

Documentación Proyecto

Plataforma de Mensajería Electrónica

NRC: 2105

Equipo No. 11

Integrantes:

Fernando Andrés Palacio
Laura Alejandra Mendoza
Lindy Johana Vásquez
Manuel Eduardo Ortega
Luis Darío García

Descripción de roles del equipo

Rol	Integrante	Descripción	Tareas
Product Owner líder	Fernando	Lidera procesos y revisión de producto	<ul style="list-style-type: none"> • Objetivos de producto • Características y creación de Backlog • Creación de historias de usuario • Gestión Backlogs • Supervisión etapas de desarrollo
Diseñador	Lindy	Es la parte inicial en la arquitectura del software diseñando diagramas.	Levantamiento Requerimientos <ul style="list-style-type: none"> • Abstracción de datos • Diseño casos de uso • Diseño diagrama de clases.
Desarrollador Back-End	Laura Alejandra	Se va a trabajar en Python y framework Flask.	<ul style="list-style-type: none"> • Codificación en Visual Code • Desarrollo de clases y métodos • Pruebas unitarias
Desarrollador Back-End	Luis	Desarrollo de la parte visual en HTML y CSS	<ul style="list-style-type: none"> • Estructura en HTML • Seguir parámetros de diseño según requerimientos
Desarrollador Full-Stack	Manuel Eduardo	Conexión con la base de datos y	<ul style="list-style-type: none"> • Creación de base de datos

		realización de prueba de consulta y la lógica de negocios de la aplicación.	en sqlite <ul style="list-style-type: none">● Aplicar CRUD● Realizar pruebas de conexión● Verificar consultas
--	--	--	---

Definición de artefactos

Backlog Sprint 1

User story	Descripción	Estimación	Responsable
Start	Lectura y análisis del problema	1 horas	Equipo
Plann1	Planificación Backlogs	4 horas	Fernando
Plann2	Definición Requisitos Funcionales	12 horas	Alejandra
Plann3	Definición Requisitos No Funcionales	12 horas	Lindy
Meeting	Reunión Equipo	2 horas	Equipo
Plann4	Desarrollo Casos de Uso	8 horas	Luis, Manuel
Plann5	Desarrollo Diagrama de Clases	12 horas	Equipo
Plann6	Desarrollo de Cronograma	4 horas	Fernando Palacio
Rev1	Check Documentación	2 horas	Equipo

Backlog Sprint 2

User story	Descripción	Estimación	Responsable
Cod1	Definición de las Clases Principales	8 horas	Alejandra
Cod2	Mockup gráfico Interfaz	8 hora	Luis
Cod3	Codificación de las Clases en Python	24 horas	Alejandra

Cod4	Realizar Pruebas Unitarias de Código	48 hora	Aleja y Lindy
Rev2	Revisión de Código	24 horas	Fernando y Manuel
Meeting2	Reunión de Equipo	2 hora	Equipo
Cod5	Estructuración y diseño web en HTML, CSS	48 horas	Luis
Cod6	Solución de Errores	24 hora	Equipo

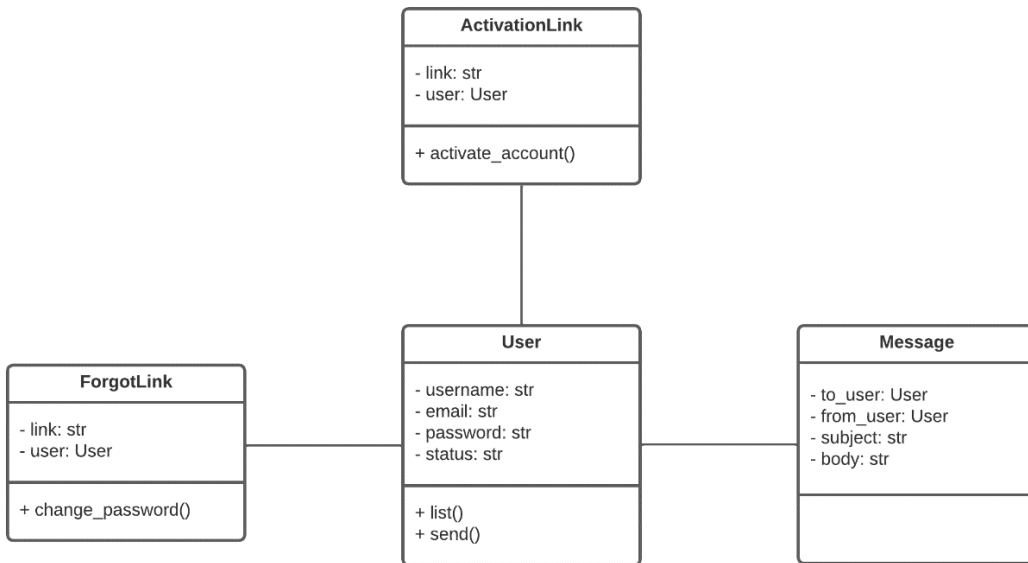
Backlog Sprint 3

User story	Descripción	Estimación	Responsable
Cod7	Creación de Bases de Datos	8 horas	Manuel
Cod8	Realizar CRUD	8 horas	Manuel
Cod9	Establecer Conexión con Software	8 horas	Manuel
Cod10	Realizar Pruebas	8 horas	Manuel, Luis
Cod11	Realizar Testeo	24 horas	Equipo
Cod12	Corrección de Bugs	8 horas	Manuel, Luis, Aleja

Backlog Sprint 4

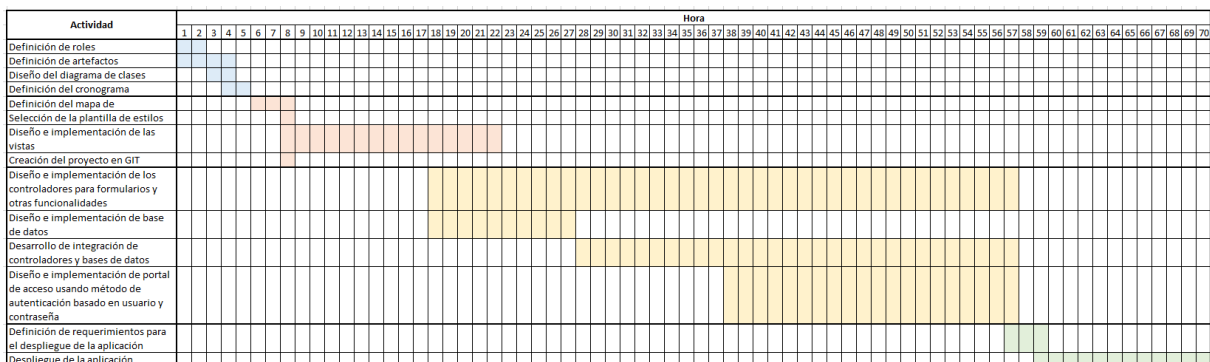
User story	Descripción	Estimación	Responsable
Rev3	Documentación (Manual User)	36 horas	Lindy
Rev4	Revisión Software	48 horas	Fernando
Final Rev	Mantenimiento 6 Mese		Alejandra, Luis, Manuel

Diagrama de clases

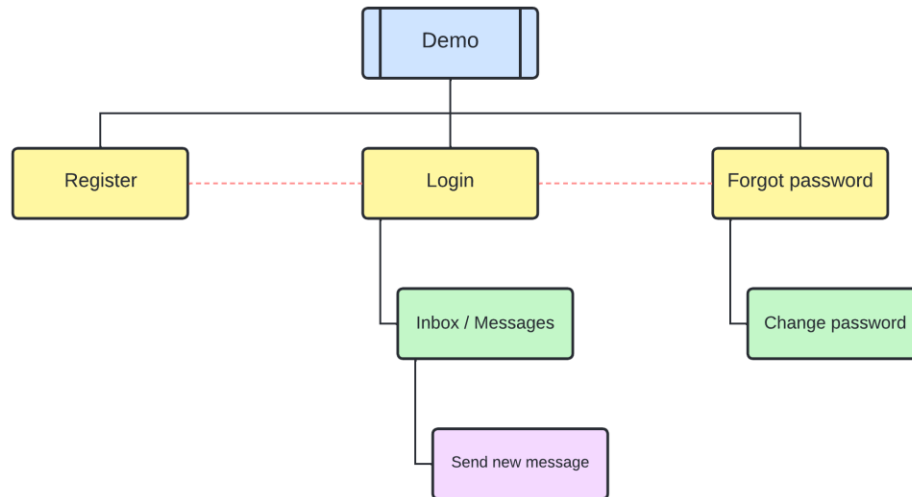


La clase usuario tendrá la información referente al usuario, es decir, su nombre de usuario, su correo, su contraseña y el estado de su cuenta. La clase mensaje tendrá la información de los mensajes de la plataforma, es decir, el usuario que envía el mensaje, el usuario que lo recibe, el asunto y el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

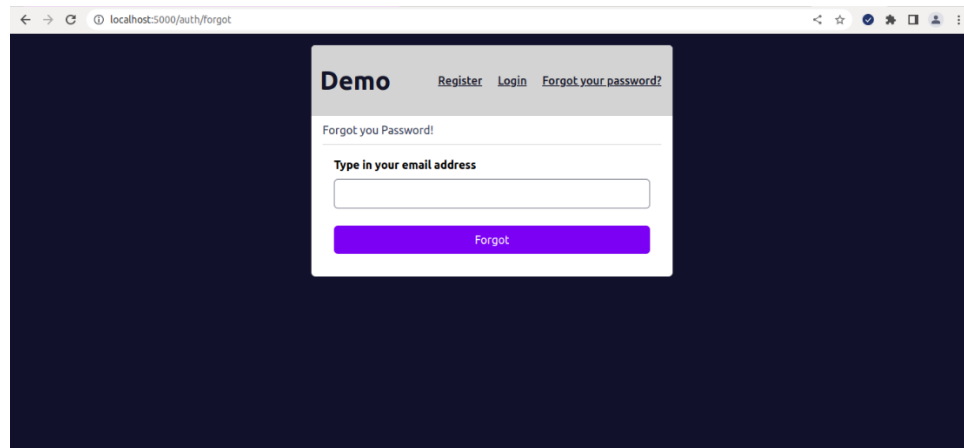
- **Vista de registro:**

The screenshot shows a web browser window with the address bar displaying 'localhost:5000/auth/register'. The page has a dark blue background. In the center, there is a white card titled 'Demo' with a navigation bar containing links for 'Register', 'Login', and 'Forgot your password?'. Below the navigation bar, the 'Register' form is displayed. It includes three input fields: 'Username', 'Email Address', and 'Password'. At the bottom of the form is a red button labeled 'Register'.

- **Vista de inicio de sesión:**

The screenshot shows a web browser window with the address bar displaying "localhost:5000/auth/login". The page features a dark blue background. In the center, there is a white login form. At the top of the form, the word "Demo" is displayed in a large, bold font. To the right of "Demo", there are three links: "Register", "Login", and "Forgot your password?". Below these links, the word "Login" is written in a smaller font. The form contains two input fields: one for "Username" and one for "Password". At the bottom of the form, there is a blue button with the text "Log in".

- Vista de solicitud de recuperación de contraseña:



localhost:5000/auth/forgot

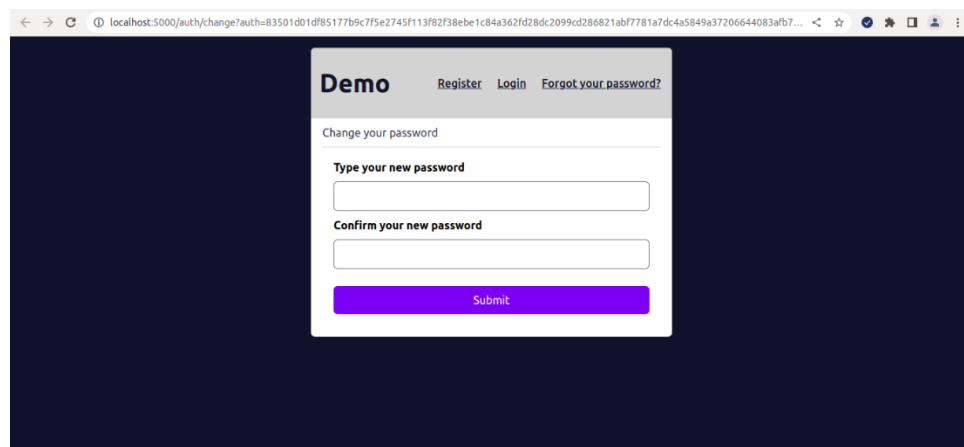
Demo [Register](#) [Login](#) [Forgot your password?](#)

Forgot you Password!

Type in your email address

Forgot

- Vista de cambio/recuperación de contraseña:



localhost:5000/auth/change?auth=83501d01d1f85177b9c7f5e2745f113f82f38be1c84a362fd28dc2099cd286821abf7781a7dc4a5949a37206644083afb7...

Demo [Register](#) [Login](#) [Forgot your password?](#)

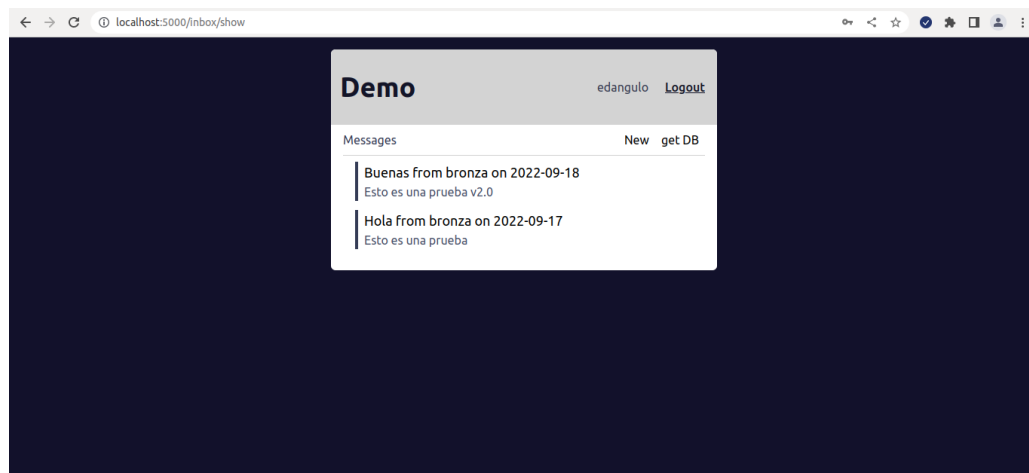
Change your password

Type your new password

Confirm your new password

Submit

- Vista de visualización de mensajes:



localhost:5000/inbox/show

Demo [edangulo](#) [Logout](#)

Messages [New](#) [get DB](#)

Buenas from bronzon on 2022-09-18
Esto es una prueba v2.0

Hola from bronzon on 2022-09-17
Esto es una prueba

- Vista de envío de un nuevo mensaje:

The screenshot shows a web browser window with the address bar displaying 'localhost:5000/inbox/send'. The page has a dark blue background. A white form titled 'New Message' is centered. The form has a header 'Demo' with links 'edangulo' and 'Logout'. Below the header, there are three input fields: 'To', 'Subject', and 'Body'. At the bottom of the form is a blue 'Send' button.

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

Nombre	Ruta	Métodos permitidos	HTTP	Lógica algorítmica
ACTIVATE	@bp.route('/activate')	"GET", "POST"		Valida la entrada de un usuario
REGISTER	@bp.route('/register')	"GET", "POST"		Registro de usuario
CONFIRM	@bp.route('/confirm')	"GET", "POST"		Confirmación de registro de usuario
CHANGE	@bp.route('/change')	"GET", "POST"		
FORGOT	@bp.route('/forgot')	"GET", "POST"		Método que hace una validación cuando el email es invalido y envía un link de recuperación
LOGIN	@bp.route('/login')	"GET", "POST"		Control para realizar el login de un usuario

LOGOUT	@bp.route('/logout')	"GET", "POST"	Función para cerrar sesión y salir

- **ACTIVATE**

- Ruta: @bp.route('/activate')
- Métodos HTTP permitidos: **"GET", "POST"**
- Lógica algorítmica:
- **Valida la entrada de un usuario**

- **REGISTER**

- Ruta: @bp.route('/register')
- Métodos HTTP permitidos: **"GET", "POST"**
- Lógica algorítmica:

El método permite el registro de usuario, aplicado en los campos de username, password y email

- ...

- **CONFIRM**

- Ruta: @bp.route('/confirm')
- Métodos HTTP permitidos: **"GET", "POST"**
- Lógica algorítmica:

Inserte una breve descripción de la lógica algorítmica del método controlador.

Por seguridad reconfirma todos los datos capturados en el método REGISTER, y vuelve a solicitarlos al usuario

-
-
-

- **FORGOT**

- Ruta: @bp.route('/forgot')
- Métodos HTTP permitidos: **"GET", "POST"**
- Lógica algorítmica:

Método que hace una validación cuando el email es invalido y envía un link de recuperación

- LOGIN

- Ruta: `@bp.route('/login')`
- Métodos HTTP permitidos: “GET”, “POST”
- Lógica algorítmica:

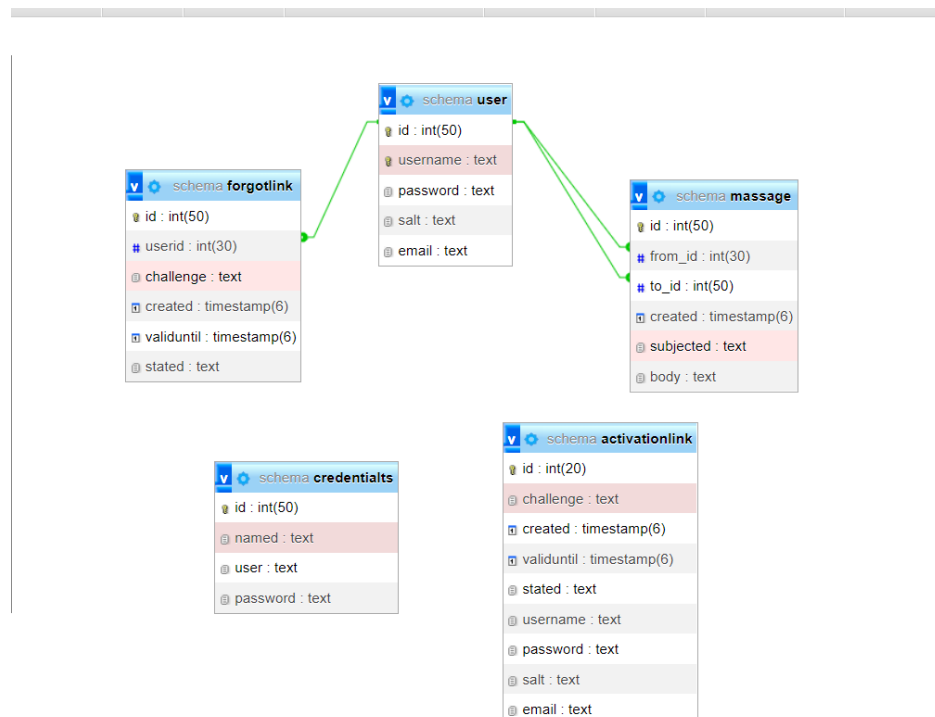
Control para realizar el login de un usuario

- LOGOUT

- Ruta: `@bp.route('/logout')`
- Métodos HTTP permitidos: “GET”, “POST”
- Lógica algorítmica

Función para cerrar sesión y salir

BASE DE DATOS



Prácticas de programación segura (Sprint 3)

Cuando se hablan buenas prácticas de programación, se habla de técnicas, metodologías que se deben implementar en software o sistema de información para que sea escalable, fácil de mantener, seguro para desarrollar, sencillo de desplegar y fácil para los usuarios.

README.md: la documentación a nivel de proyecto ayuda a tener una escritura limpia y ayudara a todo el equipo, ayuda a saber de donde provienen los datos archivos críticos y como se ejecutan las pruebas.

Pipenv: los entornos aislados del sistema operativo facilitan la vida al momento de desarrollar lógica o código.

Git: es una herramienta de versiones de código, ayuda a no sobrescribir y estar actualizando las cosas importantes cada inicio o parte desarrollar con los commits y hacer el push hacia el repositorio diariamente.

*Prepared statements, se trata de cómo hacer una parametrización o consulta de Python para hacer operaciones en la base de datos, ejemplo usar variables de Python para hacer una consulta en la base de datos algo como el ejemplo siguiente del backend del proyecto: `cur.execute('SELECT * FROM tabla = %s', [id])` y se utilizan marcadores de posición para parámetros.*

Hash: es un algoritmo matemático que transforma cualquier dato de entrada en caracteres de salida