**Documentación Proyecto**

**Plataforma correo**

**NRC: 2104**

**Equipo No.** **10**

**Integrantes:**

*Giovanny Alexander Agudelo Marroquín*

*Samuel David Arteaga Isaza*

*Ricardo Cifuentes Yepes*

*Andrés Felipe Deulofeu Torres*

*Laura María Zabala Sierra*

**Contenido**

[Descripción de roles del equipo (Sprint 1) 6](#_Toc116477377)

[Definición de artefactos (*Sprint 1*) 7](#_Toc116477378)

[Backlog Sprint 1 7](#_Toc116477379)

[Backlog Sprint 2 7](#_Toc116477380)

[Backlog Sprint 3 8](#_Toc116477381)

[Backlog Sprint 4 9](#_Toc116477382)

[Diagrama de clases (*Sprint 1*) 9](#_Toc116477383)

[Cronograma de tareas (*Sprint 1*) 10](#_Toc116477384)

[Mapa de navegabilidad (*Sprint 2*) 10](#_Toc116477385)

[Vistas de la aplicación (*Sprint 2*) 11](#_Toc116477386)

[Vista de registro 11](#_Toc116477387)

[Vista de inicio de sesión 12](#_Toc116477388)

[Vista de validación de código 12](#_Toc116477389)

[Vista de envío de mensajes 13](#_Toc116477390)

[Vista de visualización de mensajes 13](#_Toc116477391)

[Vista de cambio de contraseña 14](#_Toc116477392)

[Vista de usuario registrado 14](#_Toc116477393)

[Vista de activación de usuario 15](#_Toc116477394)

[Vista de mensaje de código de activación 15](#_Toc116477395)

[Vista de plataforma GIT 16](#_Toc116477396)

[Métodos controladores del Back-End (*Sprint 3*) 16](#_Toc116477397)

[RegistrarUsuario 16](#_Toc116477398)

[EnviarMAIL 17](#_Toc116477399)

[ActivarUsuario 17](#_Toc116477400)

[HistorialEnviados 17](#_Toc116477401)

[HistorialRecibidos 18](#_Toc116477402)

[ActualizarPassword 18](#_Toc116477403)

[Base de datos (*Sprint 3*) 18](#_Toc116477404)

[Prácticas de programación segura (*Sprint 3*) 19](#_Toc116477405)

[SQL injection 19](#_Toc116477406)

[Encriptación de contraseñas 20](#_Toc116477407)

[Excepción de errores 20](#_Toc116477408)

[Despliegue de la aplicación en la nube (*Sprint 4*) 21](#_Toc116477409)

[Requerimientos 21](#_Toc116477410)

[Proceso 21](#_Toc116477411)

[Enlace final de la aplicación (*Sprint 4*) 34](#_Toc116477412)

**Índice de tablas**

[Tabla 1: Descripción roles equipo 6](#_Toc116423000)

[Tabla 2: Descripción Backlog Sprint 1 7](#_Toc116423001)

[Tabla 3: Descripción Backlog Sprint 2 7](#_Toc116423002)

[Tabla 4: Descripción Backlog Sprint 3 8](#_Toc116423003)

[Tabla 5: Descripción Backlog Sprint 4 9](#_Toc116423004)

[Tabla 6: Descripción métodos controladores Back-End 16](#_Toc116423005)

**Índice de imágenes**

[Imagen 1: Diagrama de clases 9](#_Toc116477336)

[Imagen 2: Cronograma de tareas del proyecto 10](#_Toc116477337)

[Imagen 3: Diagrama de clases 10](#_Toc116477338)

[Imagen 4: Vista de registro 11](#_Toc116477339)

[Imagen 5: vista inicio de sesión 12](#_Toc116477340)

[Imagen 6: Validación de código 12](#_Toc116477341)

[Imagen 7: Envío de mensajes 13](#_Toc116477342)

[Imagen 8: Visualización de mensajes 13](#_Toc116477343)

[Imagen 9: Cambio de contraseña 14](#_Toc116477344)

[Imagen 10: Usuario registrado 14](#_Toc116477345)

[Imagen 11: Activación de usuario 15](#_Toc116477346)

[Imagen 12: Activación de usuario 15](#_Toc116477347)

[Imagen 13: Vista repositorio GitHub 16](#_Toc116477348)

[Imagen 14: Tablas y relaciones- BD proyecto 18](#_Toc116477349)

[Imagen 15: Modelo protección contra SQL Injection 20](#_Toc116477350)

[Imagen 16: Encriptación contraseñas 20](#_Toc116477351)

[Imagen 17: Modelo excepción de errores 21](#_Toc116477352)

[Imagen 18: Login plataforma pythonanywhere.com 21](#_Toc116477353)

[Imagen 19: Ingreso plataforma web 22](#_Toc116477354)

[Imagen 20: Selección de FLASK 23](#_Toc116477355)

[Imagen 21: Selección de versión Python 23](#_Toc116477356)

[Imagen 22: Nombre proyecto y app principal 24](#_Toc116477357)

[Imagen 23: Creación de directorio 24](#_Toc116477358)

[Imagen 24: cargue de archivos 25](#_Toc116477359)

[Imagen 25: Open Batch 25](#_Toc116477360)

[Imagen 26: Consola Linux del servidor 26](#_Toc116477361)

[Imagen 27: Enrutamiento aplicación 26](#_Toc116477362)

[Imagen 28: redirección de ruta 27](#_Toc116477363)

[Imagen 29: redirección de ruta 27](#_Toc116477364)

[Imagen 30: asignación de ruta 28](#_Toc116477365)

[Imagen 31: vista login 28](#_Toc116477366)

[Imagen 32: Error en la conexión de la BBDD con el servidor 29](#_Toc116477367)

[Imagen 33: Asignación de ruta para la BBDD 30](#_Toc116477368)

[Imagen 34: Refresh para guardar cambios 30](#_Toc116477369)

[Imagen 35: Ingreso de usuario y contraseña 31](#_Toc116477370)

[Imagen 36: Vista de mensajería 31](#_Toc116477371)

[Imagen 37: validación de mensaje recibido 32](#_Toc116477372)

[Imagen 38: Pruebas de registro de usuario 32](#_Toc116477373)

[Imagen 39: Mensaje de registro 33](#_Toc116477374)

[Imagen 40: Mensaje de registro 33](#_Toc116477375)

[Imagen 41: Formulario de cambio de contraseña 34](#_Toc116477376)

# Descripción de roles del equipo (Sprint 1)

Tabla 1: Descripción roles equipo

|  |  |  |  |
| --- | --- | --- | --- |
| Rol | Integrante | Descripción | Tareas |
| SCRUM Máster | Tutor | Persona encargada de llevar el seguimiento del proyecto. | Seguimiento del correcto desarrollo de las tareas del proyecto. |
| Desarrollador Front-End | Integrante 1 | Persona encargada de realizar la parte visual de la aplicación. | Desarrollo de las vistas y estilos de la aplicación. |
| Desarrollador Back-End | Integrante 2 | 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 | Integrante 3 | 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 | Integrante 4 | 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 (*Sprint 1*)

## Backlog Sprint 1

Tabla 2: Descripción 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 | Todos los integrantes |
| Definición de artefactos | Definición de los artefactos de la metodología SCRUM para el desarrollo del proyecto. | 4 horas | Todos los integrantes |
| Diseño del diagrama de clases | Diseño del diagrama de clases de la aplicación a desarrollar. | 2 horas | Todos los integrantes |
| Definición del cronograma | Definición del cronograma de actividades del proyecto. | 2 horas | Todos los integrantes |

Fuente: autores del proyecto

## Backlog Sprint 2

Tabla 3: Descripción 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 | Integrante 1 y 4 |
| Selección de la plantilla de estilos | Selección de la librería CSS a usar para los estilos de las vistas de la aplicación. | 1 hora | Integrante 1 y 4 |
| Diseño e implementación de las vistas | Diseño e implementación de las vistas en HTML y CSS. | 15 horas | Integrante 1 y 4 |
| Creación del proyecto en GIT | Creación del proyecto en GIT y posterior cargue a la nube en la plataforma GitHub. | 1 hora | Integrante 1 |

Fuente: autores del proyecto

## Backlog Sprint 3

Tabla 4: Descripción 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 | Integrante 2 y 3 |
| 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 | Integrante 4 |
| 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 | Integrante 2 y 3 |
| 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 | Integrante 4 |

Fuente: autores del proyecto

## Backlog Sprint 4

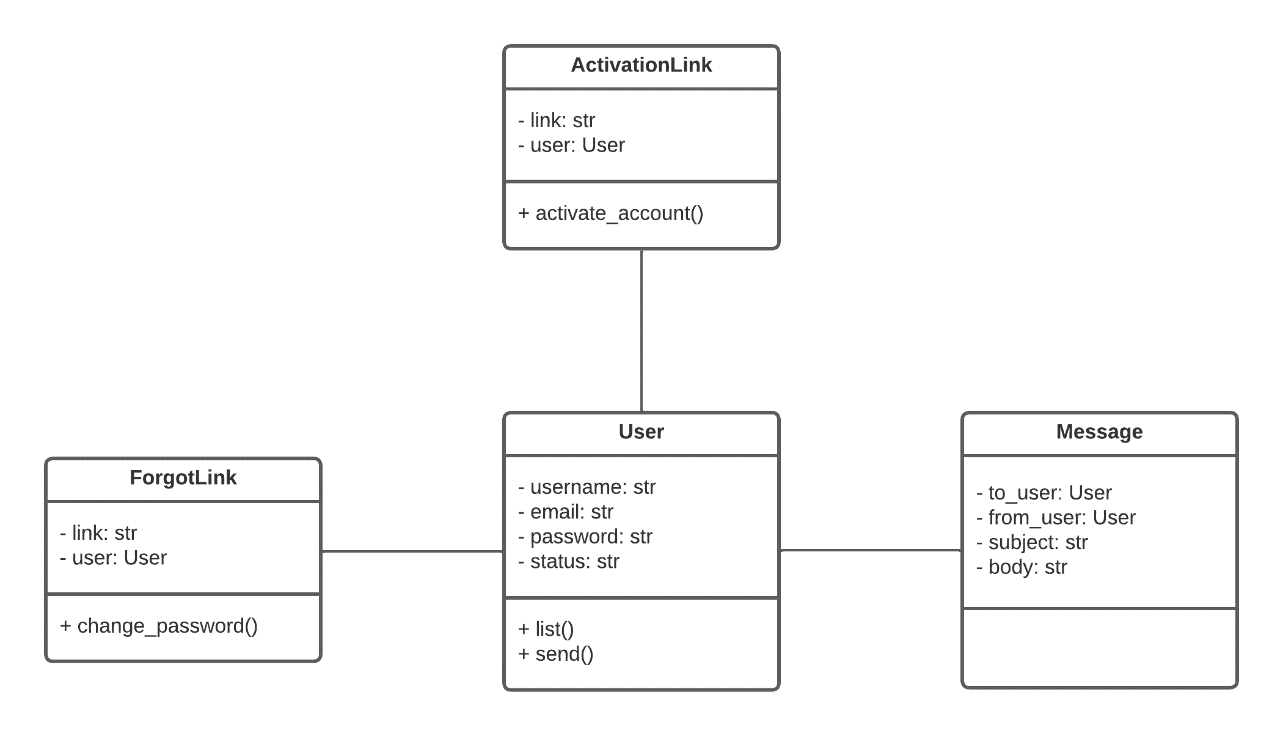
Tabla 5: Descripción Backlog Sprint 4

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | Todos los integrantes |
| Despliegue de la aplicación | Configuración, despliegue y verificación del funcionamiento de la aplicación en la plataforma PythonAnywhere | 12 horas | Todos los integrantes |

Fuente: autores del proyecto

# Diagrama de clases (*Sprint 1*)

Imagen 1: Diagrama de clases

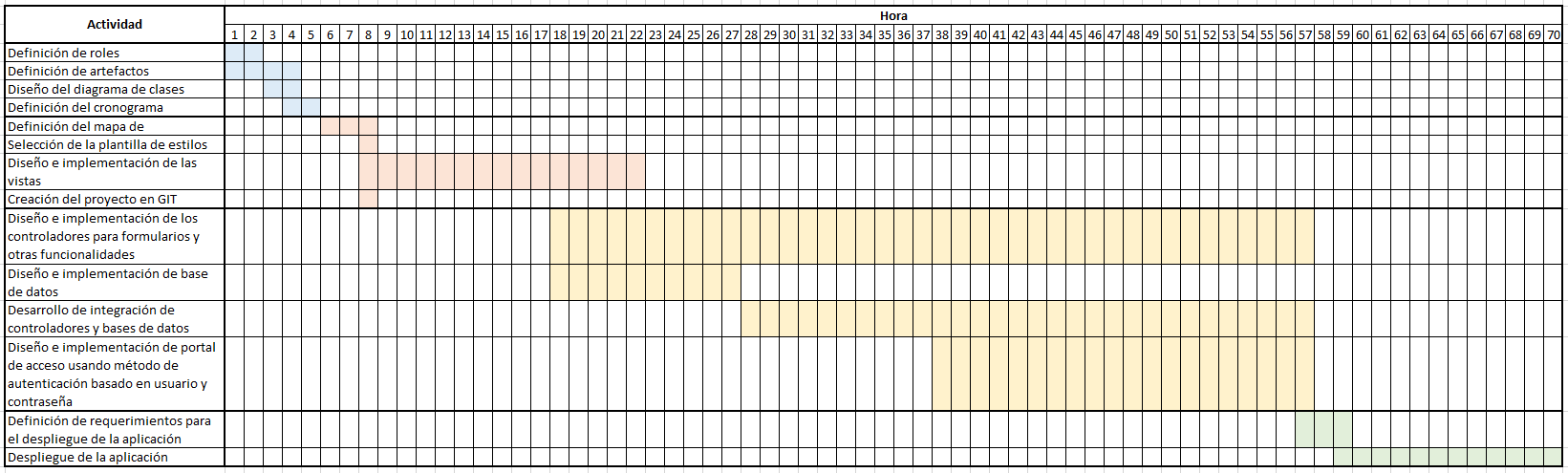


Fuente: autores del proyecto

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 (*Sprint 1*)

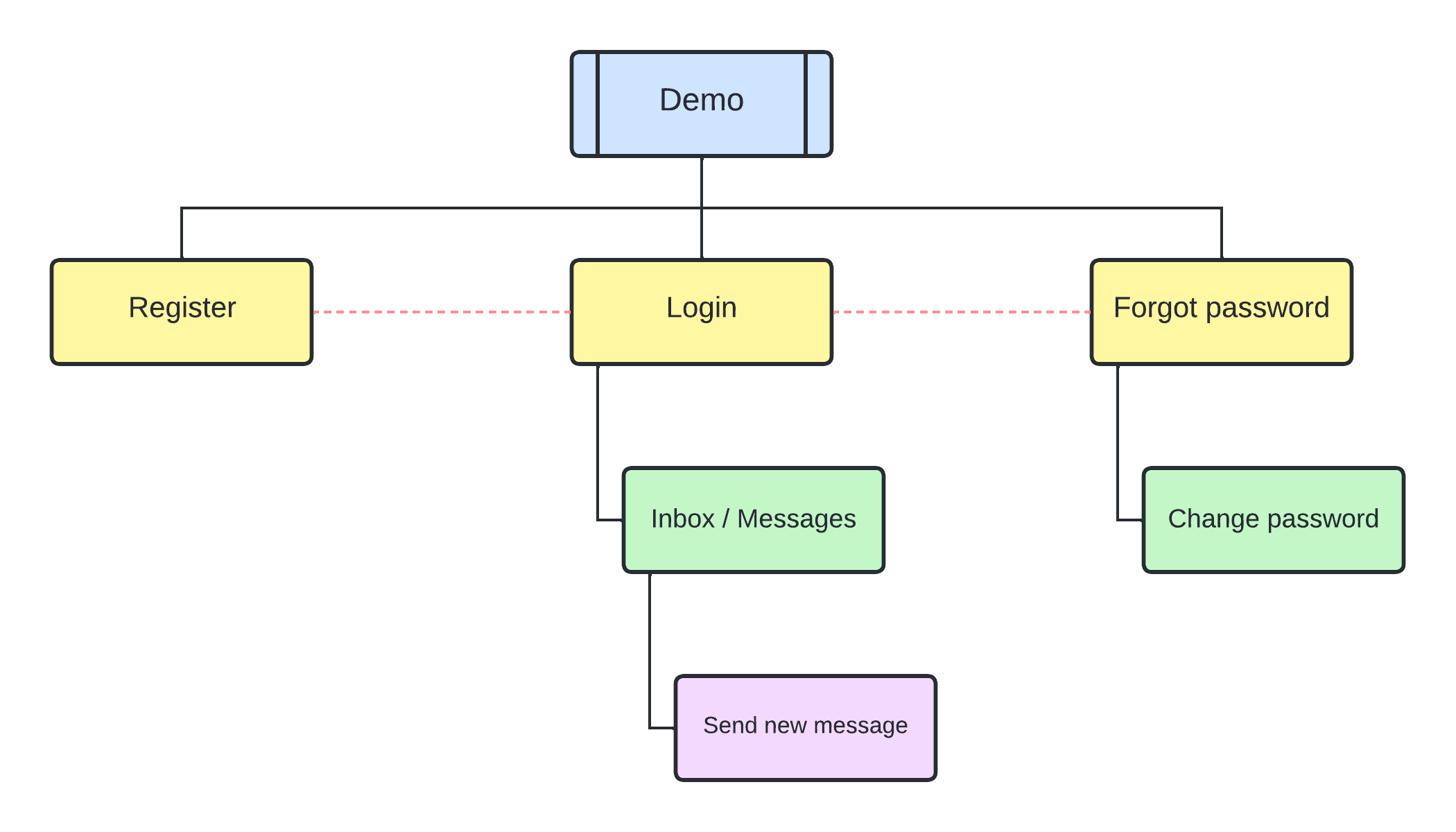
Imagen 2: Cronograma de tareas del proyecto



Fuente: autores del proyecto

# Mapa de navegabilidad (*Sprint 2*)

Imagen 3: Diagrama de clases



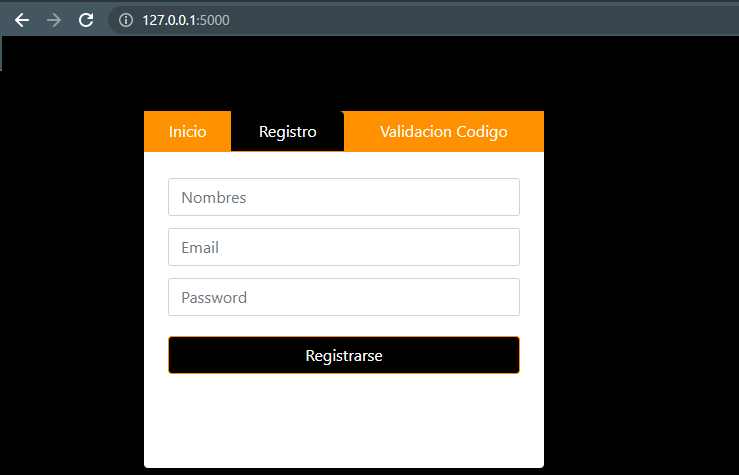
Fuente: autores del proyecto

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 (*Sprint 2*)

## Vista de registro

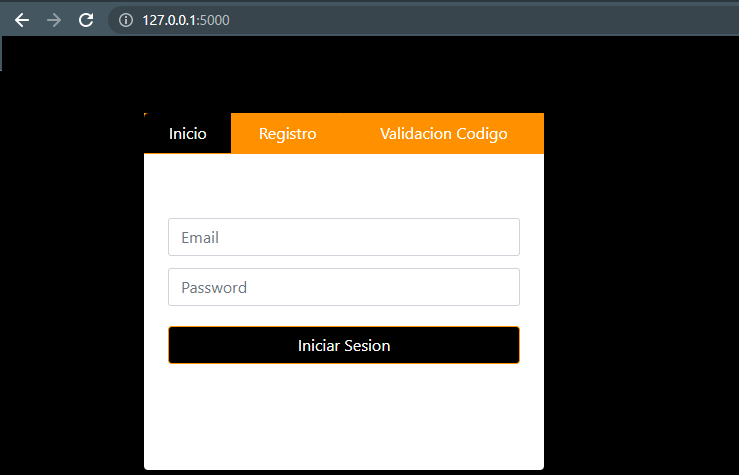
Imagen 4: Vista de registro



Fuente: autores del proyecto

## Vista de inicio de sesión

Imagen 5: vista inicio de sesión



Fuente: autores del proyecto

## Vista de validación de código

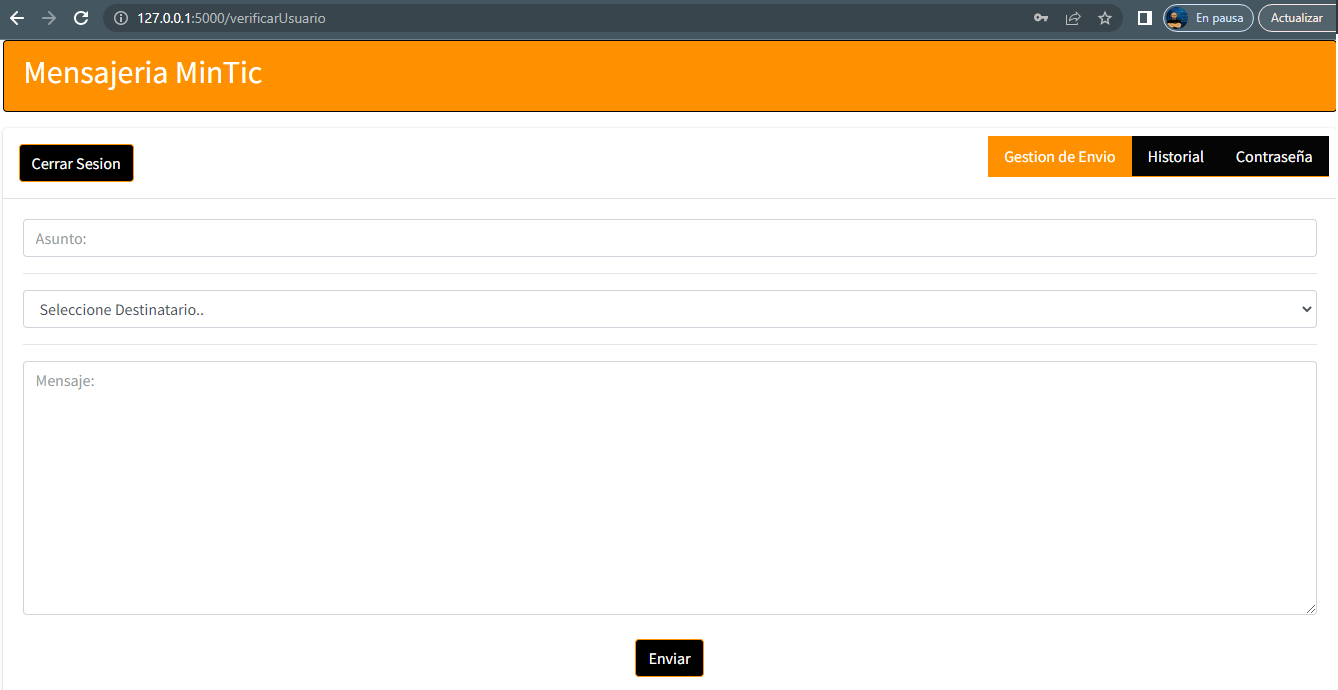
Imagen 6: Validación de código



Fuente: autores del proyecto

## Vista de envío de mensajes

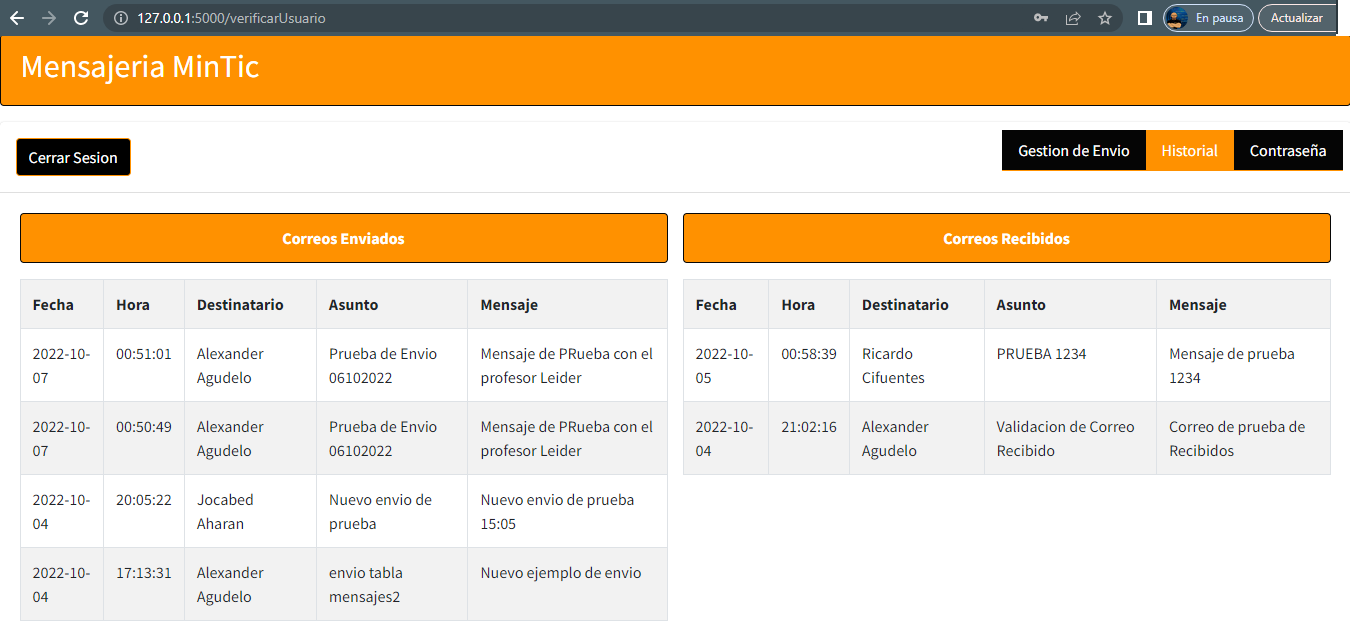
Imagen 7: Envío de mensajes



Fuente: autores del proyecto

## Vista de visualización de mensajes

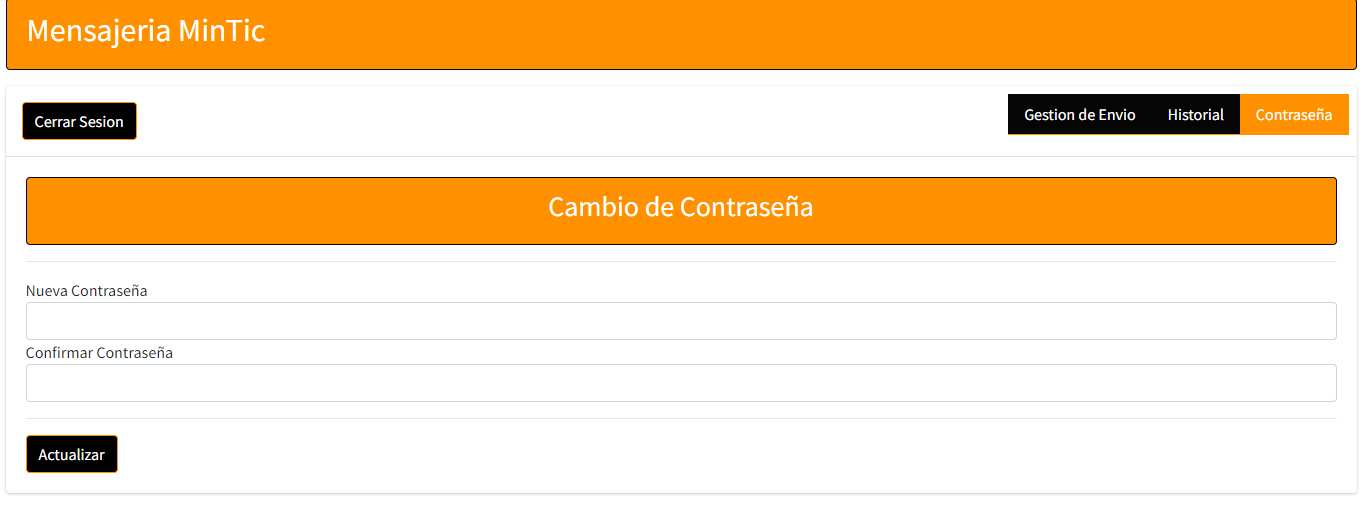
Imagen 8: Visualización de mensajes



Fuente: autores del proyecto

## Vista de cambio de contraseña

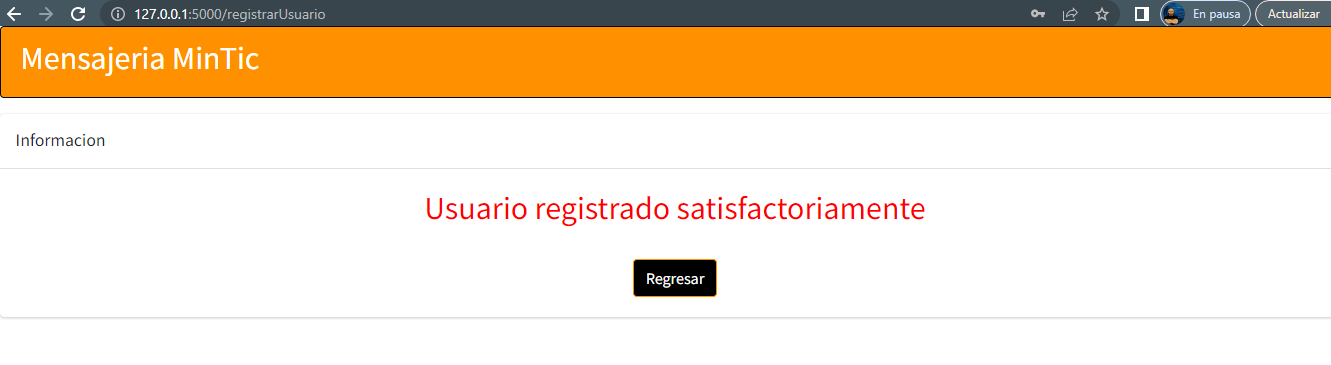
Imagen 9: Cambio de contraseña



Fuente: autores del proyecto

## Vista de usuario registrado

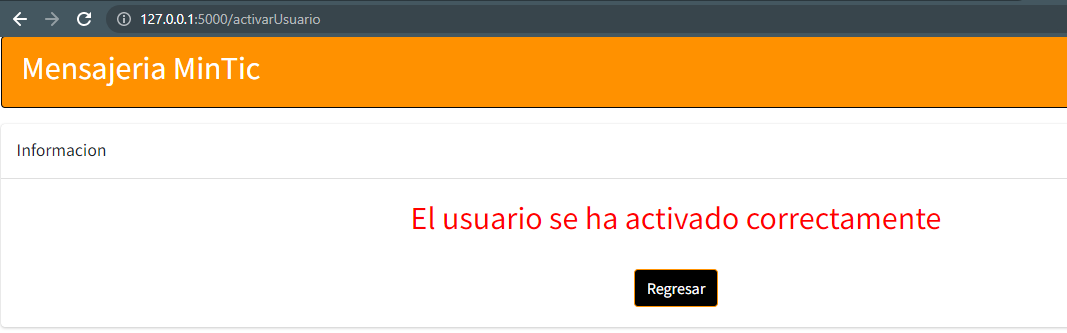
Imagen 10: Usuario registrado

****

Fuente: autores del proyecto

## Vista de activación de usuario

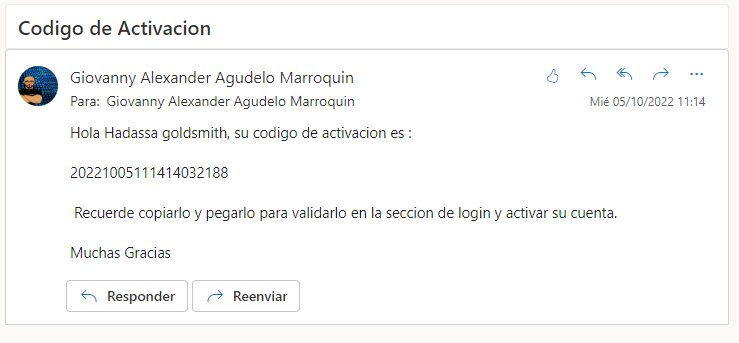
Imagen 11: Activación de usuario

****

Fuente: autores del proyecto

## Vista de mensaje de código de activación

Imagen 12: Activación de usuario

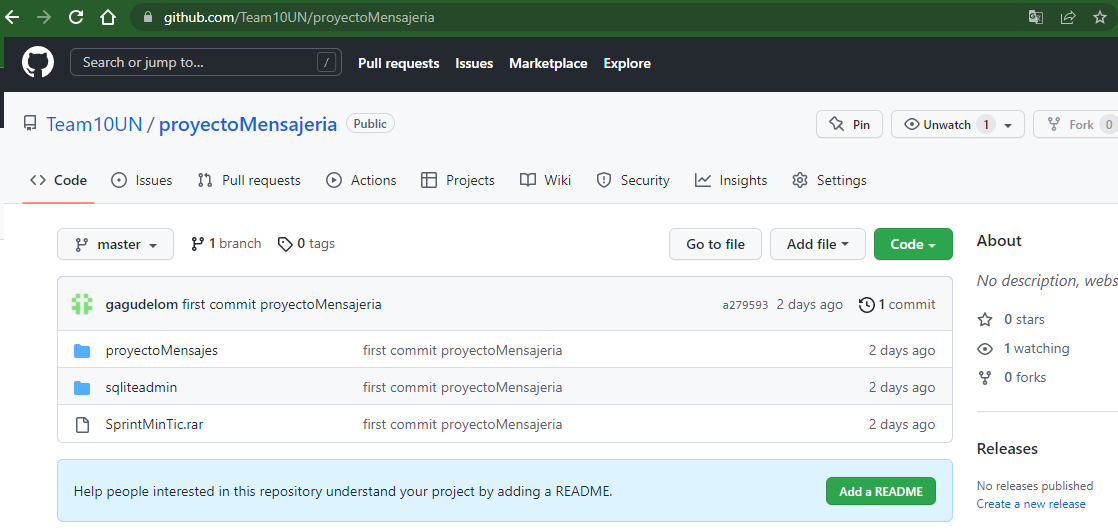


Fuente: autores del proyecto

## Vista de plataforma GIT

Para poder acceder a la plataforma GitHub y descargar el repositorio del proyecto, se recomienda seguir el siguiente enlace: <https://github.com/Team10UN/SpringProyectoMensajeria>

Imagen 13: Vista repositorio GitHub

****

Fuente: autores del proyecto

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

Tabla 6: Descripción métodos controladores Back-End

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Ruta | Métodos HTTP permitidos | Lógica algorítmica |
| registrarUsuario | **/registrarUsuario** | **POST** | **Ver sección siguiente** |
| enviarMAIL | **/enviarMAIL** | **POST** |
| activarUsuario | **/** **activarUsuario** | **POST** |
| registrarUsuario | **/registrarUsuario** | **POST** |
| historialEnviados | **/historialEnviados** | **POST** |
| historialRecibidos | **/historialRecibidos** | **POST** |
| actualizarPassword | **/actualizarPassword** | **POST** |

Fuente: autores del proyecto

## RegistrarUsuario

* + Ruta: /registrarUsuario
  + Métodos HTTP permitidos: POST
  + Lógica algorítmica:

A través de los campos colocados en HTML en la pestaña registrar se ingresa la información del usuario y se encripta con el método sha384 de la librería hashlib de Python, internamente también se genera un código de activación para ser enviados en conjunto con la información de usuario a la base de datos de la plataforma. Posteriormente se envía un correo al usuario con el código de activación para ser ingresado en la sección validar cuenta y así poder cambiar el estado de la cuenta ha activado en la base de datos.

## EnviarMAIL

* + Ruta: /enviarMAIL
  + Métodos HTTP permitidos: POST
  + Lógica algorítmica:

A través de los campos colocados en HTML en la página principal se ingresa la información de contacto del usuario al cual se le va a enviar un mensaje para ser registrada en la base de datos mediante un método registrarMail en el script controlador y posteriormente enviar un correo electrónico al usuario alertando la llegada de un mensaje en la plataforma

## ActivarUsuario

* + Ruta: /activarUsuario
  + Métodos HTTP permitidos: POST
  + Lógica algorítmica:

A través de los campos colocados en HTML en la pestaña validar el usuario ingresa el código recibido vía email para ser consultado internamente en la base de datos y validarla coincidencia de este código y cambiar el estado a activado en la base de datos y así permitir ingreso al usuario.

## HistorialEnviados

* + Ruta: /historialEnviados
  + Métodos HTTP permitidos: POST
  + Lógica algorítmica:

A través de un método en el script controlador se enviá el email del usuario activo para consulta todos los correos en la base de datos de mensajes donde el correo de origen es igual al correo del usuario, este listado de correos se transfiere a la pestaña de historial en la página principal y así presentar la información solicitada en una tabla organizada.

## HistorialRecibidos

* + Ruta: / historialRecibidos
  + Métodos HTTP permitidos: POST
  + Lógica algorítmica:

A través de un método en el script controlador se enviá el email del usuario activo para consulta todos los correos en la base de datos de mensajes donde el correo de destino es igual al correo del usuario, este listado de correos se transfiere a la pestaña de historial en la página principal y así presentar la información solicitada en una tabla organizada.

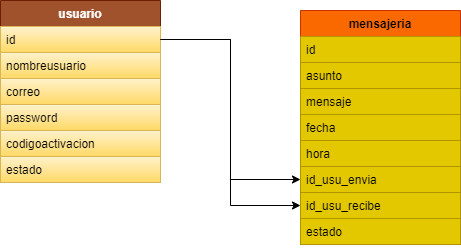
## ActualizarPassword

* + Ruta: / actualizarPassword
  + Métodos HTTP permitidos: POST
  + Lógica algorítmica:

A través de un formulario en la pestaña contraseña en la página principal el usuario puede ingresar su nueva contraseña y verificar nuevamente el campo de contraseña con el fin de encriptar la contraseña con el método has384 y enviar esta información a través de un método del script controlador que actualiza la contraseña en la base de datos de usuarios

# Base de datos (*Sprint 3*)

Imagen 14: Tablas y relaciones- BD proyecto



Fuente: autores del proyecto

La base de datos contiene 2 tablas principales: usuario y mensajería.

La tabla “usuario” contiene los siguientes campos: id (llave principal, int), nombreusuario, correo, password (encriptado en la BD bajo HASH 384), codigoactivacion (enviado por plataforma), y estado (manejado internamente). Salvo el campo id, todos son VARCHAR y su tamaño varía en función de la cantidad de información que deban albergar.

La tabla “mensajería” contiene: id (llave principal, int), asunto, mensaje, fecha, hora, id\_usuario\_envia, id\_usuario\_recibe, estado.

Salvo el campo id, todos son VARCHAR y su tamaño varía en función de la cantidad de información que deban albergar. El campo id de “usuario” es el relacionado con la tabla mensajería, en forma “uno a varios”.

# Prácticas de programación segura (*Sprint 3*)

Aproximadamente 50% de todos los errores (o vulnerabilidades) ocurren a nivel del código de programación (Shahriar y Zulkernine 2012).

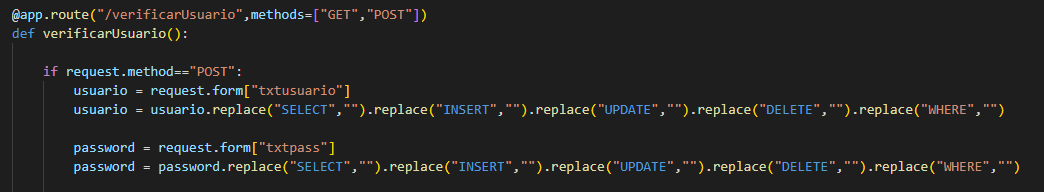
Con el enfoque de programación segura se pretende proporcionar apoyo para la implementación de programas libres de vulnerabilidades, la cual puede ser considerada como la primera línea de defensa para evitar brechas en la seguridad del programa.

De acuerdo con las vulnerabilidades investigadas (The mitre corporation, 2011), se consideraron aquellas que pueden afectar a las aplicaciones desarrolladas, entre las cuales se destacan las siguientes amenazas en el desarrollo: SQL INJECTION, o COMMAND INJECTION, la falta de autenticación con un usuario y contraseña autorizados o la asignación de permisos incorrectos.

## SQL injection

Para evitar el SQL injection, nuestro desarrollo presenta un bloqueo en los campos de ingreso de datos, por lo cual evitamos la inyección de comandos SQL que puedan afectar directamente nuestra base de datos, con información que pueda alterar, agregar o eliminar información directamente en nuestra base de datos.

Imagen 15: Modelo protección contra SQL Injection



Fuente: autores del proyecto

## Encriptación de contraseñas

Para evitar que la información de la contraseña de los usuarios del sistema pueda ser clonada o visualizada por usuarios o personas no autorizadas, se realizó una encriptación de los datos que se van agregando en el campo “password” correspondiente a la tabla de usuarios de la base de datos del sistema.

En la imagen a continuación, se distorsiona el campo “correo” por motivos de seguridad.

Imagen 16: Encriptación contraseñas

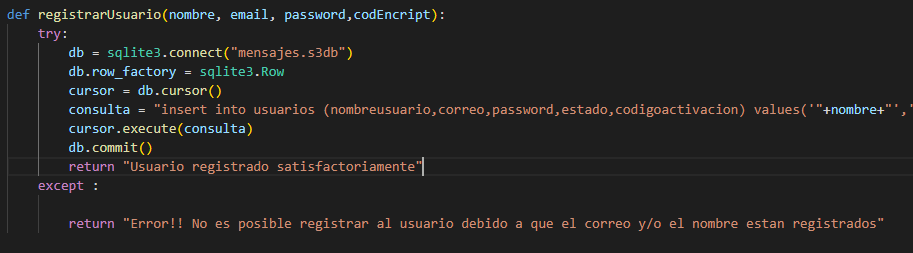


Fuente: autores del documento

## Excepción de errores

Para evitar que los usuarios vean los errores que se puedan presentar durante la ejecución del programa, y por medio de esta información se pueda realizar algún proceso de manipulación de información o validación de clases o servidores incluidos en el desarrollo, se realizo un control de excepciones en el sistema, permitiendo que los usuarios puedan visualizar la información contenida dentro de las excepciones controladas en el sistema.

Imagen 17: Modelo excepción de errores



Fuente: autores del proyecto

# Despliegue de la aplicación en la nube (*Sprint 4*)

## Requerimientos

*Los requerimientos del sistema para el correcto funcionamiento del aplicativo son los siguientes:*

*FLASK 2.2.2*

*Python 3.10.6*

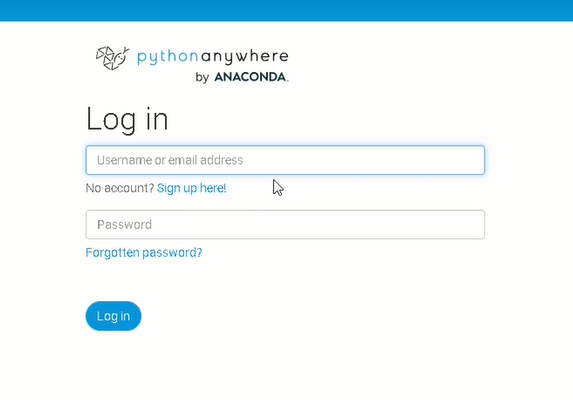
*Werkzeug 2.2.2*

## Proceso

Se realizaron los siguientes pasos para el despliegue:

1. Se ingresa a la página www.pythonanywhere.com y en la ventana principal ingresamos el usuario y contraseña

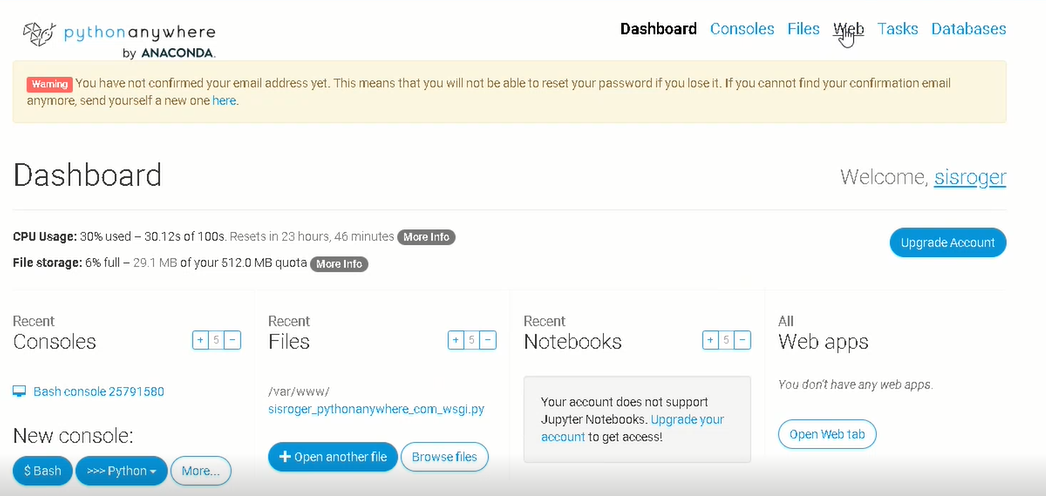
Imagen 18: Login plataforma pythonanywhere.com



Fuente: autores del proyecto

1. Nos dirigimos a la pestaña WEB dentro de la plataforma para crear nuestro sitio en línea.

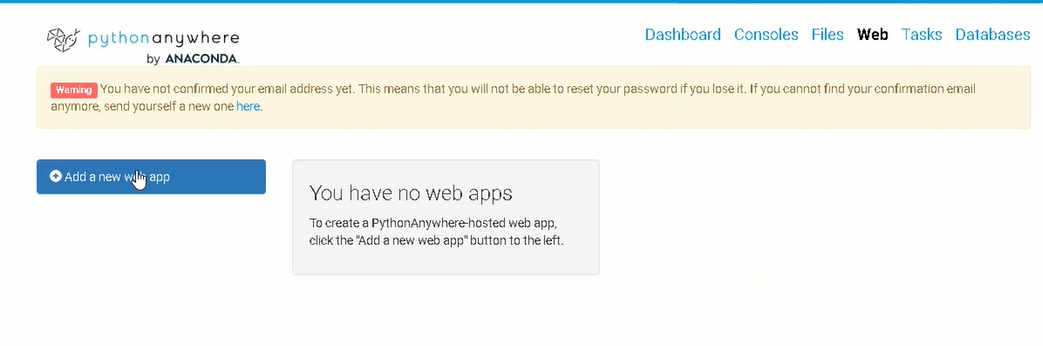
Imagen 19: Ingreso plataforma web



Fuente: autores del proyecto

1. Damos clic en adicionar un nuevo proyecto

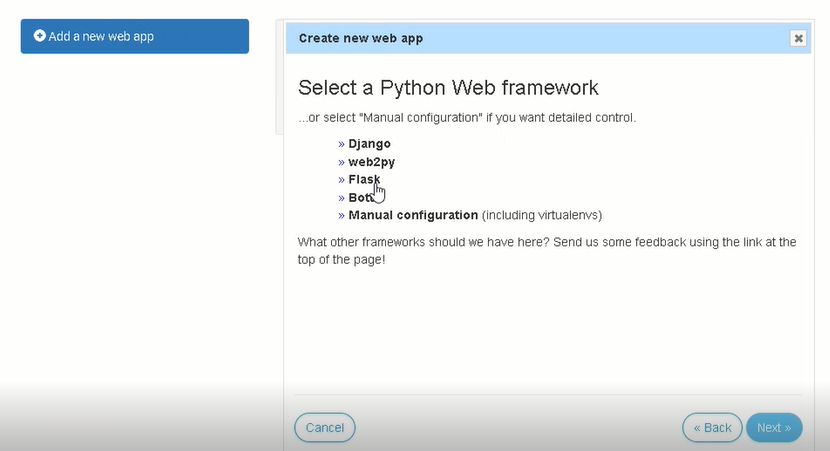
Imagen 20: Adicionar nueva app



Fuente: autores del proyecto

1. Damos clic en siguiente y seleccionamos la configuración manual de tipo FLASK

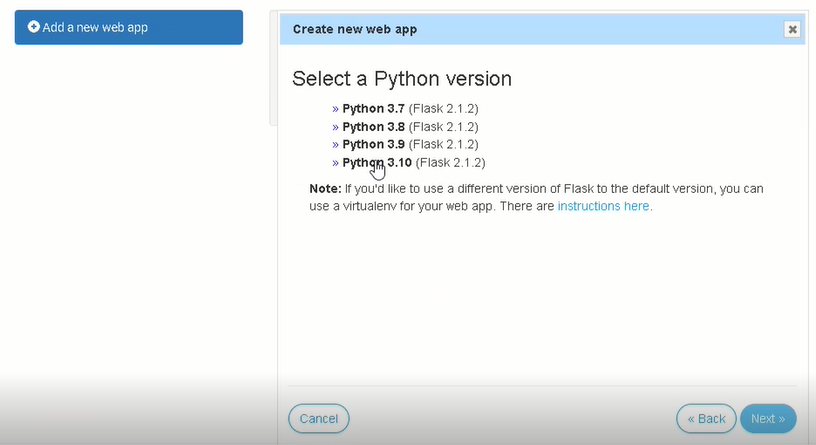
Imagen 20: Selección de FLASK



Fuente: autores del proyecto

5.Seleccionamos la versión de Python que se usó en el desarrollo

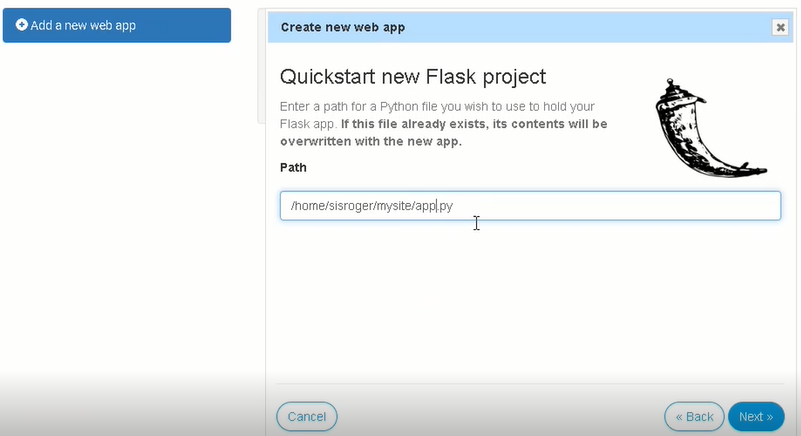
Imagen 21: Selección de versión Python



Fuente: autores del proyecto

1. Creamos nuestro sitio web básico.

Imagen 22: Nombre proyecto y app principal

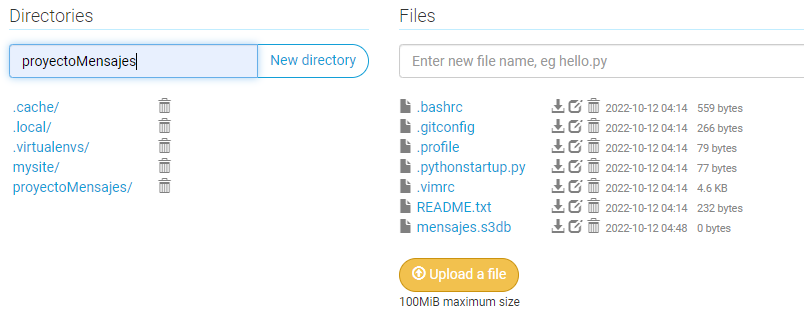


**/home/team10/mysite/app.py**

Fuente: autores del proyecto

1. Creamos el directorio **proyectoMensajes** donde van a estar los diferentes archivos del nuestro proyecto

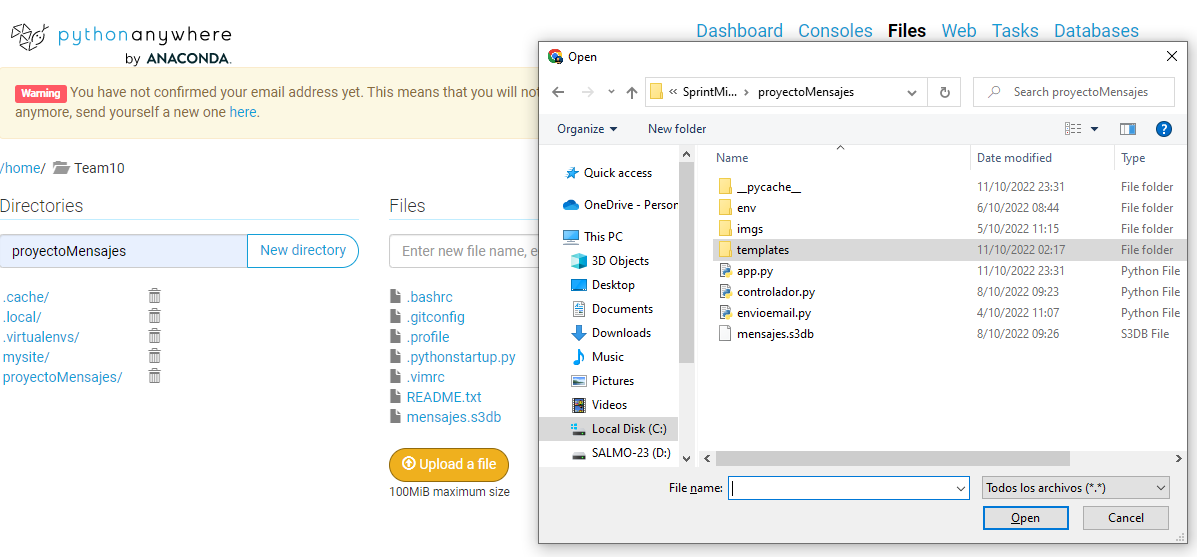
Imagen 23: Creación de directorio



Fuente: autores del proyecto

1. Cargamos los archivos en la plataforma del servidor como lo son: base de datos, archivos .py y archivos .html

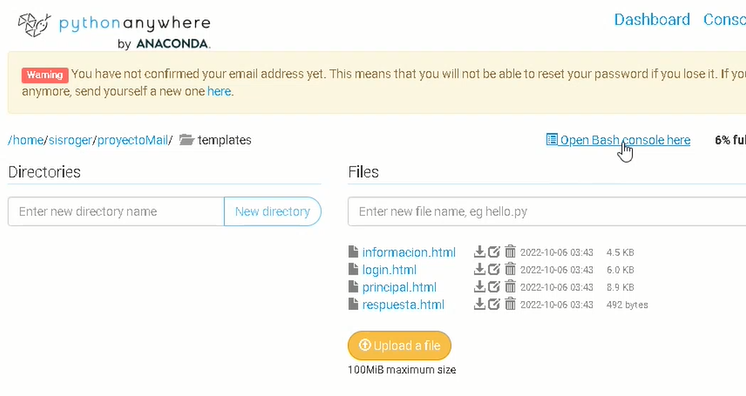
Imagen 24: cargue de archivos



Fuente: autores del proyecto

1. Abrimos la consola OPEN BATCH para descargar FLASK y habilitar el entorno virtual en línea.

Imagen 25: Open Batch

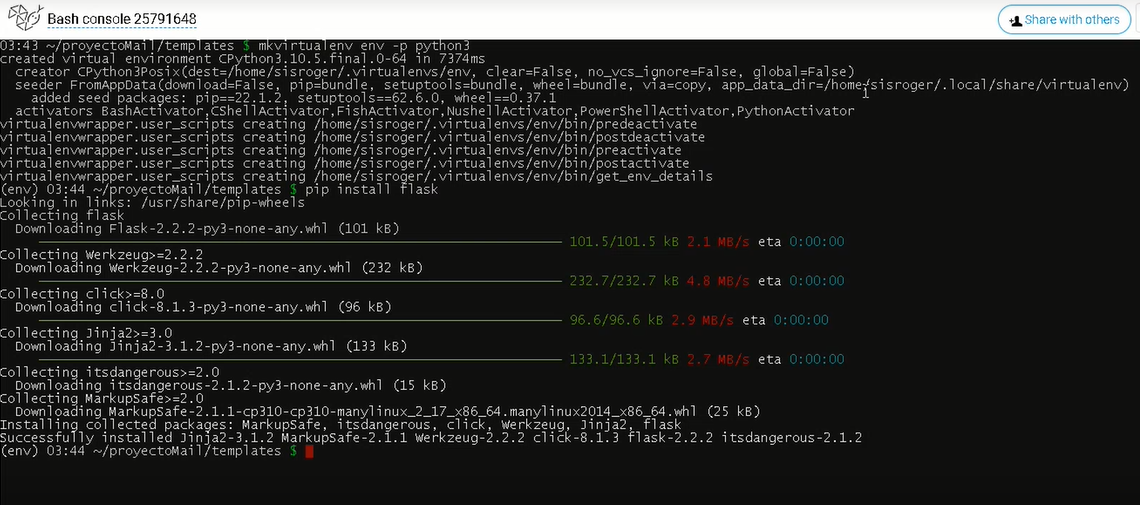


Fuente: autores del proyecto

1. Dentro de la consola escribimos los siguientes comandos:

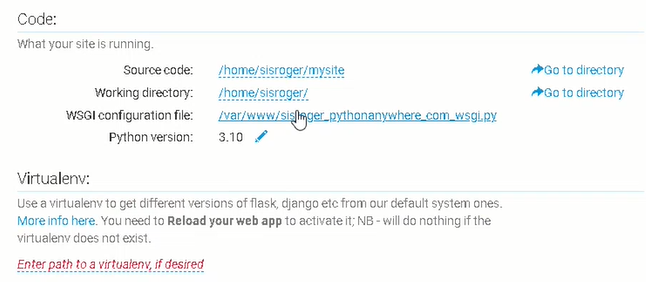
* **mkvirtualenv env -p python3** (Creamos nuestro entorno virtual)
* **pip install flask** (Para instalar flask dentro del servidor)

Imagen 26: Consola Linux del servidor



1. En la sección **CODE** cambiamos la configuración **WSGI** para que la ruta sea la correcta cuando se llama el desarrollo en línea.

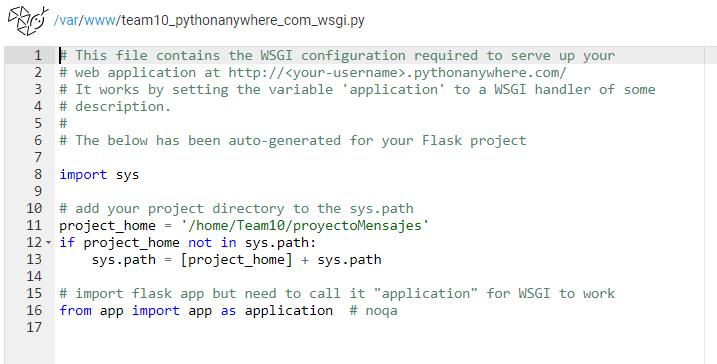
Imagen 27: Enrutamiento aplicación



Fuente: autores del proyecto

1. Redireccionamos la ruta en el sistema: **/home/Team10/proyectosMensajes**

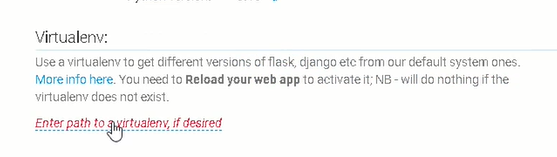
Imagen 28: redirección de ruta



Fuente: autores del proyecto

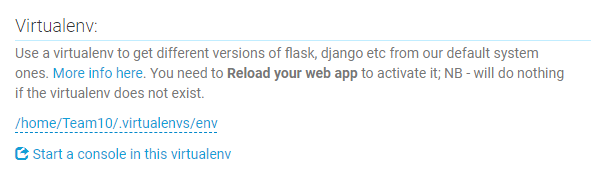
1. En la sección **VIRTUALENV** creamos el entorno virtual dentro de la plataforma para que el proyecto pueda cargar en línea sin ninguna complicación

Imagen 29: redirección de ruta



1. Asignamos la ruta del proyecto creado

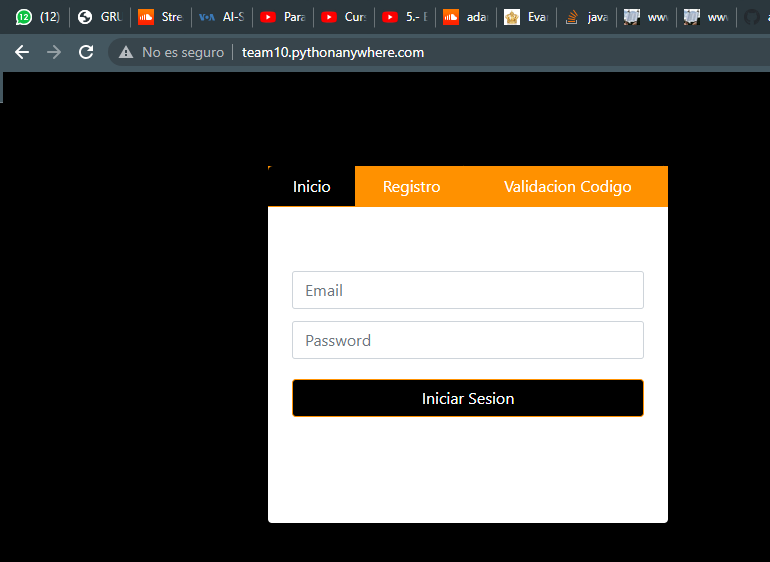
Imagen 30: asignación de ruta



Fuente: autores del proyecto

1. Se actualiza el servidor y validamos el cargue del proyecto bajo la ruta asignada en línea

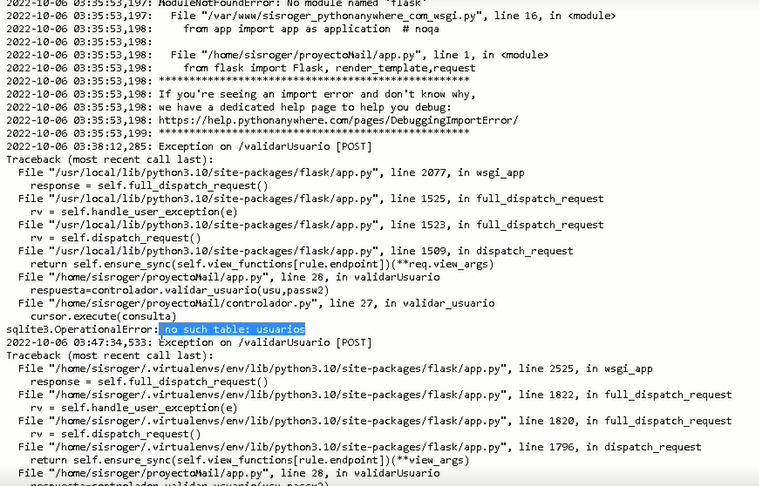
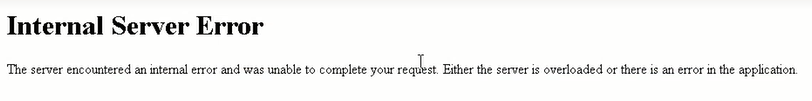
Imagen 31: vista login



Fuente: autores del proyecto

1. Cuando se realiza un proceso de ingreso de usuario y contraseña, el sistema presenta el siguiente error 500 por problema de conectividad con la base de datos

Imagen 32: Error en la conexión de la BBDD con el servidor



Fuente: autores del proyecto

1. En la sección FILES dentro de la consola, abrimos el archivo controlador.py y redireccionamos las rutas de los procesos que se relacionan con la base de datos del proyecto

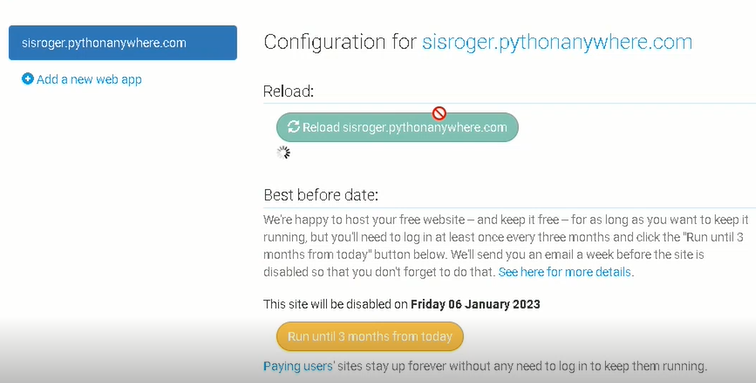
Imagen 33: Asignación de ruta para la BBDD



Fuente: autores del proyecto

1. Refrescamos el servidor para guardar los cambios realizados dentro del proyecto

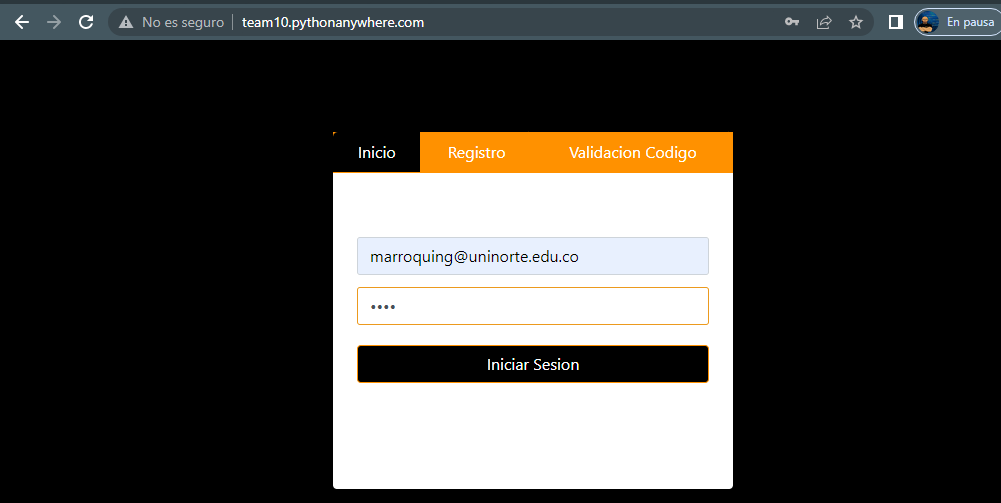
Imagen 34: Refresh para guardar cambios



Fuente: autores del proyecto

1. Realizamos pruebas de funcionamiento con el ingreso de un usuario autorizado

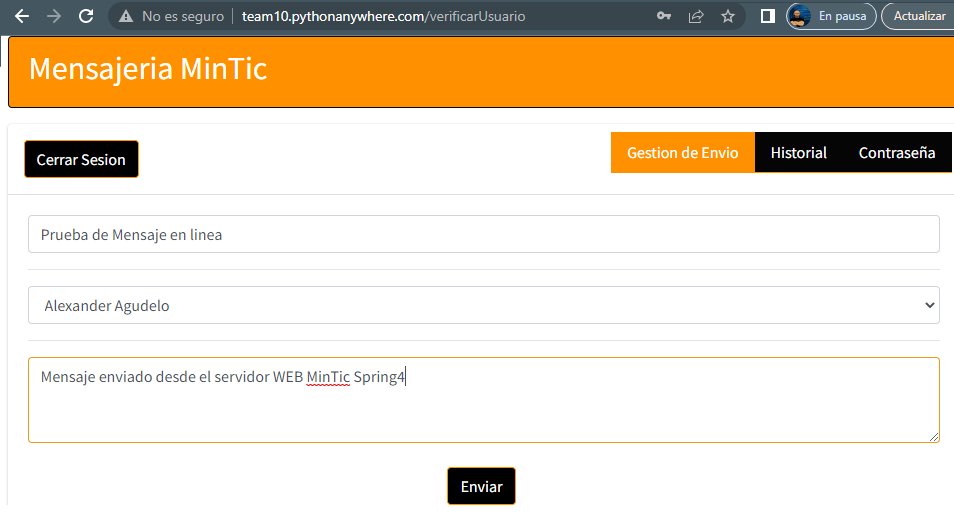
Imagen 35: Ingreso de usuario y contraseña



Fuente: autores del proyecto

1. Ingresamos a la sección de **Mensajería** y realizamos pruebas de funcionamiento

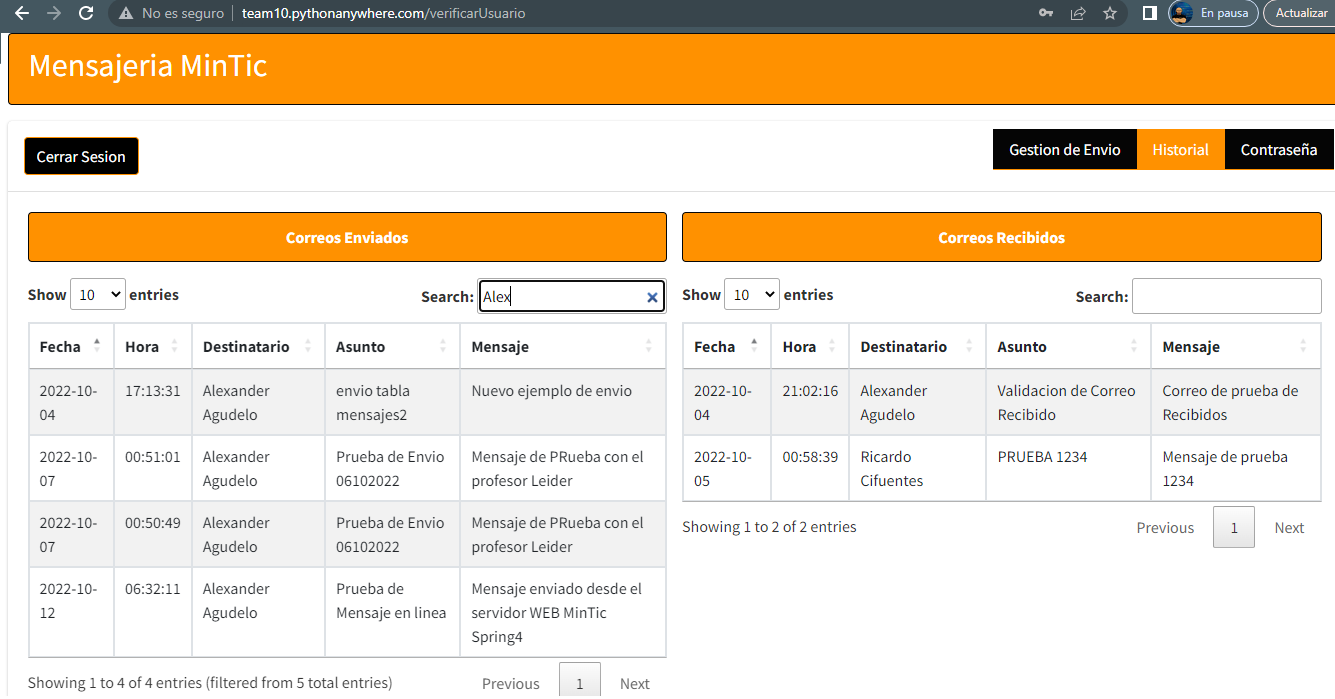
Imagen 36: Vista de mensajería



Fuente: autores del proyecto

1. Se recibe la información dentro de la sección de HISTORIAL don se validan los mensajes enviados y recibidos

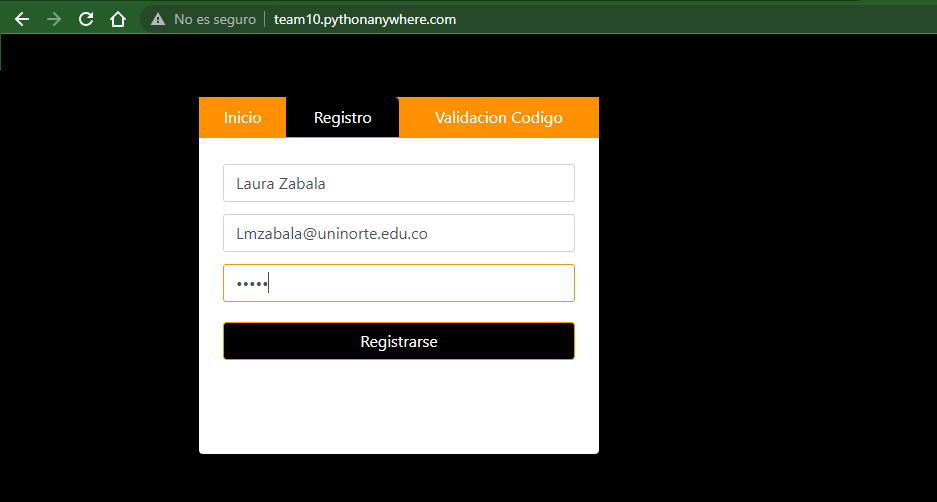
Imagen 37: validación de mensaje recibido



Fuente: autores del proyecto

1. Se hacen pruebas de registro de nuevo usuario en el sistema

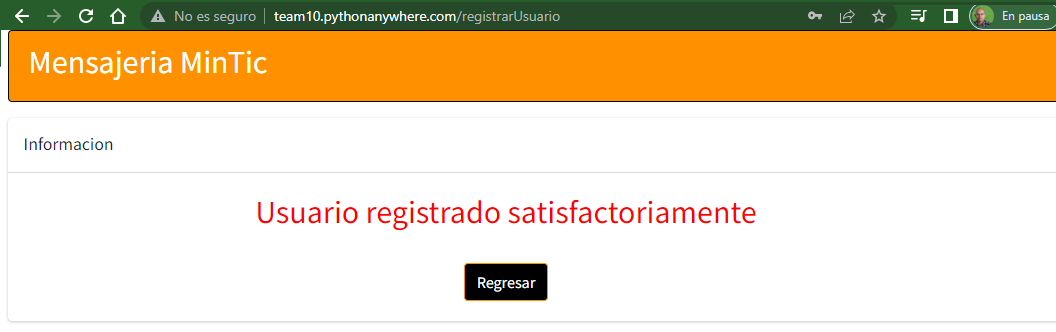
Imagen 38: Pruebas de registro de usuario



Fuente: autores del proyecto

1. Se visualiza una nueva ventana donde se informa que el usuario quedo registrado satisfactoriamente

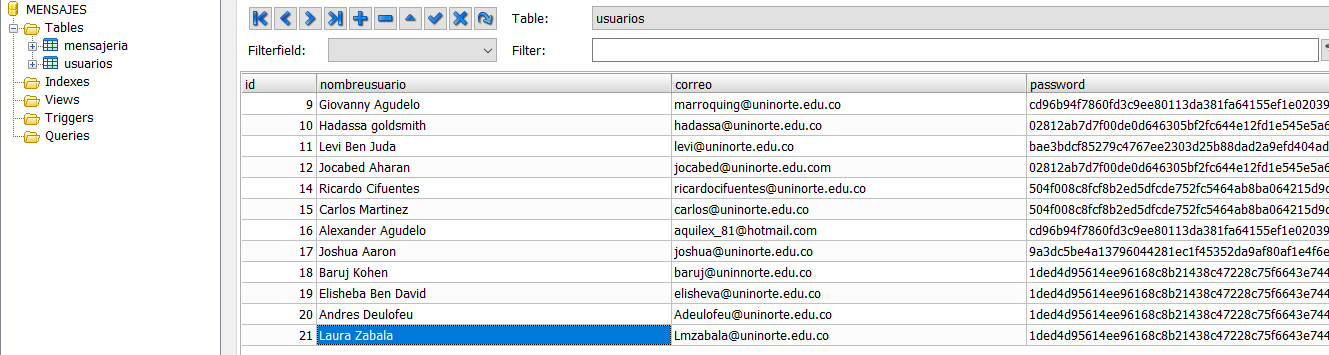
Imagen 39: Mensaje de registro



Fuente: autores del proyecto

1. Visualización del registro del nuevo usuario en la Base de datos

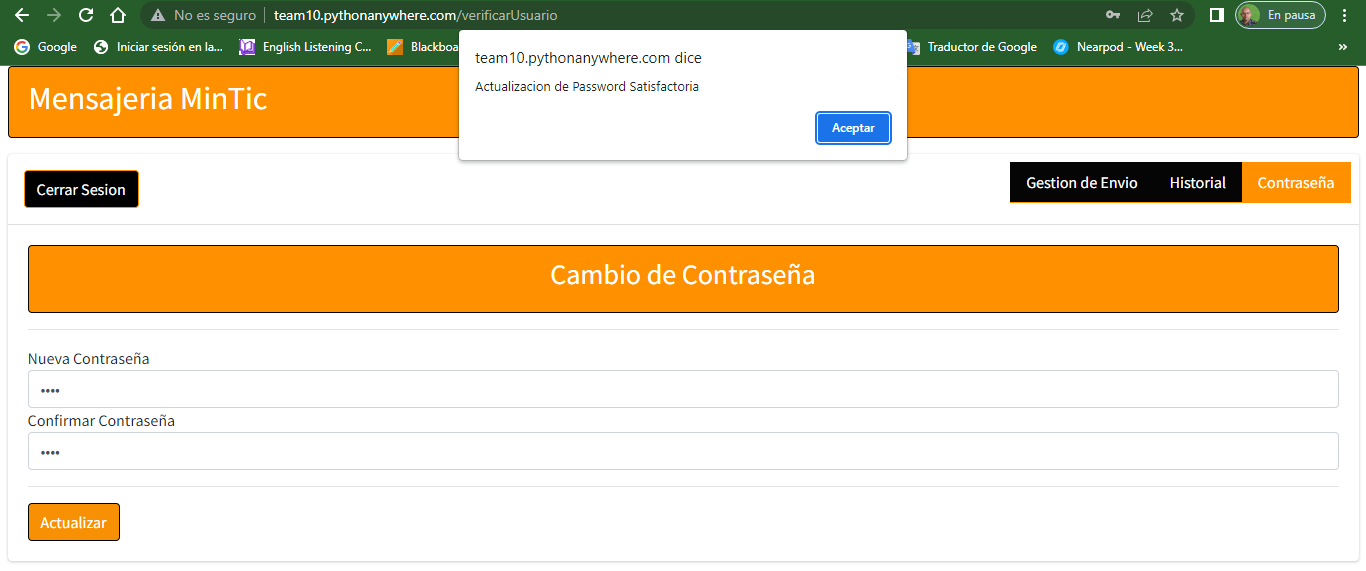
Imagen 40: Mensaje de registro



Fuente: autores del proyecto

1. Se visualiza la ventana de cambio de contraseña, la cual realiza el cambio de contraseña en el sistema sin ninguna novedad

Imagen 41: Formulario de cambio de contraseña



Fuente: autores del proyecto

# Enlace final de la aplicación (*Sprint 4*)

La información correspondiente al proceso de implementación se podrá consultar en el siguiente enlace: <http://team10.pythonanywhere.com/>