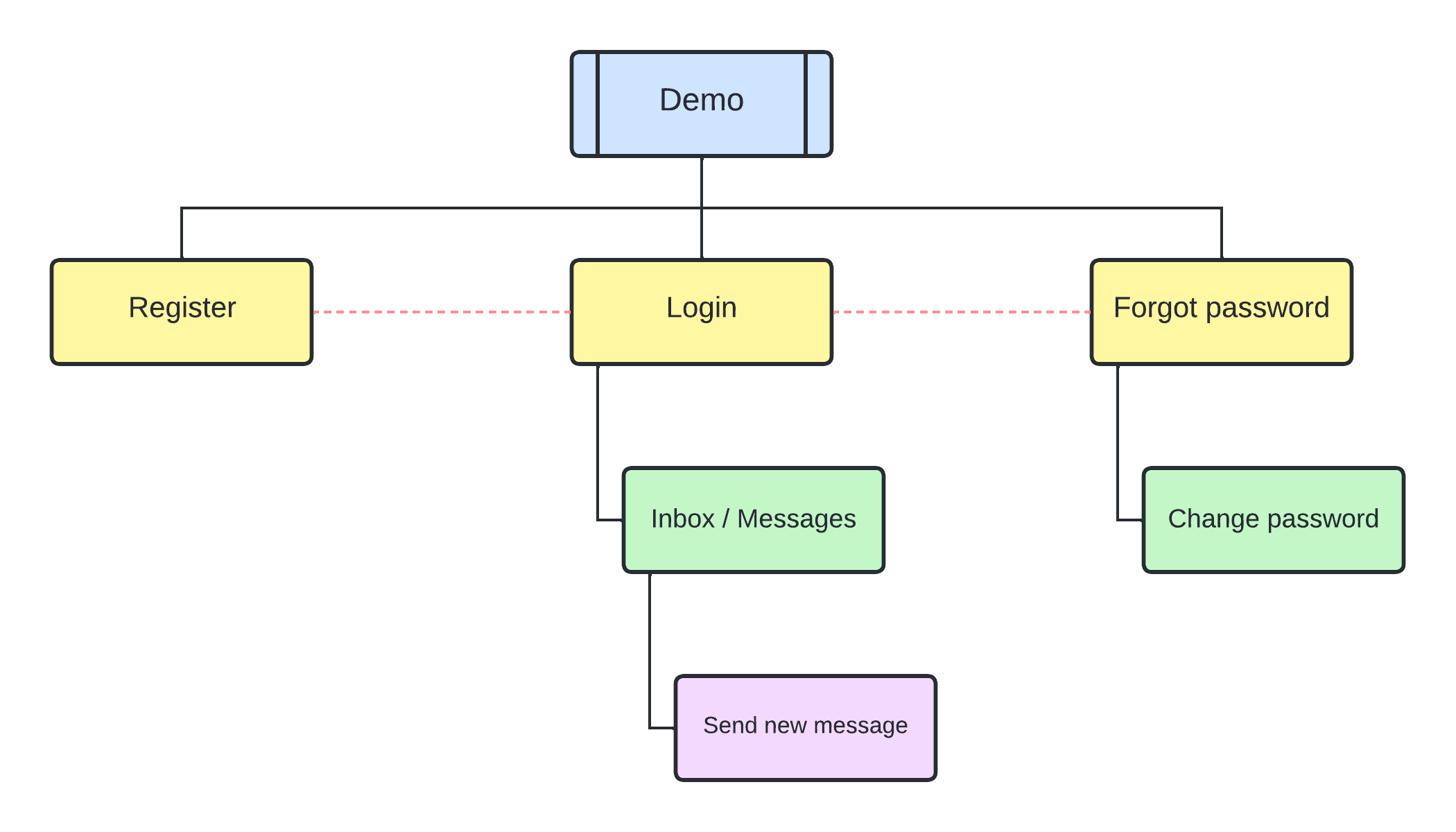


La clase usuario tendrá la información referente al usuario, es decir, su nombre de usuario, su correo, su contraseña y el estado de su cuenta. La clase mensaje tendrá la información de los mensajes de la plataforma, es decir, el usuario que envía el mensaje, el usuario que lo recibe, el asunto y el cuerpo del mensaje.

**Cronograma de tareas**



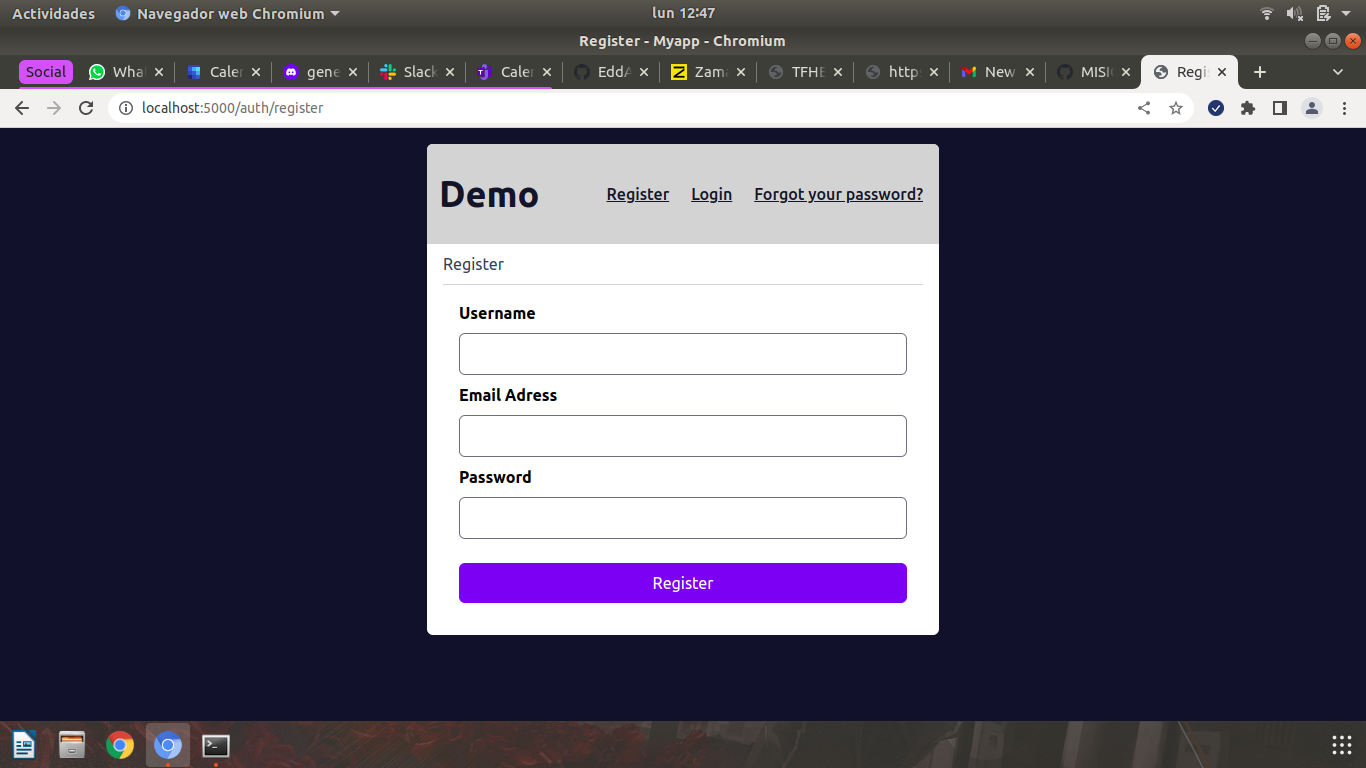
**Mapa de navegabilidad**



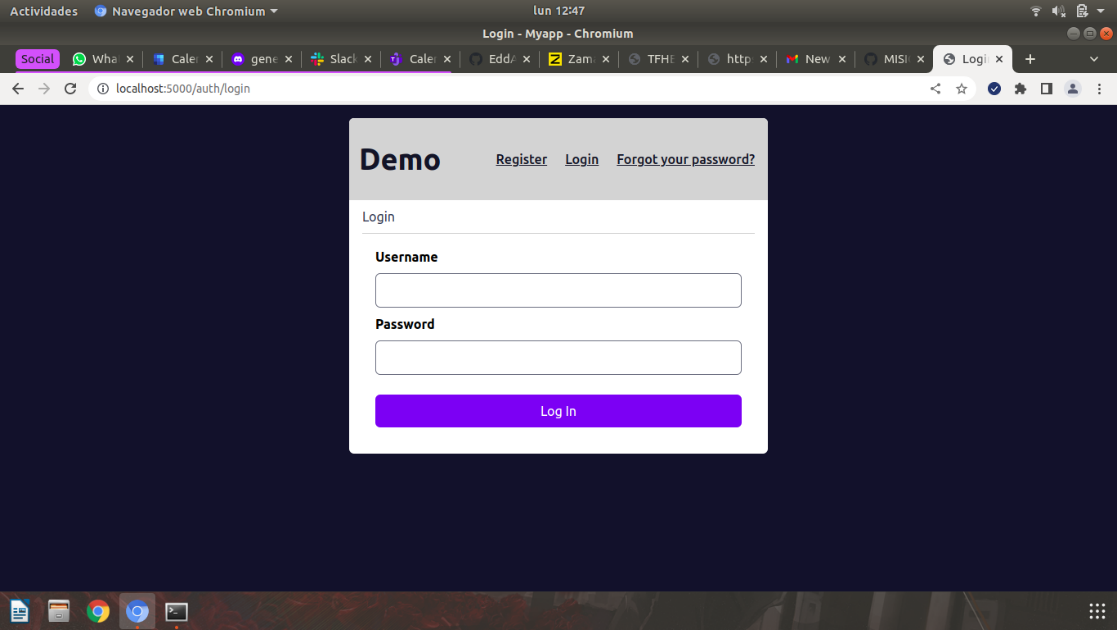
La aplicación en su pantalla principal tiene 3 opciones, estas son el registro, el inicio de sesión y la recuperación de contraseña, y se puede desplazar entre ellas. La pantalla de recuperación de contraseña, dará paso a la pantalla de cambio de contraseña al acceder a ella mediante el enlace enviado. La pantalla de inicio de sesión da paso a la de visualización de mensajes y esta a su vez da paso a la pantalla de envío de nuevo mensaje.

**Vistas de la aplicación**

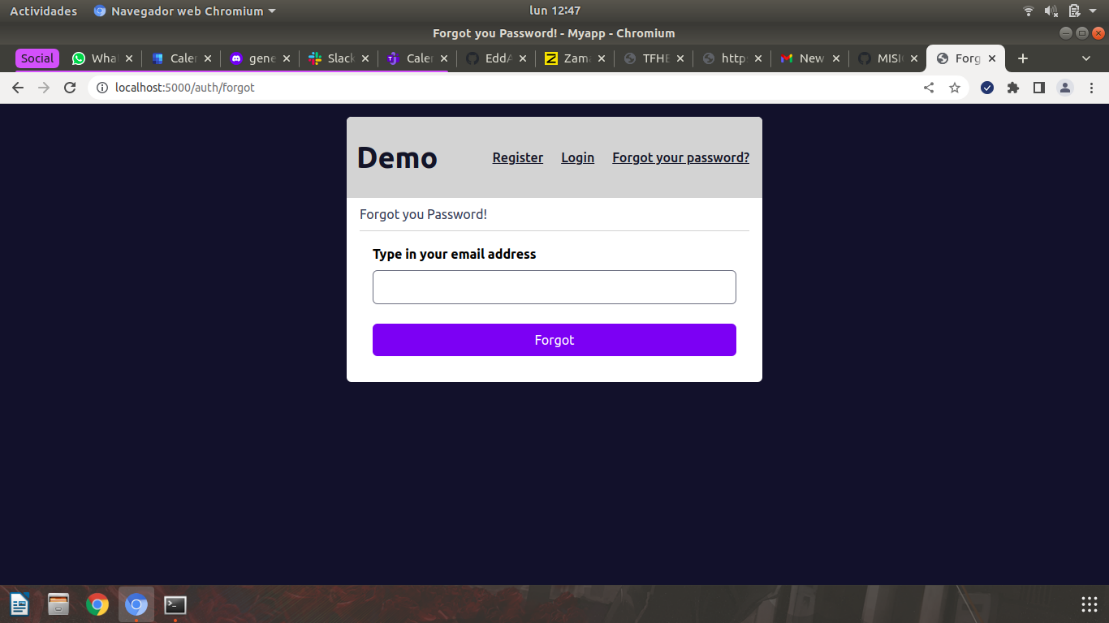
* **Vista de registro:**



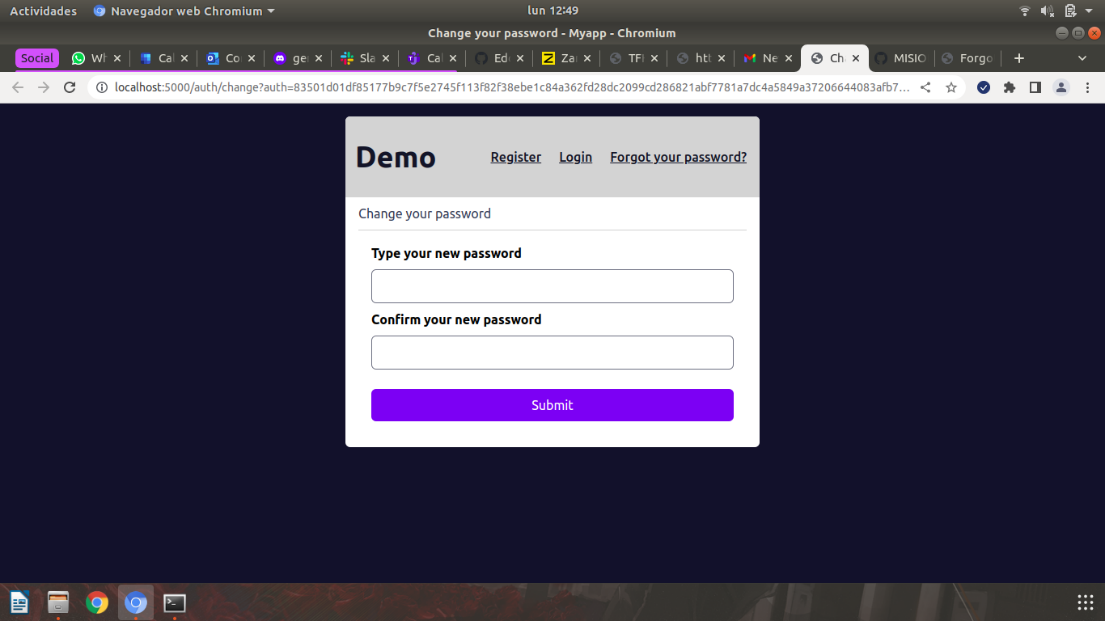
* **Vista de inicio de sesión:**



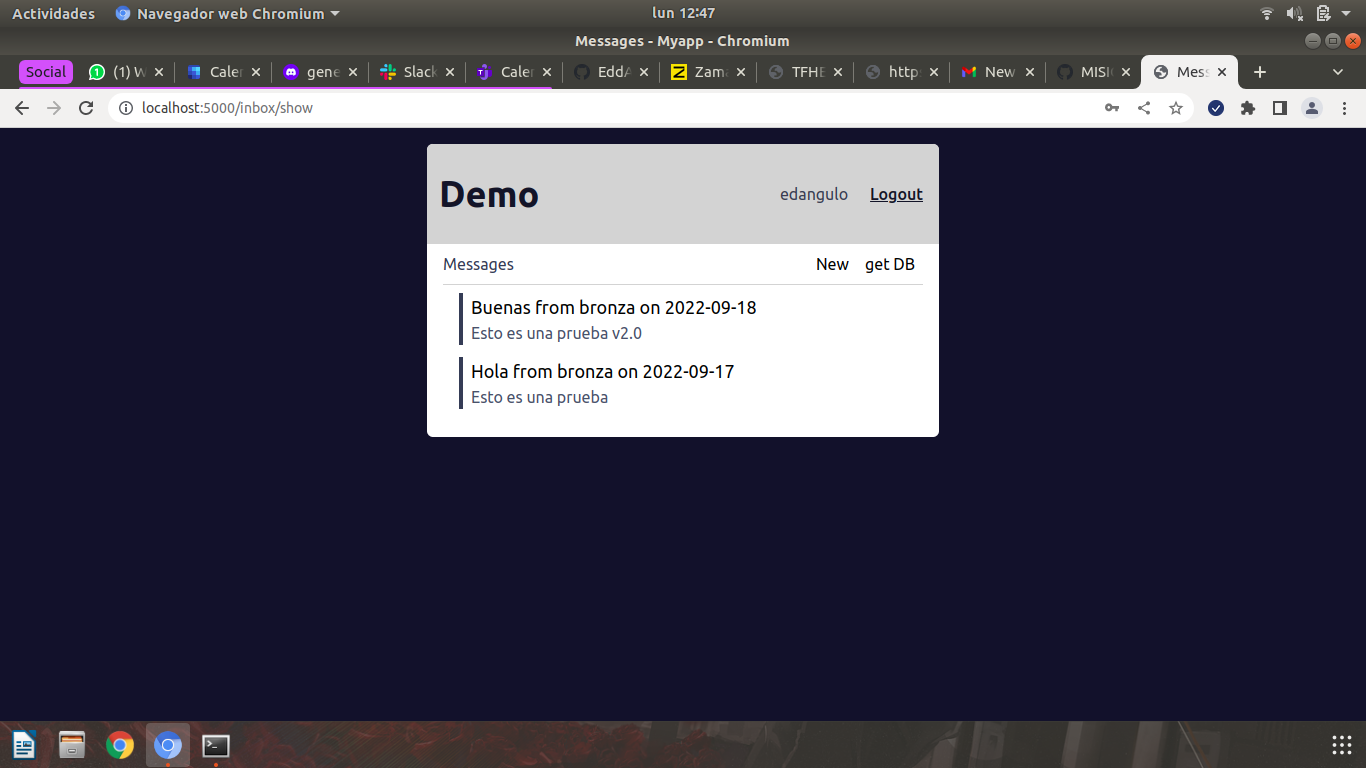
* **Vista de solicitud de recuperación de contraseña:**



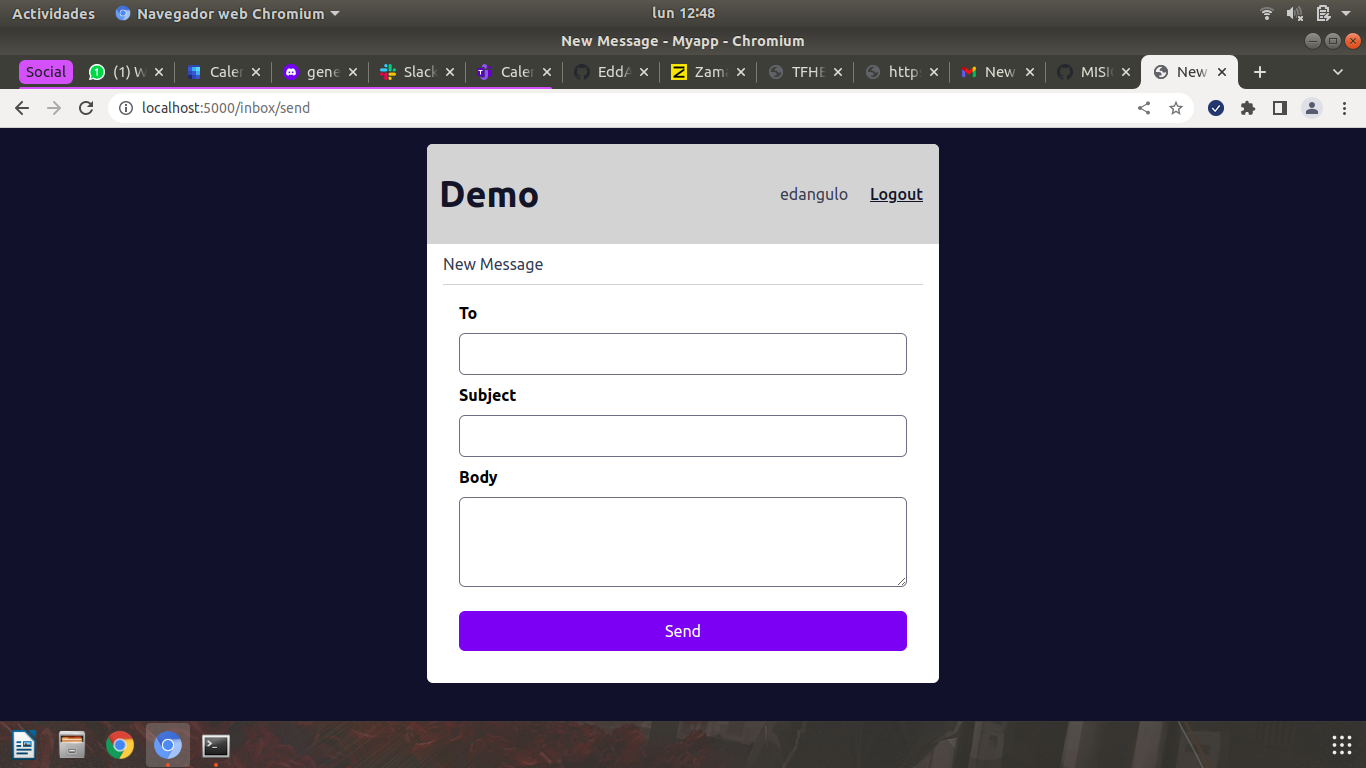
* **Vista de cambio/recuperación de contraseña:**



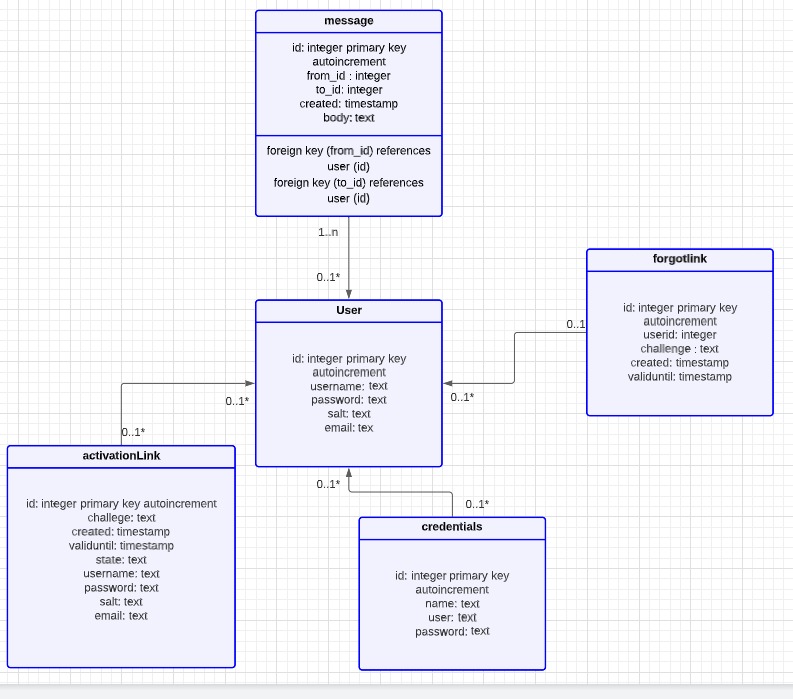
* **Vista de visualización de mensajes:**



* **Vista de envío de un nuevo mensaje:**



**Base de datos** (*Sprint 3*)



En la base de datos de la plataforma de mensajería electrónica demuestra que todo surge desde la creación de un usuario(user), que debe de contener requisitos como su nombre, id, contraseña, correo luego de esto se envía el enlace de activación de la cuenta (activationLink) al correo que este usuario escribió en la creación de la cuenta, teniendo en cuenta que los usuarios de esta aplicación debe de ser exclusivamente para los empleados de la empresa, luego de esto existe la clase de enviar y recibir mensajes (message) que permite la interacción sincrónica entre sus participantes, y si por algún motivo el usuario olvida su contraseña(forgoLink) este la puede recuperar por medio de su correo en el cual se envía el enlace del cambio de la contraseña, cada vez que el usuario ingrese a la plataforma este le pedirá sus credenciales (credentials).

**Métodos controladores del Back-End** (Sprint 3)

* Nombre método controlador 1
  + Ruta: bp.route - /activate
  + Métodos HTTP permitidos: 'GET', 'POST'
  + Lógica algorítmica:

Este controlador permite realizar la activación del usuario, mediante un método GET, esta solicitud se redirige a la base de datos, en donde si se encuentra el usuario, solicita usuario y contraseña. Si no se encuentra se ejecuta actualizar los estados de activación.

* Nombre método controlador 2
  + Ruta: bp.route -/register
  + Métodos HTTP permitidos:'GET','POST'
  + Lógica algorítmica:

Este controlador realiza el registro del usuario, se utiliza el método POST. Se solicita que sean llenados las solicitudes de los formularios de manera correcta (nombre de usuario, contraseña e Email), cuando detecta que hay un error en el registro en cualquiera de los formularios, hace un retorno especificando el requerimiento (por ejemplo, el usuario debe ser alfanumérico con signos o el correo es invalido, o ese usuario ya fue registrado, etc). Cuando se completan los requerimientos correctos, se ejecuta un enlace dirigido al correo electrónico, para finalizar con éxito el registro, este enlace hace un retorno a la url de ingreso.

* Nombre método controlador 3
  + Ruta: bp.route -/confirm
  + Métodos HTTP permitidos: GET, POST
  + Lógica algorítmica: Este controlador es el que se ocupa de la validación de la contraseña, se basa en el correcto diligenciamiento de la contraseña, verifica su validez, su confirmación y cumplimiento de requerimientos o especificaciones. Adicionalmente genera la contraseña hash.
* Nombre método controlador 4
  + Ruta: bp.route -/change'
  + Métodos HTTP permitidos: GET, POST
  + Lógica algorítmica: Este controlador utiliza el método GET, se encarga del redirigir al usuario a la bandeja de entrada para el ingreso de los datos.
* Nombre método controlador 5
  + Ruta: bp.route -/forgot'
  + Métodos HTTP permitidos: GET, POST
  + Lógica algorítmica: cuando el usuario ha olvidado usuario o contraseña, este controlador es el que se encarga hacer la solicitud del ingreso de un correo valido para enviar el enlace, este correo es verificado en la base de datos, si no esta, envía un mensaje indicando que no es el correo correcto, si lo es envía un enlace al correo.
* Nombre método controlador 6
  + Ruta: bp.route -/login
  + Métodos HTTP permitidos: GET, POST
  + Lógica algorítmica:

Este controlador utiliza el método POST, es el encargado de validar el correcto ingreso de los datos de usuario y contraseña, si alguno de ellos o ambos son incorrectos, envía la alerta de error o inválidos. O cuando no se llena alguno de los espacios, envía la alerta.

**Prácticas de programación segura (Sprint 3)**

En nuestra aplicación utilizamos las siguientes prácticas de programación segura:

* **Función de derivación de claves criptográficas**.

Usamos el métodos del paquete: **from werkzeug.security** como librería segura para generar y validar password encriptados. De este paquete utilizamos los métodos:

• **generate\_password\_hash**: Para crear hashes.

• **check\_password\_hash**: Para verificar una cadena generada por el método anterior.

* **Uso prepared statements.**

Se utilizaron las consultas de base de datos para actualizar y agregar de acuerdo con la instrucción.

* **Funciones hash.**

Del paquete **werkzeug.security** usamos **check\_password\_hash, generate\_password\_hash** para comparar el valor hash de la contraseña recuperado de la base de datos con la contraseña ingresada por el usuario. El valor de retorno de True indica que la contraseña es correcta.

* **Se usaron librerías seguras para manejo de sesión.**