

# **Documentación Proyecto**

## **Plataforma de Mensajería Electrónica**

**NRC: 2134**

**Equipo No. 03**

**Integrantes:**

*José Figueroa*  
*Alexander Vásquez*  
*Jennifer Ramírez*  
*Daniela Fontecha*  
*Starling Téllez*

## Descripción de roles del equipo

| Rol                             | Integrante               | Descripción   | Tareas   |
|---------------------------------|--------------------------|---|--|
| <b>SCRUM Master</b>             | Tutor                    | Persona encargada de llevar el seguimiento del proyecto.                                | Seguimiento del correcto desarrollo de las tareas del proyecto.  |
| <b>Desarrollador Front-End</b>  | Jennifer Ramirez         | Persona encargada de realizar la parte visual de la aplicación.                         | Desarrollo de las vistas y estilos de la aplicación.   |
| <b>Desarrollador Back-End</b>   | <b>Alexander Vásquez</b> | 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.   |
| <b>Desarrollador Back-End</b>   | <b>José Figueroa</b>     | 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.   |
| <b>Desarrollador Full-Stack</b> | <b>Daniela Fontecha</b>  | 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. |
| <b>Desarrollador Front-End</b>  | <b>Jennifer Ramírez</b>  | Persona encargada de realizar la parte visual de la aplicación.                         | Desarrollo de las vistas y estilos de la aplicación.   |

# Definición de artefactos

## Backlog Sprint 1

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

## Backlog Sprint 2

| User story                                   | Descripción  | Estimación | Responsable                         |
|--|--|------------|-------------------------------------|
| <b>Definición del mapa de navegabilidad</b>  | Definición del mapa de navegabilidad que mostrará la distribución de las vistas de la aplicación | 4 horas    | Straling Téllez                     |
| <b>Selección de la plantilla de estilos</b>  | Selección de la librería CSS a usar para los estilos de las vistas de la aplicación.             | 2 hora     | Jennifer Ramirez Y Daniela Fontecha |
| <b>Diseño e implementación de las vistas</b> | Diseño e implementación de las vistas en HTML y CSS.   | 10 horas   | Jennifer Ramirez Y Daniela Fontecha |
| <b>Creación del proyecto en GIT</b>          | Creación del proyecto en GIT y posterior   | 2 hora     | José Figueroa                       |

|  |   |  |  |
|--|---|--|--|
|  | cargue a la nube en la plataforma Github. |  |  |
|--|---|--|--|

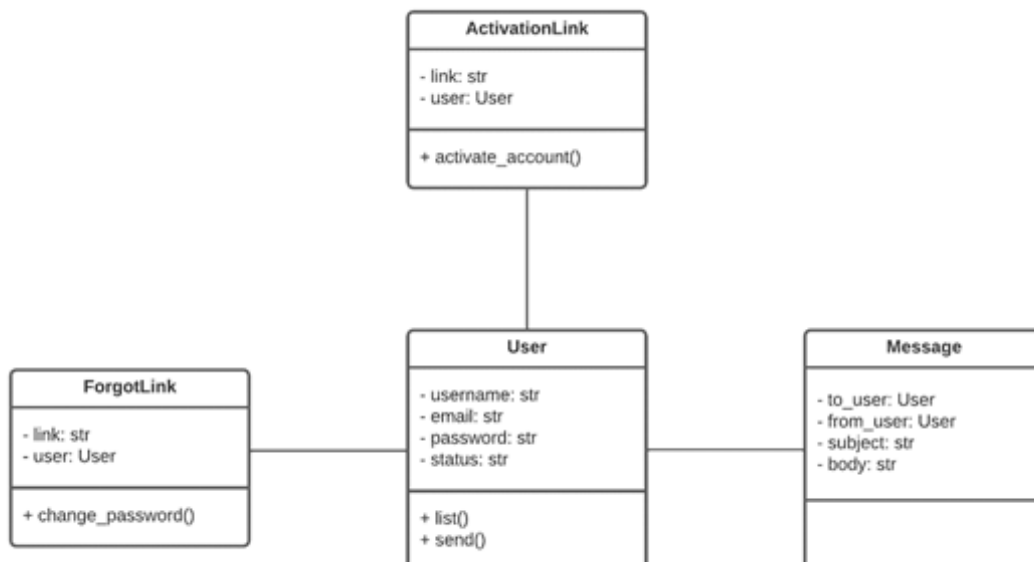
### Backlog Sprint 3

| User story   | Descripción  | Estimación | Responsable                           |
|--|--|------------|---------------------------------------|
| <b>Diseño e implementación de los controladores para formularios y otras funcionalidades</b>                     | Diseño e implementación de los controladores que representan la lógica de negocios de la aplicación.                                 | 30 horas   | José Figueroa                         |
| <b>Diseño e implementación de base de datos</b>  | 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   | Starling Téllez<br>José Figueroa      |
| <b>Desarrollo de integración de controladores y bases de datos</b>   | Integración de la base de datos con los controladores para la búsqueda y almacenamiento de información persistente de manera segura. | 30 horas   | Alexander Vásquez<br>Daniela Fontecha |
| <b>Diseño e implementación de portal de acceso usando método de autenticación basado en usuario y contraseña</b> | Diseño e implementación del método de autenticación de los usuarios y manejo de sesiones.  | 20 horas   | José Figueroa<br>Jennifer Ramírez     |

## Backlog Sprint 4

| User story  | Descripción  | Estimación | Responsable           |
|---|--|------------|-----------------------|
| <b>Definición de requerimientos para el despliegue de la aplicación</b> | Definición de los requerimientos necesarios para realizar el despliegue de la aplicación en la plataforma PythonAnywhere | 4 horas    | Todos los integrantes |
| <b>Despliegue de la aplicación</b>                                      | Configuración, despliegue y verificación del funcionamiento de la aplicación en la plataforma PythonAnywhere             | 10 horas   | Todos los integrantes |

## Diagrama de clases



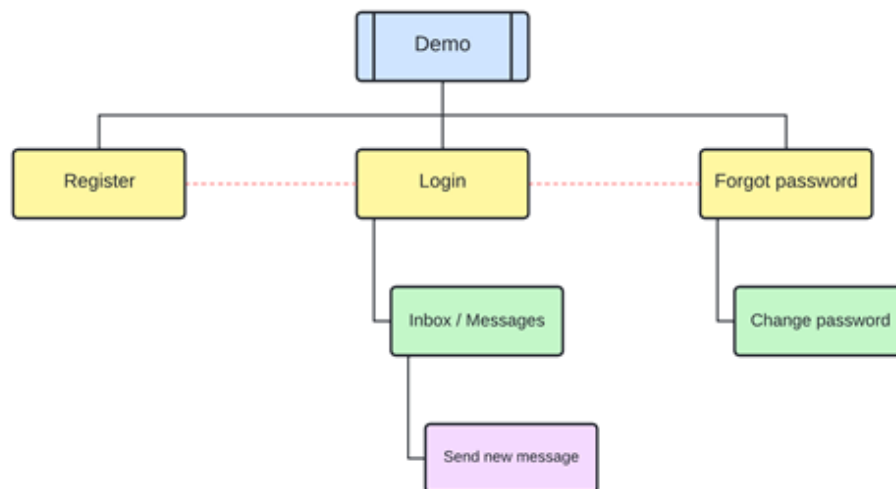
La clase usuario tendrá la información referente al usuario, es decir, su nombre de usuario, su correo, su contraseña y el estado de su cuenta. La clase mensaje tendrá

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

[illegible]

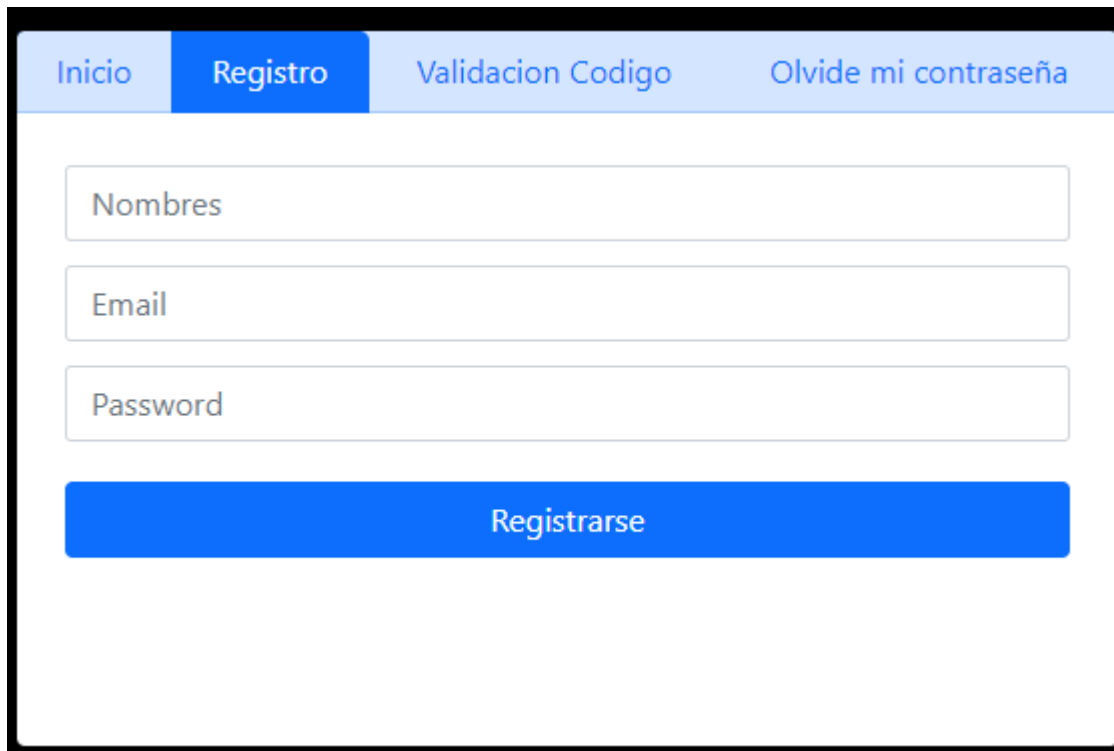
## Mapa de navegabilidad



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

## 1. Vista de registro:

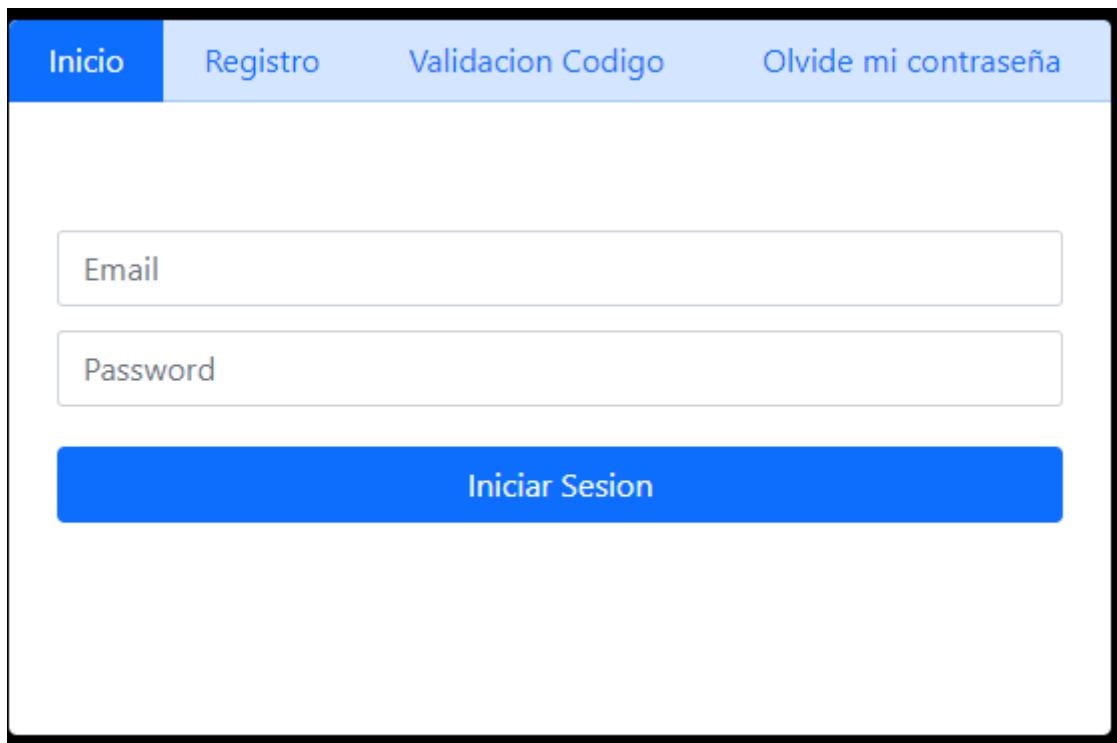


The registration form is displayed within a navigation bar. The 'Registro' tab is selected and highlighted in blue. Below the navigation bar, there are three input fields: 'Nombres', 'Email', and 'Password'. At the bottom of the form is a large blue button labeled 'Registrarse'.

|        |          |                  |                      |
|--------|----------|------------------|----------------------|
| Inicio | Registro | ValidacionCodigo | Olvide mi contraseña |
|--------|----------|------------------|----------------------|

Registrarse

## Vista de inicio de sesión:



The login form is displayed within a navigation bar. The 'Inicio' tab is selected and highlighted in blue. Below the navigation bar, there are two input fields: 'Email' and 'Password'. At the bottom of the form is a large blue button labeled 'Iniciar Sesion'.

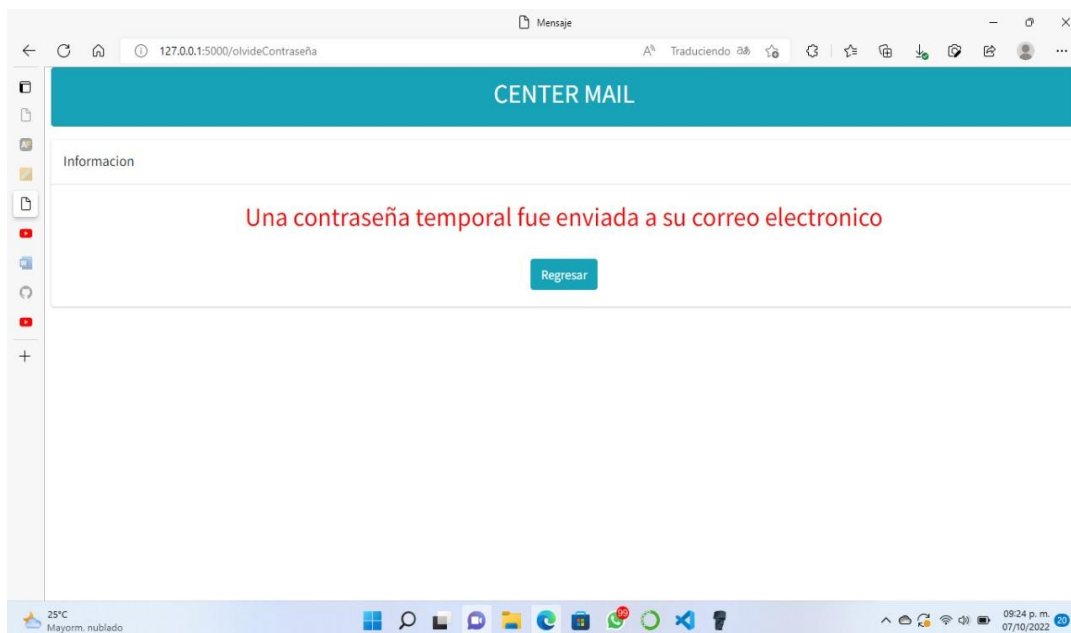
|        |          |                  |                      |
|--------|----------|------------------|----------------------|
| Inicio | Registro | ValidacionCodigo | Olvide mi contraseña |
|--------|----------|------------------|----------------------|

Iniciar Sesion

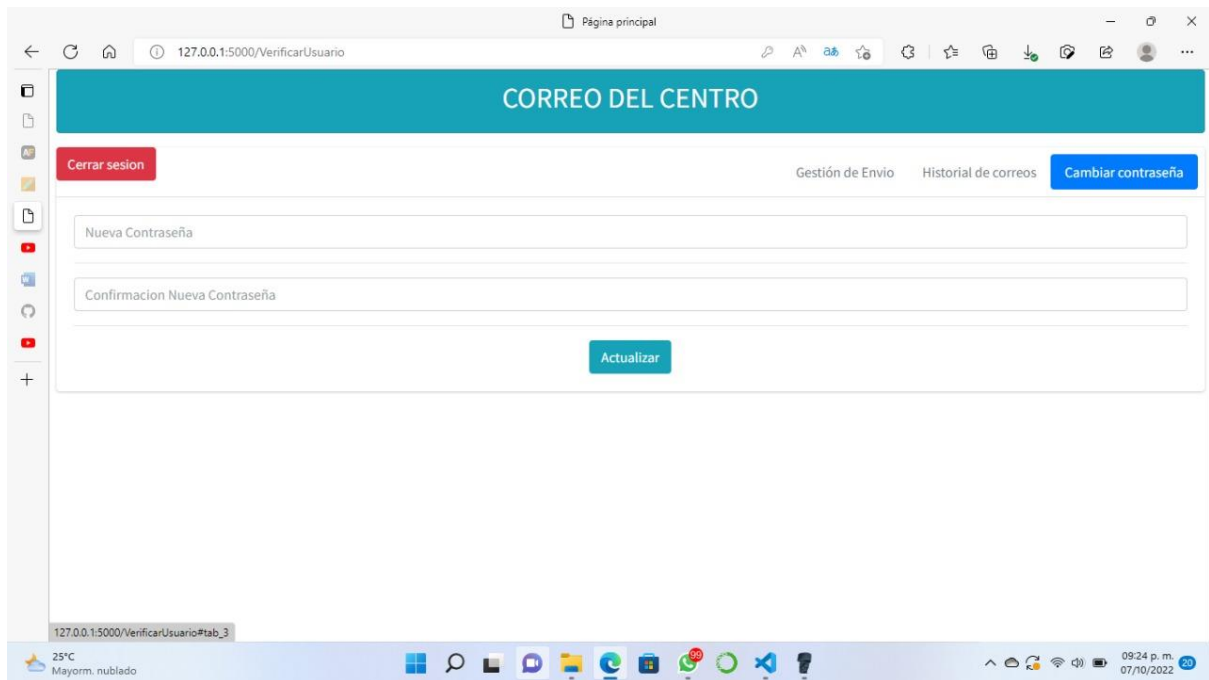
## 2. Vista de solicitud de recuperación de contraseña:

[Inicio](#) [Registro](#) [ValidacionCodigo](#) [Olvide mi contraseña](#)

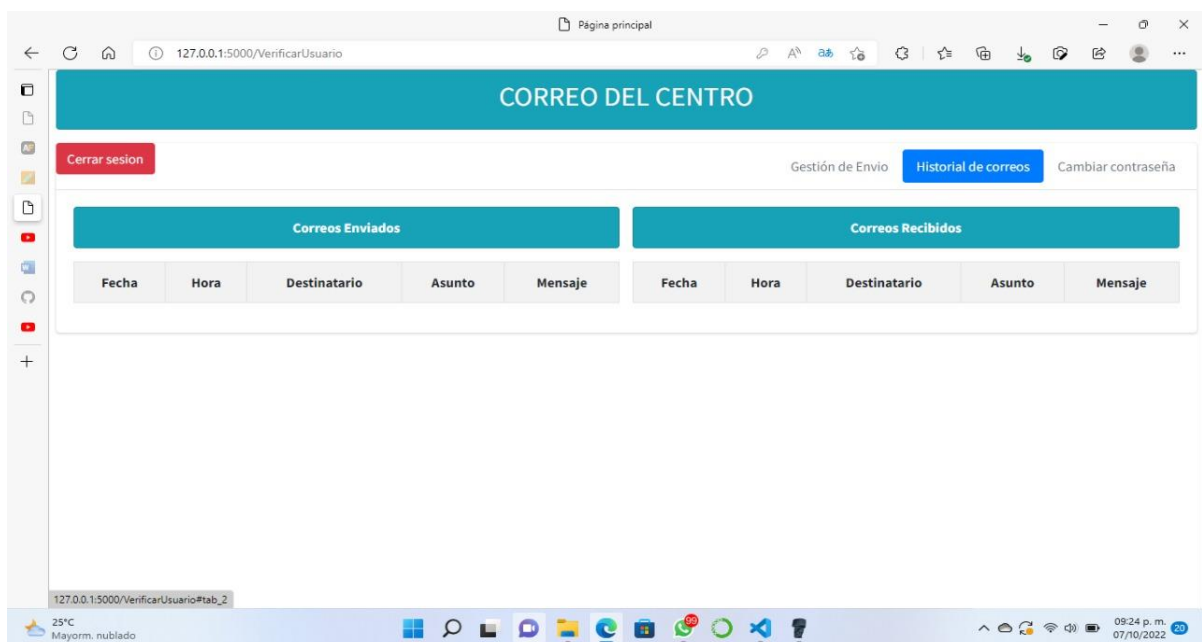
Buscar







### 3.Visualización de contraseña



### 4.Vista de visualización de mensajes:

## 5.Vista de envío de un nuevo mensaje:

CENTER MAIL

Cerrar sesion

Gestion de Envio

Historial de correos

Cambiar contraseña

Envio de Mensaje

Asunto

Seleccione un Destinatario...

Mensaje

Enviar

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

| Nombre   | Ruta      | Métodos HTTP permitidos | Lógica algorítmica   |
|----------|-----------|-------------------------|--|
| Register | /register | POST                    | Este método valida que los campos donde el usuario ingreso los datos de registro no estén vacíos y cumplan con ciertos criterios de diligenciamiento. Además, valida también que el usuario no se encuentre ya registrado en la aplicación y después inserta la contraseña encriptada y los demás datos a la tabla "activationlink" . Por último, envía enlace de activación a la cuenta de correo |

|          |           |      |  |
|----------|-----------|------|--|
|          |           |      | registrada con las respectivas indicaciones.   |
| Activate | /activate | GET  | Esta función valida que el enlace de activación enviado al correo del solicitante, todavía este vigente y que este coincida con el enviado, si es así, actualiza el estado del campo state de la tabla "activationlink" e inserta los datos de registro que el usuario ingreso, en la tabla "user" y redirige al template login.html   |
| Login    | /login    | POST | <p>Este método realiza las siguientes validaciones:</p> <ul style="list-style-type: none"> <li>• Si hay sesión activa redirige al inbox.show</li> <li>• Que los campos de logueo no estén vacíos.</li> <li>• Que los datos ingresados coincidan con los registrados en la base de datos.</li> <li>• Redirige al template show.html, si se cumplen todas las validaciones, sino se queda en el template login.html</li> </ul> |
| Forgot   | /forgot   | POST | <p>Este método realiza las siguientes validaciones:</p> <ul style="list-style-type: none"> <li>• Si hay sesión activa redirige al inbox.show</li> <li>• Valida que el campo correo no este vacío y que además este registrado en la base de datos</li> <li>• Actualiza el campo state de la tabla forgotlink</li> <li>• Envía enlace de cambio de contraseña al correo registrado.</li> </ul>                                |

|         |          |      |   |
|---------|----------|------|---|
| Change  | /Change  | GET  | <p>Esta función realiza las siguientes validaciones.</p> <ul style="list-style-type: none"> <li>• Si hay sesión activa redirige al inbox.show</li> <li>• Valida que el enlace de cambio de contraseña enviado al correo del solicitante, todavía este vigente y que este coincida con el enviado.</li> <li>• El enlace redirige al template change.html</li> </ul>  |
| Confirm | /confirm | POST | <p>Esta función realiza la siguiente rutina.</p> <ul style="list-style-type: none"> <li>• Si hay sesión activa redirige al inbox.show</li> <li>• Valida que los campos de cambio de contraseñas no estén vacíos y que además ambos coincidan. Sino renderiza el template change.html.</li> <li>• Si se cumplen las validaciones, actualiza el campo state de la tabla "forgotlink"</li> <li>• Actualiza el campo password de la tabla "user".</li> <li>• Y Redirige al template de login.html.</li> </ul> |
| Logout  | /logout  |      | Este método cierra la sesión y redirige al template login.html  |
| Show    | /show    |      | Esta función renderiza el template show.html donde aparecen los mensajes que han enviado a mi usuario.  |
| Send    | /send    | POST | Esta función ejecuta la siguiente lógica algorítmica.   |

|  |  |  |   |
|--|--|--|---|
|  |  |  | <ul style="list-style-type: none"> <li>• Valida que los campos to_id, subject, body no estén vacíos.</li> <li>• Valida que el destinatario exista en la base de datos.</li> <li>• Almacena en la tabla “message” la información correspondiente al mensaje enviado.</li> <li>• Si cumple con todas las validaciones redirige al template show.html, sino renderiza el template send.html</li> </ul> |
|--|--|--|---|

## Base de datos (Sprint 3)

| USUARIO          |  | MENSAJERIA    |
|------------------|--|---------------|
| ID               |  | ID            |
| NOMBREUSUARIO    |  | ASUNTO        |
| CORREO           |  | MENSAJE       |
| PASSWORD         |  | FECHA         |
| ESTADO           |  | HORA          |
| CODIGOACTIVACION |  | ID_USU_ENVIA  |
|                  |  | ID_USU_RECIBE |
|                  |  | ESTADO        |

La tabla user, almacenará la información de los usuarios y tendrá una relación de uno a muchos con las tablas “message” y “forgotlink”. Por otra parte, tenemos a la tabla “activationlink” la cual almacenara la información referente a la generación de enlaces de activación y la tabla “credentials” la cual almacenara los datos de la cuenta encargada de generar los enlaces de activación y recuperación de contraseña, ambas sin relaciones con las demás tablas.

## Prácticas de programación segura (Sprint 3)

*Las siguientes son buenas prácticas de programación aplicadas en el desarrollo de este proyecto:*

*Librerías seguras para generar y validar passwords encriptados (from `werkzeug.security` import `check_password_hash`, `generate_password_hash`)*

*Librería segura para Manejo de sesión.*

*No se ejecuta sentencias QUERY concatenando las variables para prevenir el ataque de inyección SQL a la base de datos. Evitando vulnerabilidades de acceso y modificación de datos.*

*Se establece y controla la entrada de datos desde el frontend validando los datos antes de ser procesados.*

**LINK DEL REPOSITORIO CON EL CODIGO FUENTE DESARROLLADO HASTA EL MOMENTO**

<https://github.com/djkdevise/Sprint3>