

---

# **Universidad Peruana de Ciencias Aplicadas - Informe de Trabajo Final**

Aplicaciones para Dispositivos Móviles - SW65 - CC238

Startup: CodeMinds, Producto: Temporaly, Profesor:  
Mayta Guillermo, Jorge Luis, Integrantes:, Ortega  
Huaraca, Abel Angel: U20201B380, Avila Asto, Alex  
Ramon: U20221A322, Vilchez Rios, Mateo Alejandro:  
U202210059, Ramos Rios, Belén del Rocio: U202216246



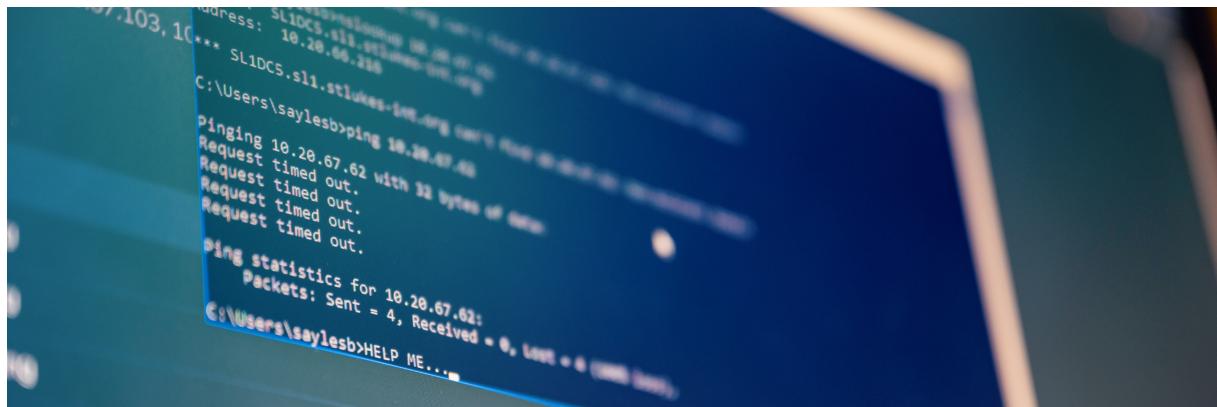
2024-02

# Índice

<b>1 Capítulo V: Product Implementation &amp; Validation</b>	<b>1</b>
1.1 <i>Software Configuration Management</i> . . . . .	1
1.1.1 <i>Software Development Environment Configuration</i> . . . . .	1
1.1.2 <i>Source Code Management</i> . . . . .	2
1.1.3 <i>Source Code Style Guide &amp; Conventions</i> . . . . .	2
1.1.4 <i>Software Deployment Configuration</i> . . . . .	5
1.2 <i>Landing Page &amp; Mobile Application Implementation</i> . . . . .	15
1.2.1 <i>Sprint 2</i> . . . . .	15
1.2.1.1 <i>Sprint Planning 2</i> . . . . .	15
1.2.1.2 <i>Sprint Backlog 2</i> . . . . .	16
1.2.1.3 <i>Development Evidence for Sprint Review</i> . . . . .	23
1.2.1.4 <i>Testing Suite Evidence for Sprint Review</i> . . . . .	24
1.2.1.5 <i>Execution Suite for Sprint Review</i> . . . . .	36
1.2.1.6 <i>Services Documentation Evidence for Sprint Review</i> . . . . .	42
1.2.1.7 <i>Software Deployment Evidence for Sprint Review</i> . . . . .	42
1.2.1.8 <i>Team Collaboration Insights during Sprint</i> . . . . .	45
1.3 <i>Validation Interviews</i> . . . . .	48
1.3.1 Diseño de entrevistas . . . . .	48
1.3.2 Registro de entrevistas . . . . .	50
1.3.3 Evaluaciones segun heurísticas . . . . .	54

# 1 Capítulo V: Product Implementation & Validation

## 1.1 Software Configuration Management



**Figura 1.1:** Imagen extraída de Canva

En esta sección se resume toda la información recopilada y se analizan que pasos se realizarán en el trayecto del proyecto:

### 1.1.1 Software Development Environment Configuration

Esta sección recopila y resume toda la información obtenida, y se analizan los próximos pasos a seguir en el desarrollo del proyecto. Se detallan las acciones clave para asegurar un avance alineado con los objetivos establecidos.

- **UXPressia:** Plataforma colaborativa que nos permitirá crear user personas e integrados con los múltiples mapas para evaluar sus prioridades.
- **Figma:** Herramienta colaborativa que nos permitirá desarrollar *Wireframes*, *Mockups*, *Wireflows*, *UsersFlows* y *Tags*.
- **Mermaid:** Plataforma colaborativa que nos permitirá crear nuestro diagrama de base de datos.

- **PlantUML:** Plataforma que nos permitirá crear diagramas de clases.
- **Webstorm:** *IDE* que utilizaremos para trabajar con *javascript* y desarrollar la *landing page mobile*.
- **IntelliJ IDEA:** *IDE* que utilizaremos para trabajar con *Java* y desarrollar el *backend*.
- **Android Studio/Visual Studio Code:** *IDE* que utilizaremos para trabajar con *Kotlin/Flutter* y desarrollar el *frontend*.

### 1.1.2 Source Code Management

Este proyecto se desarrolló en torno a cinco ramas principales:

- **Main:** Rama principal del proyecto, que contiene las publicaciones oficiales y actualizadas del mismo.
- **Ramas de integrantes del equipo:** Compuesto por un equipo de 4 miembros, cada integrante de CodeMinds trabajó de manera independiente en sus respectivas asignaciones. Además, se colaboró en conjunto para abordar los puntos que lo requerían.

**Temporaly - Auth - Backend** Para acceder al flujo del *backend*, haga click a la [URL](#)

**Temporaly - Email Generate - Backend** Para acceder al flujo del *backend*, haga click a la [URL](#)

**Temporaly - Landing Page** Para acceder al flujo de la *landing page*, haga click a la [URL](#)

**Temporaly - Mobile App** Para acceder al flujo del *mobile app*, haga click a la [URL](#)

### 1.1.3 Source Code Style Guide & Conventions

Para el desarrollo de nuestros proyectos, hemos empleado diversas nomenclaturas, referencias y lenguajes que forman parte de la solución propuesta:

**Landing Page**

- **Tecnologías:** Para el desarrollo de la landing page, utilizamos tecnologías como *HTML5*, *CSS* y *JavaScript*.
- **Herramientas:** Nos apoyamos en herramientas ampliamente utilizadas y recomendadas para el desarrollo web, tales como: *Visual Studio Code*, *WebStorm*, *GitHub*, *Git* y *Figma*.

- **Convenciones de idioma:** Adoptamos el uso del inglés como el idioma principal para la escritura de nuestro código. Sin embargo, la landing page admite *i18n* para usuarios de habla hispana.
- **Lenguaje Gherkin:** Empleamos Gherkin para diseñar las pruebas de cada historia de usuario, siguiendo su estructura básica.

#### **Auth - Backend**

- **Tecnologías:** Para el desarrollo del *backend*, enfocado en el *auth service*, utilizamos tecnologías como *Java*, *Spring Boot* y *Spring Security*.
- **Herramientas:** Nos apoyamos en herramientas ampliamente utilizadas y recomendadas para el desarrollo *backend*, tales como: *Swagger*, *IntelliJ Idea*, *Github*, *Git* y *Docker*.
- **Convenciones de idioma:** Adoptamos el uso del inglés como el idioma principal para la escritura de nuestro código y la documentación de este en general.
- **Lenguaje Gherkin:** Empleamos Gherkin para diseñar las pruebas de cada historia de usuario, siguiendo su estructura básica.

#### **Email Generate - Backend**

- **Tecnologías:** Para el desarrollo del backend, enfocado en el *generate temporary email service*, utilizamos tecnologías como *Python*, *Flask* y *TempMail*.
- **Herramientas:** Nos apoyamos en herramientas ampliamente utilizadas y recomendadas para el desarrollo *backend*, tales como: *Flasgger*, *Termux*, *Github*, *Git* y *Vim*.
- **Convenciones de idioma:** Adoptamos el uso del inglés como el idioma principal para la escritura de nuestro código y la documentación de este en general.
- **Lenguaje Gherkin:** Empleamos Gherkin para diseñar las pruebas de cada historia de usuario, siguiendo su estructura básica.

#### **Mobile App - Android**

- **Tecnologías:** Para el desarrollo del *mobile app* utilizamos tecnologías como *Kotlin* y *Jetpack Compose*.
- **Herramientas:** Nos apoyamos en herramientas ampliamente utilizadas y recomendadas para el desarrollo *mobile android*, tales como: *Android Studio*, *Github* y *Git*
- **Convenciones de idioma:** Adoptamos el uso del inglés como el idioma principal para la escritura de nuestro código y la documentación de este en general.

- **Lenguaje Gherkin:** Empleamos *Gherkin* para diseñar las pruebas de cada historia de usuario, siguiendo su estructura básica.

Convenciones de Commits: Nuestro equipo de desarrollo sigue las pautas establecidas por las Convenciones de Commits, adoptando el formato “Conventional Commits” en su versión 1.0.0 (disponible en [URL](#)). Esto nos permite mantener un historial de cambios claro y comprensible. Nos guiamos por la siguiente estructura:

Donde:

```
<type> [scope opcional] : <description>
```

- **type:** Indica el tipo de modificación realizada, como feat, fix, docs, entre otros.
- **scope:** Define opcionalmente el alcance del cambio en el código.
- **description:** Proporciona un resumen breve y claro de los cambios implementados.

## Convenciones de versionado de lanzamientos

Para la gestión de versiones, seguimos el estándar “Semantic Versioning 2.0.0”. En este formato, las versiones se presentan como (X.Y.Z), con las siguientes interpretaciones:

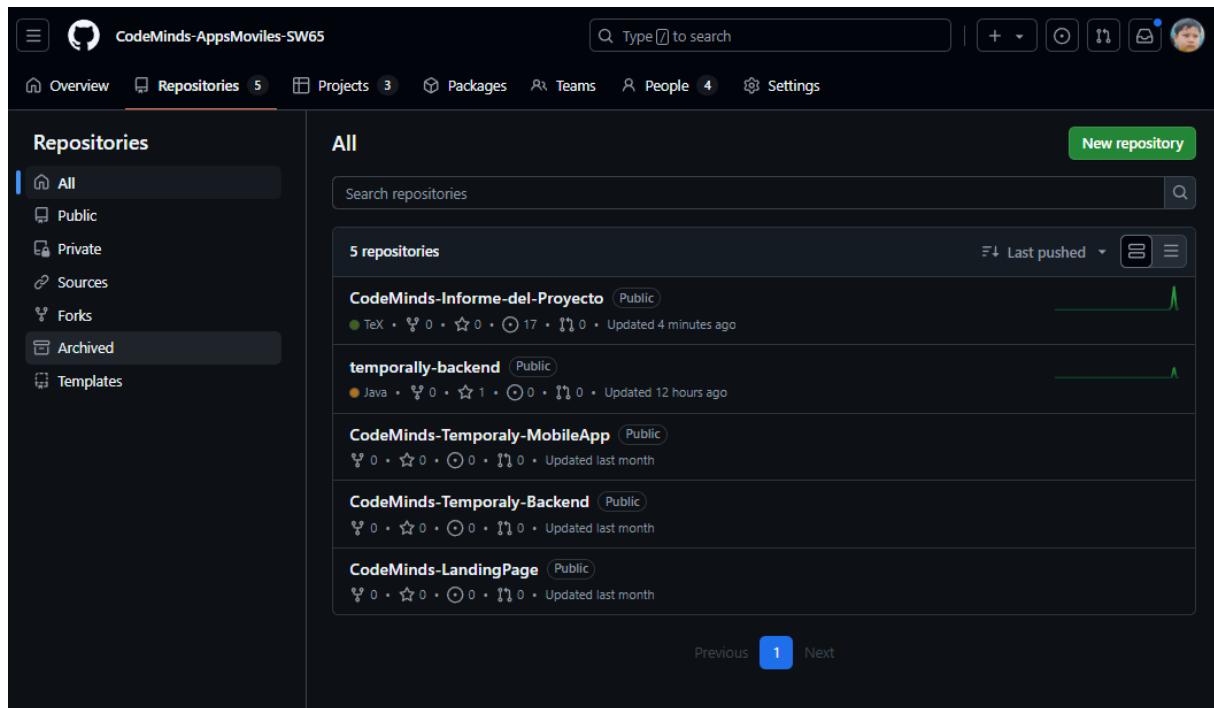
- **X:** Indica una versión principal que introduce cambios incompatibles con versiones anteriores. Comenzamos en 0 durante la fase de desarrollo inicial y pasamos a 1 cuando la versión está lista para su lanzamiento público. Por convención, los números Y y Z se reinician a 0 cada vez que X aumenta.
- **Y:** Representa una versión secundaria que incluye cambios compatibles con versiones anteriores. También abarca los commits de las “release branches” cuando se agregan nuevas funcionalidades. Al igual que con X, Z se reinicia a 0 cuando Y aumenta.
- **Z:** Refleja parches o correcciones de errores menores. Se integra con commits provenientes de la “rama de corrección” y se fusiona con la rama principal.

### 1.1.4 Software Deployment Configuration

Para acceder al flujo de trabajo, haga click a la [URL](#)

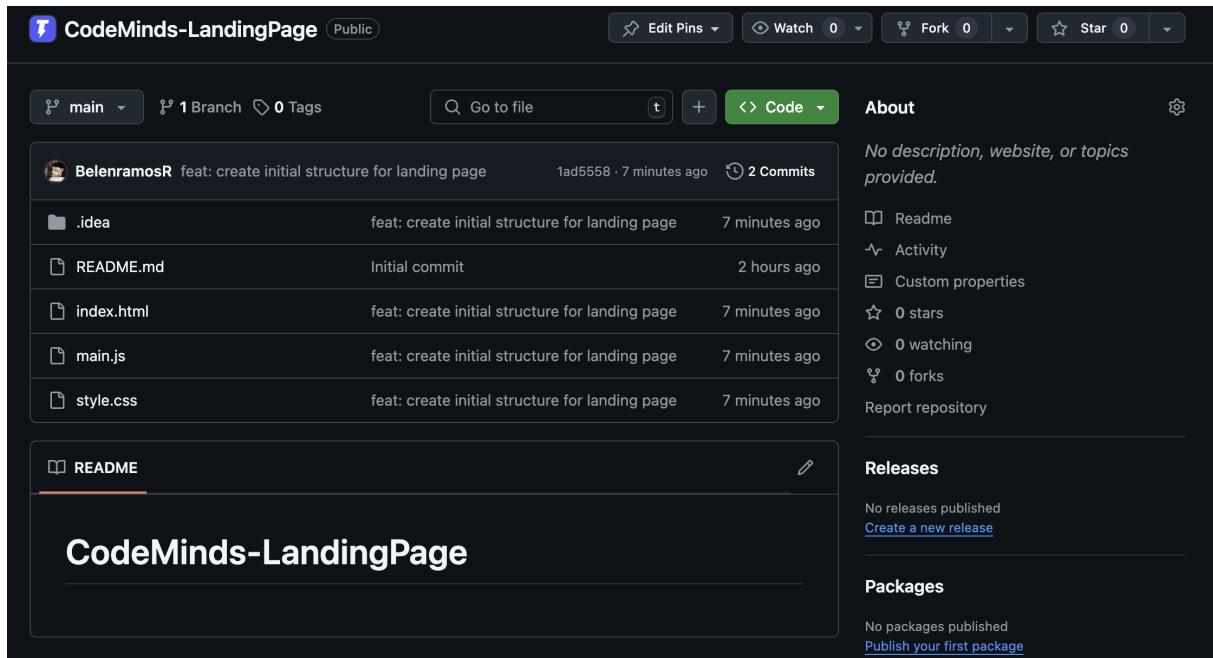
#### Desplegar Landing Page

1. Seleccionar el repositorio *CodeMinds-LandingPage*



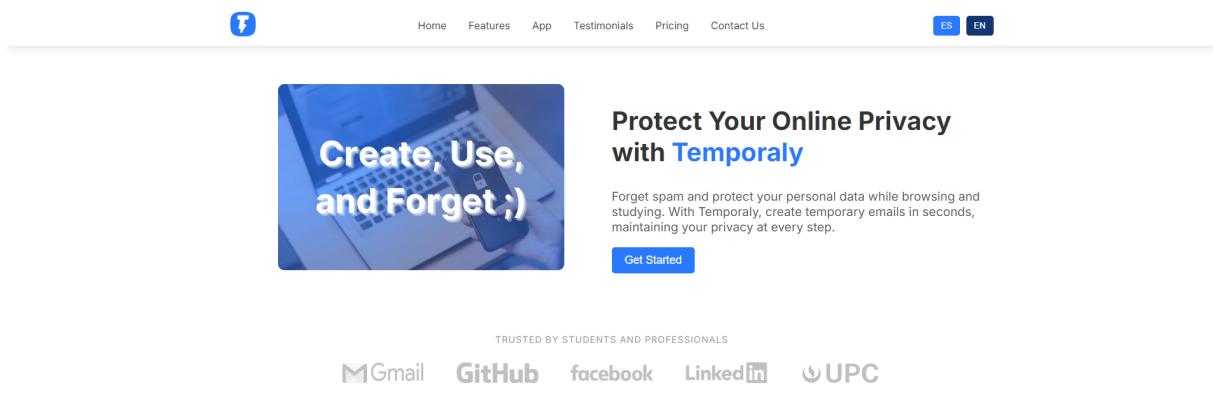
**Figura 1.2:** Organización CodeMinds, imagen extraída de Github

## 2. Abrir, descargar y ejecutar en VSCode o WebsTorm el archivo index.html

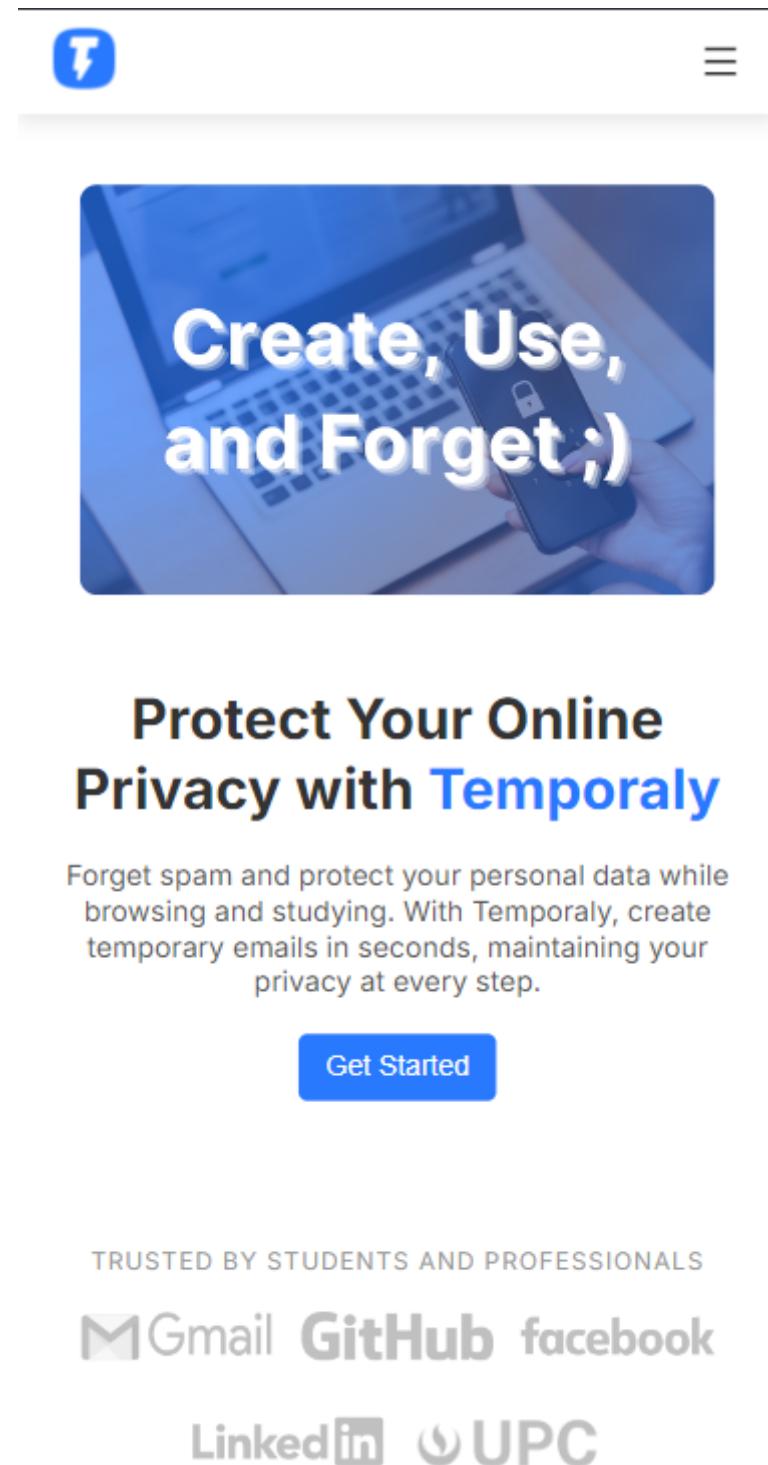


**Figura 1.3:** Repositorio de la Landing Page, imagen extraída de Github

## 3. Visualización de la *Landing Page*



**Figura 1.4:** Web Landing Page - CodeMinds, imagen extraída de Github



**Figura 1.5:** Mobile Landing Page - CodeMinds, imagen extraída de Github

## Desplegar Auth - Backend

1. Seleccionar el repositorio *CodeMinds-Temporaly-Backend*

The screenshot shows the GitHub interface for the organization 'CodeMinds-AppsMovies-SW65'. The left sidebar shows categories: All, Public, Private, Sources, Forks, Archived, and Templates. The main area displays a list of repositories under the heading 'All'. The 'temporaly-backend' repository is visible, along with others such as 'CodeMinds-Informe-del-Proyecto', 'CodeMinds-Temporaly-MobileApp', and 'CodeMinds-LandingPage'. Each repository card includes its name, public status, language, commit count, star count, and last update time.

**Figura 1.6:** Organización CodeMinds, imagen extraída de Github

2. Abrir, descargar y ejecutar en *IntelliJ Idea*

The screenshot shows the GitHub repository page for 'CodeMinds-Temporaly-Backend'. At the top, it shows the repository name, a public status badge, and metrics: 2 branches, 0 tags, 1 watch, 0 forks, and 1 star. Below this is a search bar and a 'Code' dropdown menu. The main content area displays a list of 44 commits from 'Ryeon'. Each commit includes a message, a file list, and a timestamp. To the right of the commits, there's an 'About' section with a note 'No description, website, or topics provided.', and sections for 'Releases' (no releases) and 'Packages' (no packages). The repository has 1 star and 1 watcher.

**Figura 1.7:** Repositorio del backend, imagen extraída de Github

### 3. Visualización de la documentación Swagger

The screenshot shows the Swagger UI interface for the Temporally API. At the top, there's a header with the Swagger logo, the URL '/v1/api-docs', and a green 'Explore' button. Below the header, the title 'Temporally API' is displayed along with version 'V1.0.0' and 'OAS 3.0'. A sub-header indicates it's for the 'Temporally project' using 'Apache 2.0'. On the left, there's a 'Servers' dropdown set to 'https://temporally-api.ryzeon.me'. On the right, there's an 'Authorize' button with a lock icon. The main content area is divided into sections: 'Authentication' (containing four POST endpoints) and 'OAuth' (containing three GET endpoints). Each endpoint is shown with its method, path, and a lock icon.

Method	Path	Lock Icon
POST	/api/v1/authentication/sign-up	Locked
POST	/api/v1/authentication/sign-in	Locked
POST	/api/v1/authentication/refresh-token/{refreshToken}	Locked
POST	/api/v1/authentication/change-password	Locked
GET	/user	Locked
GET	/success	Locked
GET	/logout	Locked

**Figura 1.8:** Swagger, imagen extraída de Github

### Desplegar Email Generate - Backend

1. Seleccionar el repositorio *CodeMinds-Temporaly-Backend-GenerateMails-Python*

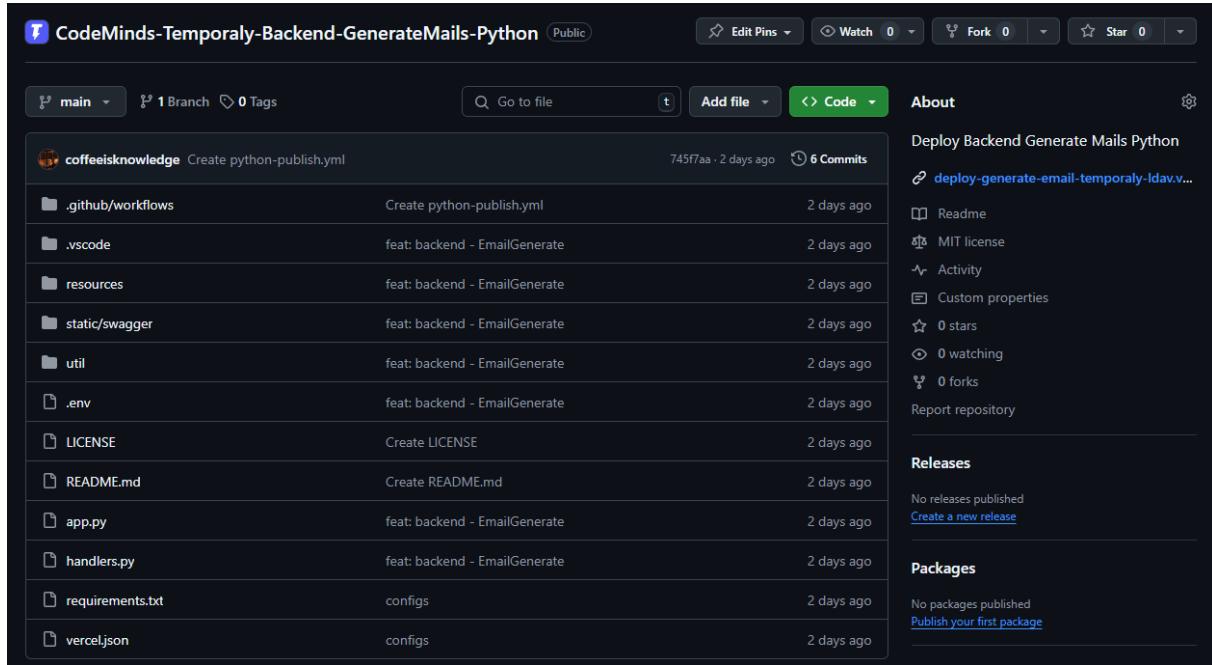
The screenshot shows the GitHub organization page for 'CodeMinds-AppsMovies-SW65'. The left sidebar has a 'Repositories' section with options: All (selected), Public, Private, Sources, Forks, Archived (selected), and Templates. The main area displays 'All' repositories with a search bar and a 'New repository' button. There are five repositories listed:

- CodeMinds-Informe-del-Proyecto** (Public)  
• Tex • 0 • 0 stars • 17 commits • 0 issues • Updated 4 minutes ago
- temporally-backend** (Public)  
• Java • 0 • 0 stars • 0 commits • 0 issues • Updated 12 hours ago
- CodeMinds-Temporaly-MobileApp** (Public)  
• 0 • 0 stars • 0 commits • 0 issues • Updated last month
- CodeMinds-Temporaly-Backend** (Public)  
• 0 • 0 stars • 0 commits • 0 issues • Updated last month
- CodeMinds-LandingPage** (Public)  
• 0 • 0 stars • 0 commits • 0 issues • Updated last month

At the bottom, there are navigation buttons: Previous, 1 (highlighted), and Next.

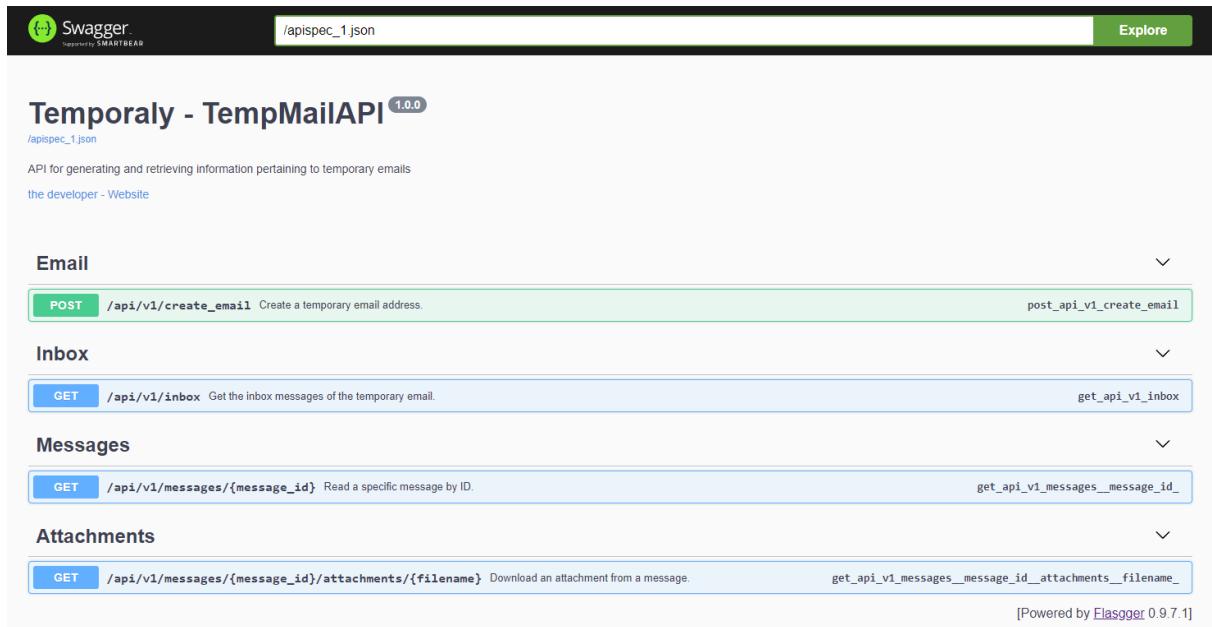
**Figura 1.9:** Organización CodeMinds, imagen extraída de Github

## 2. Abrir, descargar y ejecutar en VSCode, Vim o su editor de preferencia



**Figura 1.10:** Repositorio del backend, imagen extraída de Github

## 3. Visualización de la documentación Flasgger



**Figura 1.11:** Flasgger, imagen extraída de Github

### Desplegar Mobile App - Temporaly

1. Seleccionar el repositorio *CodeMinds-Temporaly-MobileApp*

The screenshot shows the GitHub organization interface for 'CodeMinds-AppsMovies-SW65'. The left sidebar has a 'Repositories' section with options: All (selected), Public, Private, Sources, Forks, Archived (highlighted), and Templates. The main area displays 'All' repositories with a search bar and a 'New repository' button. There are five repositories listed:

- CodeMinds-Informe-del-Proyecto** (Public)  
• Tex • 0 stars • 17 commits • Updated 4 minutes ago
- temporally-backend** (Public)  
• Java • 0 stars • 0 commits • Updated 12 hours ago
- CodeMinds-Temporaly-MobileApp** (Public)  
• 0 stars • 0 commits • 0 issues • Updated last month
- CodeMinds-Temporaly-Backend** (Public)  
• 0 stars • 0 commits • 0 issues • Updated last month
- CodeMinds-LandingPage** (Public)  
• 0 stars • 0 commits • 0 issues • Updated last month

Pagination at the bottom shows 'Previous' and 'Next' buttons.

**Figura 1.12:** Organización CodeMinds, imagen extraída de Github

## 2. Abrir, descargar y ejecutar en Android Studio



Asegurese de tener la versión Android Studio Koala - 2024 - Patch 1 o 2

The screenshot shows a GitHub repository page for 'CodeMinds-Temporaly-MobileApp'. The main area displays a list of commits:

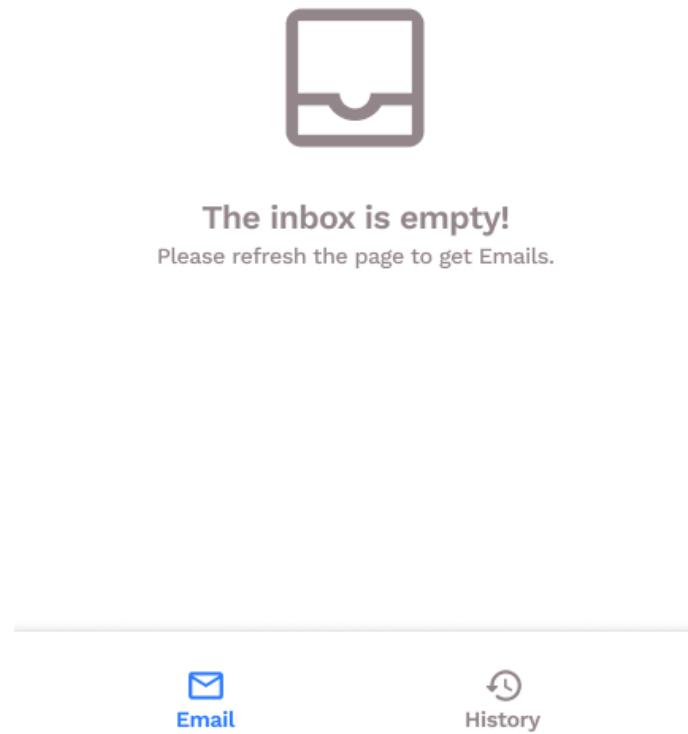
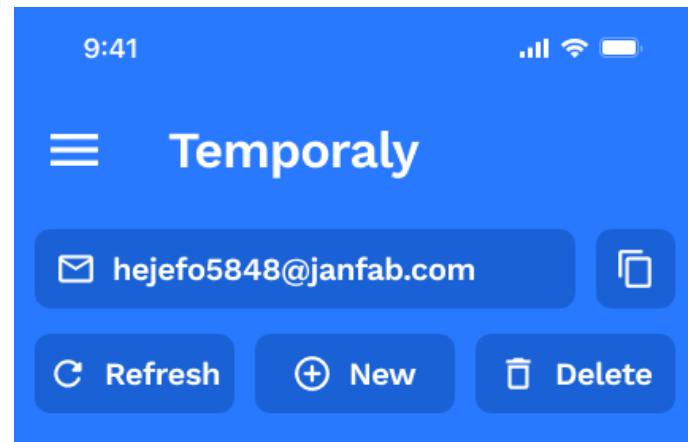
Author	Commit Message	Date
coffeeisknowledge	fix: Fix with Backup	e91782e · 19 hours ago
app	fix: Fix with Backup	19 hours ago
gradle	fix: Fix with Backup	19 hours ago
.gitignore	feat: initial commit	5 days ago
build.gradle.kts	feat: initial commit	5 days ago
gradle.properties	feat: initial commit	5 days ago
gradlew	feat: initial commit	5 days ago
gradlew.bat	feat: initial commit	5 days ago

The right sidebar contains the following information:

- About:** No description, website, or topics provided.
- Activity:** Empty.
- Custom properties:** Empty.
- Report repository:** Empty.
- Releases:** No releases published. [Create a new release](#)

**Figura 1.13:** Repositorio del Mobile App, imagen extraída de Github

3. Visualización de la aplicación *Temporaly*



**Figura 1.14:** Flasgger, imagen extraída de Github

## 1.2 Landing Page & Mobile Application Implementation

### 1.2.1 Sprint 2

#### 1.2.1.1 Sprint Planning 2

Sprint Planning Background	
Date	27/09/2024
Time	21:00
Location	Discord Virtual Meeting
Prepared By	Ortega Huaraca, Abel Angel
Attendees (to planning meeting)	Ortega Huaraca, Abel Angel / Avila Asto, Alex Ramon / Vilchez Rios, Mateo Alejandro / Ramos Rios, Belen del Rocio
Sprint 2 Review Summary	Implementación de los User Stories e integración con el Backend (Cap 4 y 5)
Sprint 2 Retrospective Summary	El trabajo se realizó a tiempo, y cada integrante terminó lo que se le encomendó. Asimismo, se trabajó en equipo cuando la situación del proyecto lo requería.
Sprint Goal & User Stories	Establecer una minidemo de la aplicación Temporaly, en donde se muestre la entrega y cumplimiento de los user stories para este sprint 2
Sprint 2 Velocity	1 Sprint cada semanas
Sum of Story Points	- 106
Sprint	<b>Sprint 2</b>

### **1.2.1.2 Sprint Backlog 2**

En este segundo sprint, el foco ha sido completar e integrar funcionalidades críticas dentro de la aplicación móvil de Temporaly, así como afianzar la conexión con el backend. Las historias de usuario incluidas en este sprint cubren tareas relacionadas con la creación y gestión de correos temporales, personalización de dominios, y optimización del proceso de generación y visualización de correos, asegurando que la experiencia de usuario sea intuitiva y eficiente.

Ademas, se ha priorizado garantizar una interacción fluida entre las interfaces móviles desarrolladas en Kotlin y las APIs del backend, permitiendo a los usuarios gestionar correos temporales de manera más personalizada. Además, se incluyeron optimizaciones en la API y el manejo de estado en la aplicación móvil.

A continuación, se detallan las Historias de Usuario, los Work Items (WI) asignados, las horas estimadas, los responsables y el estado de cada tarea.

ID	User Story	Work Item (WI)	Description	Estimation (Hours)	Assigned To	Status
US38	Descubrimiento intuitivo	WI01	Diseño de la estructura HTML y navegación principal para facilitar el descubrimiento intuitivo en la landing page	6	Belén Ramos	Done
		WI02	Implementación de los elementos de navegación y estructura responsive	4	Belén Ramos	Done
		WI03	Pruebas de usabilidad para navegación y descubrimiento	2	Mateo Vílchez	Done
US39	Contenido informativo	WI04	Redacción de todo el contenido informativo para la landing page	3	Belén Ramos	Done
		WI05	Implementación del contenido en la landing page	2	Belén Ramos	Done
US40	Compatibilidad móvil	WI06	Desarrollo del diseño responsive para dispositivos móviles	4	Belén Ramos	Done

		WI07	Pruebas de responsividad en diferentes resoluciones	3	Belén Ramos	Done
US41	Formulario de contacto	WI08	Diseño e implementación del formulario de contacto	5	Belén Ramos	Done
		WI09	Validación de formulario de contacto	2	Mateo Vílchez	Done
US42	Contenido multimedia	WI10	Integración de imágenes, íconos y secciones visuales multimedia	3	Belén Ramos	Done
		WI11	Optimización de recursos multimedia para tiempos de carga más rápidos	2	Mateo Vílchez	Done
US43	Call-to-action claro	WI12	Diseño y optimización de los botones de call-to-action (CTA) en la landing page	3	Mateo Vílchez	Done
US32	RESTful API Registro de usuario	WI13	Implementación de la API para el registro de usuario	6	Alex Avila	Done
		WI14	Añadir index para role seed en persistencia	3	Alex Avila	Done
		WI15	Middleware para manejo de excepciones	4	Alex Avila	Done

US33	RESTful API Inicio de sesión de usuario	WI16	Implementación de la API para inicio de sesión	5	Alex Avila	Done
		WI17	Implementación de pipeline de seguridad	5	Alex Avila	Done
US34	Autenticación basada en token JWT	WI18	Implementación de autenticación JWT en backend	6	Alex Avila	Done
		WI19	Configuración inicial de OpenAPI	5	Alex Avila	Done
US35	Recuperación de contraseña	WI20	Implementación de perfiles en Spring Boot	4	Alex Avila	Done
US36	RESTful API Creación de sesiones	WI21	Implementación de la API para la creación de sesiones	4	Alex Avila	Done
		WI22	Creación de Dockerfile para build y compose	5	Alex Avila	Done
		WI23	Creación de Dockerfile para entorno de desarrollo	4	Alex Avila	Done
US37	Paginación y filtrado de resultados	WI24	Implementación de la paginación y filtrado	5	Alex Avila	In Process
US05	Generación de correo temporal con un clic	WI25	Implementación del recurso POST para generar un correo temporal	5	Abel Ortega	Done
		WI26	Implementación del recurso GET para obtener la lista de correos temporales generados	4	Abel Ortega	Done
US06	Duración específica del correo temporal	WI27	Implementación del recurso PUT para actualizar la duración específica del correo temporal	4	Abel Ortega	Done
US07	Confirmación visual de creación de correo	WI28	Implementación del recurso GET para confirmar la creación visual del correo temporal	3	Abel Ortega	Done

US08	Personalización del dominio del correo temporal	WI29	Implementación del recurso PUT para personalizar el dominio del correo temporal	5	Abel Ortega	Done
US01	Registro de usuario	WI30	Diseño de la pantalla de registro utilizando Jetpack Compose	6	Alex Avila	Done
		WI02	Conectar la pantalla de registro con la API	5	Alex Avila	Done
		WI03	Implementar validación de campos en la pantalla de registro	4	Alex Avila	Done
		WI04	Pruebas de usabilidad para el registro de usuario	3	Alex Avila	Done
US02	Confirmación de creación de cuenta	WI05	Implementar la funcionalidad de confirmación de cuenta con la API	5	Alex Avila	Done
		WI31	Diseño de la pantalla de confirmación de cuenta en Compose	4	Alex Avila	Done
		WI32	Pruebas para la confirmación de cuenta	2	Alex Avila	Done
US03	Verificación de cuenta	WI33	Conectar la verificación de cuenta con la API	4	Alex Avila	Done
		WI34	Implementar la pantalla de verificación de cuenta	3	Alex Avila	Done
US04	Inicio de sesión de cuenta	WI35	Crear la pantalla de inicio de sesión en Compose	5	Alex Avila	Done
		WI36	Conectar la pantalla de inicio de sesión con la API	5	Alex Avila	Done
US05	Generación de correo temporal con un clic	WI37	Diseño de la pantalla de generación de correos temporales	4	Abel Ortega	Done
		WI38	Implementar la funcionalidad de generación de correos con un clic	5	Abel Ortega	Done

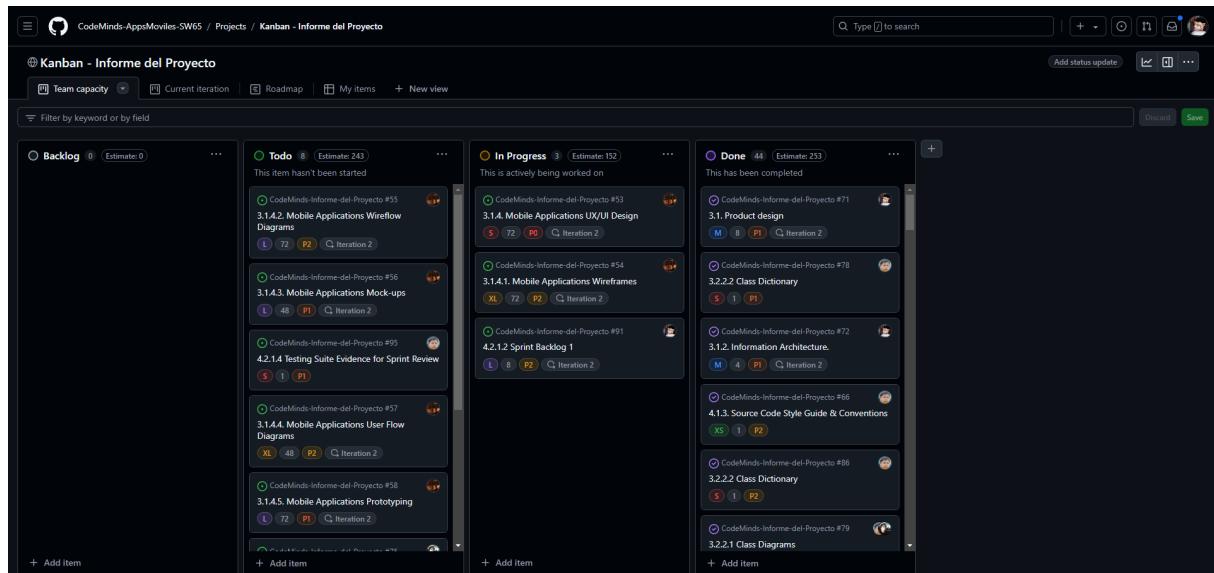
		WI39	Pruebas de generación de correos temporales	2	Abel Ortega	Done
US06	Duración específica del correo temporal	WI40	Implementar opción para seleccionar duración del correo	4	Abel Ortega	Done
		WI41	Conectar la opción de duración con la API	4	Abel Ortega	Done
US07	Confirmación visual de creación de correo	WI42	Añadir confirmación visual en la pantalla de generación de correos	3	Abel Ortega	Done
		WI43	Pruebas de confirmación visual	2	Abel Ortega	Done
US09	Generación múltiple de correos temporales	WI44	Implementar la opción para generar múltiples correos	4	Abel Ortega	Done
		WI45	Pruebas para la generación de múltiples correos	3	Abel Ortega	Done
US10	Copiar correo temporal al portapapeles	WI46	Implementar la funcionalidad para copiar correos al portapapeles	4	Abel Ortega	Done
		WI47	Pruebas para la funcionalidad de copiar al portapapeles	2	Abel Ortega	Done
US11	Visualización de correos temporales activos	WI48	Diseño de la interfaz de visualización de correos activos	5	Abel Ortega	Done
		WI49	Implementación de la funcionalidad de visualización de correos activos	6	Abel Ortega	Done
		WI50	Pruebas para la visualización de correos activos	2	Abel Ortega	Done
US13	Proceso de generación rápido y fluido	WI51	Implementar un proceso rápido de generación de correos temporales	4	Abel Ortega	Done
		WI52	Pruebas del proceso rápido de generación	2	Abel Ortega	Done

US15	Acceso a bandeja de entrada de correos temporales	WI53	Implementar la funcionalidad de acceso a la bandeja de entrada	5	Abel Ortega	Done
		WI54	Pruebas para el acceso a la bandeja de entrada	2	Abel Ortega	Done
US16	Visualización de correos recibidos	WI55	Implementar la funcionalidad de visualización de correos recibidos	5	Abel Ortega	Done
		WI56	Pruebas para la visualización de correos recibidos	2	Abel Ortega	Done
US32	Manejo de consentimiento de usuario	WI57	Implementación del manejo de estado del consentimiento de usuario en la app móvil	3h	Belen Ramos	Done
		WI58	Implementación de la lógica para mostrar/ocultar mensajes según aceptación del consentimiento	3h	Belen Ramos	Done
		WI59	Integración de la funcionalidad de "I Agree" con el flujo de consentimiento y su persistencia	3h	Belen Ramos	Done
		WI60	Implementación de la navegación entre la pantalla de consentimiento y la pantalla principal	2h	Mateo Vilchez	In Process
		WI70	Implementación del click handler para navegar a la sección completa de "Privacy Policy"	2h	Belen Ramos	Done
		WI71	Pruebas unitarias para el componente de consentimiento y verificación de aceptación	3h	Mateo Vilchez	To Do

	WI72	Manejo de estados para permitir o deshabilitar interacciones con botones según el consentimiento	2h	Mateo Vilchez	Done
	WI73	Pruebas de usabilidad para asegurar el flujo correcto en la sección de "Privacy Policy"	2h	Mateo Vilchez	In Process
	WI74	Validación de errores y feedback visual al usuario en caso de fallos en el consentimiento	3h	Belen Ramos	Done

### Gestión del Sprint 2 (Tablero Kanban):

Para mejorar la gestión y seguimiento de las tareas de este *sprint*, se implementó un tablero Kanban. Este tablero permite visualizar claramente los elementos clave a desarrollar, asignar responsables para cada tarea, y utilizar etiquetas (*labels*) que categorizan los Issues de forma precisa. Además, los *Milestones* fueron utilizados estratégicamente para planificar las fechas de entrega y gestionar los entregables, facilitando un control eficiente del progreso del proyecto.



**Figura 1.15:** Tablero Kanban del equipo, imagen extraída de Github

### 1.2.1.3 Development Evidence for Sprint Review

Evidencia de colaboración en Equipo:

The screenshot shows a GitHub Kanban board titled "Kanban - Informe del Proyecto". The board has columns for "Title", "Priority", "Size", "Estimate", and "Iteration". There are 18 items listed, each with a small icon and a link. The items are categorized into three main sections: 3.1.4 (items 1-6), 4.2.1 (items 7-11), and 3.1.2 (items 12-18). The "Iteration" column indicates that most items belong to "Iteration 2".

	Title	Priority	Size	Estimate	Iteration
1	3.1.4. Mobile Applications UX/UI Design #53	P0	S	72	Iteration 2
2	3.1.4.1. Mobile Applications Wireframes #54	P2	XL	72	Iteration 2
3	3.1.4.2. Mobile Applications Wireflow Diagrams #55	P2	L	72	Iteration 2
4	3.1.4.3. Mobile Applications Mock-ups #56	P1	L	48	Iteration 2
5	3.1.4.4. Mobile Applications User Flow Diagrams #57	P2	XL	48	Iteration 2
6	3.1.4.5. Mobile Applications Prototyping #58	P1	L	72	Iteration 2
7	4.2.1.7 Services Deployment Evidence for Sprint Review #73	P1	L		
8	4.2.1.6 Services Documentation Evidence for Sprint Review #74	P1	M	2	
9	4.2.1.5 Execution Evidence for Spring Review #75	P1	S		
10	Student Outcomes #43				
11	4.2.1.2 Sprint Backlog 1 #91	P2	L	8	Iteration 2
12	3.1. Product design #71	P1	M	8	Iteration 2
13	3.2.2.2 Class Dictionary #78	P1	S	1	
14	4.2.1.4 Testing Suite Evidence for Sprint Review #95	P1	M	1	
15	3.1.2. Information Architecture. #72	P1	M	4	Iteration 2
16	4.1.3. Source Code Style Guide & Conventions #66	P2	XS	1	
17	3.2.2.2 Class Dictionary #86	P2	S	1	
18	3.2.2.1 Class Diagrams #79	P1	M	2	

**Figura 1.16:** Imagen extraída de Github

#### **1.2.1.4 Testing Suite Evidence for Sprint Review**

##### **Landing US - Gherking**

US38 - Descubrimiento intuitivo

Feature: Descubrimiento intuitivo de la landing page

As a user

I want to navigate the landing page easily  
So that I can discover the content intuitively

Scenario: User navigates the landing page smoothly

Given the user is on the landing page  
When the user scrolls and clicks on navigation links  
Then the user should be able to access all sections of the page smoothly

Examples:

section	
Home	
Features	
Contact Information	

US39 - Contenido informativo

Feature: Contenido informativo en la landing page

As a user

I want to find clear and concise information  
So that I understand the purpose of the application

Scenario: User reads about features and benefits

Given the user is on the "Features" section of the landing page  
When the user reads through the content provided  
Then the user should understand the main features and benefits of the application

Examples:

content_section	
Application Uses	
User Testimonials	
Pricing Options	

US40 - Compatibilidad móvil

Feature: Compatibilidad móvil para la landing page

As a mobile user

I want to view the landing page responsivamente  
So that I can easily navigate on any device

```
Scenario: User accesses the landing page on a mobile device
Given the user opens the landing page on a smartphone
When the page loads
Then the content should adjust to fit the screen size without losing functionality
```

Examples:

device	
Smartphone	
Tablet	
Small Laptop	

#### US41 - Formulario de contacto

Feature: Formulario de contacto en la landing page

```
As a user
I want to fill out a contact form
So that I can get in touch with the team
```

```
Scenario: User submits the contact form
Given the user is on the "Contact" section of the landing page
When the user fills out all required fields and clicks "Submit"
Then the form should be successfully sent, and a confirmation message should appear
```

Examples:

name	email	message	
John Doe	john@example.com	Hello!	
Jane Smith	jane_smith@example.com	Need more info	

#### US42 - Contenido multimedia

Feature: Contenido multimedia en la landing page

```
As a user
I want to see multimedia content like images and videos
So that I can understand the information visually
```

```
Scenario: User views images and videos on the landing page
Given the user is navigating through different sections
When multimedia content (images, videos) is present
Then the user should be able to view the content clearly and it should load quickly
```

Examples:

content_type	
Images	
Videos	
GIFs	

US43 - Call-to-action claro

Feature: Call-to-action (CTA) en la landing page

As a user

I want to find clear CTA buttons

So that I can take actions like sign-up or contact easily

Scenario: User clicks on CTA buttons

Given the user is on the landing page

When the user sees a CTA button like "Sign Up Now"

Then the button should be prominently displayed and redirect to the correct action page

Examples:

button_text	action
Sign Up Now	Registration
Contact Us	Contact Form
Learn More	Features Section

### **Frontend US - Gherking**

US01 - Registro de usuario

Feature: Registro de un nuevo usuario

As a user

I want to register an account in the app

So that I can access all its features

Scenario: Registro exitoso

Given the user is on the registration page

When they fill out all required fields with valid information

And click "Register"

Then the system should create a new user account

And display a success message

Examples:

username	email	password
user_one	user1@example.com	pass123
user_two	user2@example.com	pass456

US02 - Confirmación de creación de cuenta

Feature: Confirmación del registro de cuenta

As a user

I want to receive a confirmation email

So that I can verify my account

Scenario: Recepción de correo de confirmación

Given a user registers successfully

```
When the system creates the account
Then a confirmation email should be sent to the user's email address
```

Examples:

email	
user1@example.com	
user2@example.com	

#### US03 - Verificación de cuenta

Feature: Verificación de cuenta por correo

As a user  
I want to verify my account through email  
So that I can activate my access

Scenario: Verificar cuenta exitosamente

Given a user receives a confirmation email  
When they click on the verification link in the email  
Then the system should verify their account and display a success message

Examples:

email	
user1@example.com	
user2@example.com	

#### US04 - Inicio de sesión de cuenta

Feature: Iniciar sesión en la app

As a user  
I want to log in to my account  
So that I can access personalized features

Scenario: Inicio de sesión exitoso

Given the user is on the login page  
When they provide valid credentials  
Then the system should log the user in and redirect to the dashboard

Examples:

email	password
user1@example.com	pass123
user2@example.com	pass456

#### US05 - Generación de correo temporal con un clic

Feature: Generar correo temporal

As a user  
I want to create a temporary email with one click  
So that I can use it quickly without hassle

```
Scenario: Crear correo temporal
Given the user is on the email generation page
When they click the "Generate Email" button
Then a new temporary email should be generated immediately
```

Examples:

action
Generate Email

#### US06 - Duración específica del correo temporal

```
Feature: Especificar duración del correo temporal
As a user
I want to set how long my temporary email will last
So that I can control its expiration
```

Scenario: Seleccionar duración del correo

```
Given the user is creating a temporary email
When they select a specific duration
Then the system should set the expiration based on the chosen duration
```

Examples:

duration
10 minutes
30 minutes

#### US07 - Confirmación visual de creación de correo

```
Feature: Confirmación visual de correo temporal
As a user
I want to receive visual feedback when my temporary email is created
So that I can be sure of its creation
```

Scenario: Mostrar confirmación

```
Given a user generates a temporary email
When the email is created successfully
Then the system should display a confirmation message on the screen
```

Examples:

status
created
failed

#### US08 - Personalización del dominio del correo temporal

```
Feature: Cambiar dominio del correo temporal
As a user
I want to choose a domain for my temporary email
So that I can customize its address
```

**Scenario:** Seleccionar dominio personalizado  
Given the user is on the email creation page  
When they choose a domain from the available options  
Then the temporary email should be created with the selected domain

Examples:

domain
tempmail.com
example.net

#### US09 - Generación múltiple de correos temporales

**Feature:** Generar varios correos temporales  
As a user  
I want to create multiple temporary emails  
So that I can use different addresses for various purposes

**Scenario:** Crear múltiples correos

Given the user is on the email generation page  
When they click the "Generate Multiple Emails" button  
Then multiple temporary emails should be created

Examples:

action
Generate Multiple

#### US10 - Copiar correo temporal al portapapeles

**Feature:** Copiar correo al portapapeles  
As a user  
I want to copy my temporary email address to the clipboard  
So that I can paste it quickly where needed

**Scenario:** Copiar correo al portapapeles

Given a temporary email is displayed  
When the user clicks "Copy to Clipboard"  
Then the email address should be copied to the clipboard

Examples:

button_text
Copy to Clipboard

#### US11 - Visualización de correos temporales activos

**Feature:** Ver correos temporales activos  
As a user  
I want to see a list of my active temporary emails  
So that I can manage them effectively

**Scenario:** Mostrar lista de correos activos  
Given the user has active temporary emails  
When they navigate to the "My Emails" section  
Then the system should display a list of all active temporary emails

Examples:

section_name
My Emails

#### US12 - Sugerencias automáticas de nombres

**Feature:** Sugerencias de nombres para correos temporales  
As a user  
I want the system to suggest names for my temporary emails  
So that I can choose a name quickly

**Scenario:** Mostrar sugerencias de nombres

Given a user is creating a temporary email  
When the system generates name suggestions  
Then the user should be able to select one of the suggested names

Examples:

suggestion
random_name1
random_name2

#### US13 - Proceso de generación rápido y fluido

**Feature:** Rápida generación de correos  
As a user  
I want to create temporary emails quickly and smoothly  
So that I can avoid delays

**Scenario:** Generar correo rápidamente

Given a user wants to create a temporary email  
When they start the creation process  
Then the system should complete the email generation quickly

Examples:

action
Generate Now

#### US14 - Advertencia de expiración de correo temporal

**Feature:** Advertencia de expiración  
As a user  
I want to receive a warning before my temporary email expires  
So that I can take action if needed

**Scenario:** Mostrar advertencia de expiración  
Given a temporary email is about to expire  
When the expiration time is near  
Then the system should display a warning message to the user

Examples:

time_remaining
5 minutes
1 minute

#### US15 - Acceso a bandeja de entrada de correos temporales

**Feature:** Ver bandeja de entrada de correos temporales  
As a user  
I want to access the inbox of my temporary email  
So that I can read incoming messages

**Scenario:** Mostrar bandeja de entrada  
Given the user has an active temporary email  
When they click on the inbox icon  
Then the system should open the inbox view for that email

Examples:

action
Open Inbox

#### US16 - Visualización de correos recibidos

**Feature:** Ver correos recibidos  
As a user  
I want to see all emails received in my temporary inbox  
So that I can manage my messages

**Scenario:** Mostrar correos recibidos  
Given the user is in the inbox view  
When new emails are received

#### US32 - Manejo de consentimiento de usuario

**Feature:** Consent management for users  
As a user  
I want to give my consent to the platform  
So that I can use the service while agreeing to its policies

**Scenario:** Accept user consent  
Given the user is presented with a consent form  
When they click on "I Agree"  
Then the user's consent should be recorded in the system

```
And the user should be redirected to the main app page
```

Examples:

consent_action
I Agree
Accept Terms

### **Backend US - Gherking**

```
US32 - RESTful API Registro de usuario
```

Feature: API de registro de usuario

As a user

I want to register to the system via the API

So that I can have an account

Scenario: Registro exitoso de un nuevo usuario

Given a user provides valid registration details

When the user submits their information to the "/register" endpoint

Then a new user account should be created

And the system should respond with status 201

Examples:

email	password
user1@example.com	pass123
user2@example.com	pass456

```
US33 - RESTful API Inicio de sesión de usuario
```

Feature: API de inicio de sesión

As a user

I want to log into the system via the API

So that I can access my account

Scenario: Inicio de sesión exitoso

Given a registered user provides valid credentials

When the user submits their credentials to the "/login" endpoint

Then the system should authenticate the user

And respond with a JWT token and status 200

Examples:

email	password
user1@example.com	pass123
user2@example.com	pass456

```
US34 - Autenticación basada en token JWT
```

Feature: Token JWT para autenticación

As a developer

```
I want to use JWT tokens
So that the authentication is secure

Scenario: Autenticación de usuario con JWT
Given a user has logged in successfully
When they access a protected resource
Then the system should verify the JWT token
And allow access to the resource if valid

Examples:
| token_status |
| valid         |
| expired       |
```

## US35 – Recuperación de contraseña

```
Feature: Recuperación de contraseña de usuario
As a user
I want to recover my password
So that I can regain access to my account

Scenario: Solicitud de recuperación de contraseña
Given a user has forgotten their password
When they request a password reset via "/forgot-password"
Then the system should send a password reset email

Examples:
| email           |
| user1@example.com |
| user2@example.com |
```

## US36 – RESTful API Creación de sesiones

```
Feature: Creación de sesiones en el sistema
As a user
I want to create sessions via API
So that I can interact with the system in a stateful way

Scenario: Creación de sesión exitosa
Given a user provides valid session data
When they submit a request to "/create-session"
Then the system should create a new session and return status 201

Examples:
| session_data    |
| valid_session   |
| invalid_session |
```

## US37 - Paginación y filtrado de resultados

Feature: Paginación y filtrado de resultados en API

As a developer

I want to paginate and filter results

So that users can retrieve data efficiently

Scenario: Paginación de resultados

Given there are multiple users in the system

When the user requests page 2 with 10 results per page

Then the system should return the correct set of results

Scenario: Filtrado de resultados por rol

Given users with different roles exist

When the user filters by role "admin"

Then only users with the "admin" role should be returned

Examples:

page	results_per_page	role
1	10	admin
2	5	user

## US05 - Generación de correo temporal con un clic

Feature: Generación de correos temporales

As a user

I want to generate a temporary email address with one click

So that I can use it for short-term purposes

Scenario: Generar correo temporal

Given a user is on the email generation page

When they click on "Generate"

Then a new temporary email should be created instantly

Examples:

action
Generate

## US06 - Duración específica del correo temporal

Feature: Especificar la duración de correos temporales

As a user

I want to specify the duration of my temporary email

So that the email expires after a set period

Scenario: Seleccionar duración de expiración personalizada

Given a user is creating a temporary email

When they select a duration of 30 minutes

Then the system should set the email expiration time to 30 minutes

Examples:

duration
10 min
30 min

#### US07 - Confirmación visual de creación de correo

Feature: Confirmación visual de correos temporales

As a user

I want to see a visual confirmation when my temporary email is created  
So that I know the process was successful

Scenario: Mostrar confirmación visual

Given a user generates a temporary email  
When the email is created  
Then the system should display a visual confirmation message

Examples:

status
created
failed

#### US08 - Personalización del dominio del correo temporal

Feature: Personalización del dominio de correos temporales

As a user

I want to personalize the domain of my temporary email  
So that I can use a domain of my choice

Scenario: Cambiar el dominio del correo

Given a user is creating a temporary email  
When they select a different domain from a list  
Then the email should be created with the chosen domain

Examples:

domain
example.com
tempmail.net

### 1.2.1.5 Execution Suite for Sprint Review

Durante este primer sprint se realizó la implementación de las screen y la conexión con el backend desde la aplicación móvil. Features:

- Se implementó un sistema de inicio y registro de usuarios.
- Se implementó el sistema de generación de correos falsos
- Se implementó la vista de correos falsos y los mensajes de inbox.

En las imágenes y videos adjuntos se podrá apreciar de mejor manera los avances mencionados.

### Implementación del sistema de correos falso

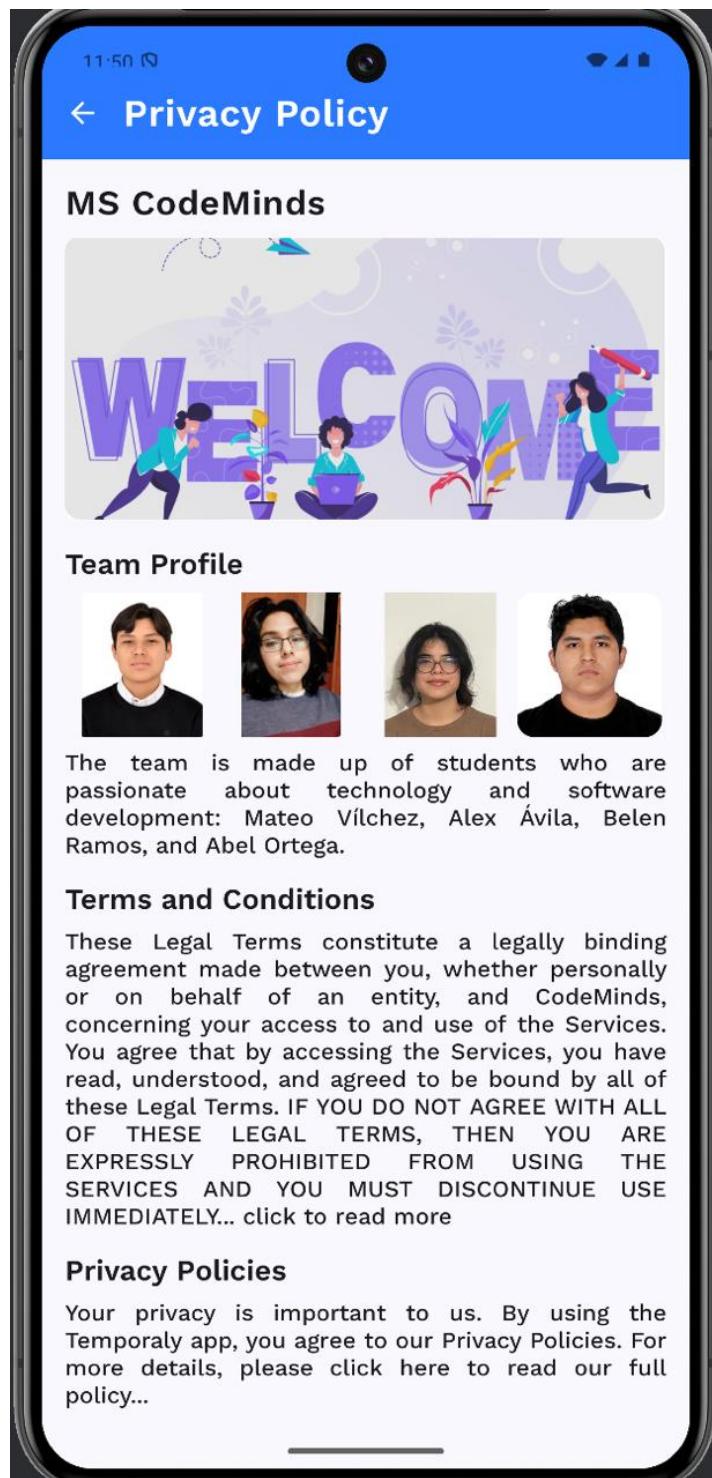
The screenshot shows the API documentation for Temporaly - TempMailAPI version 1.0.0. It includes sections for Email, Inbox, Messages, and Attachments, each listing specific endpoints with their methods, URLs, descriptions, and corresponding Python code snippets.

- Email**:
  - POST /api/v1/create\_email** Create a temporary email address. (post\_api\_v1\_create\_email)
- Inbox**:
  - GET /api/v1/inbox** Get the inbox messages of the temporary email. (get\_api\_v1\_inbox)
- Messages**:
  - GET /api/v1/messages/{message\_id}** Read a specific message by ID. (get\_api\_v1\_messages\_\_message\_id\_)
- Attachments**:
  - GET /api/v1/messages/{message\_id}/attachments/{filename}** Download an attachment from a message. (get\_api\_v1\_messages\_\_message\_id\_\_attachments\_\_filename\_)

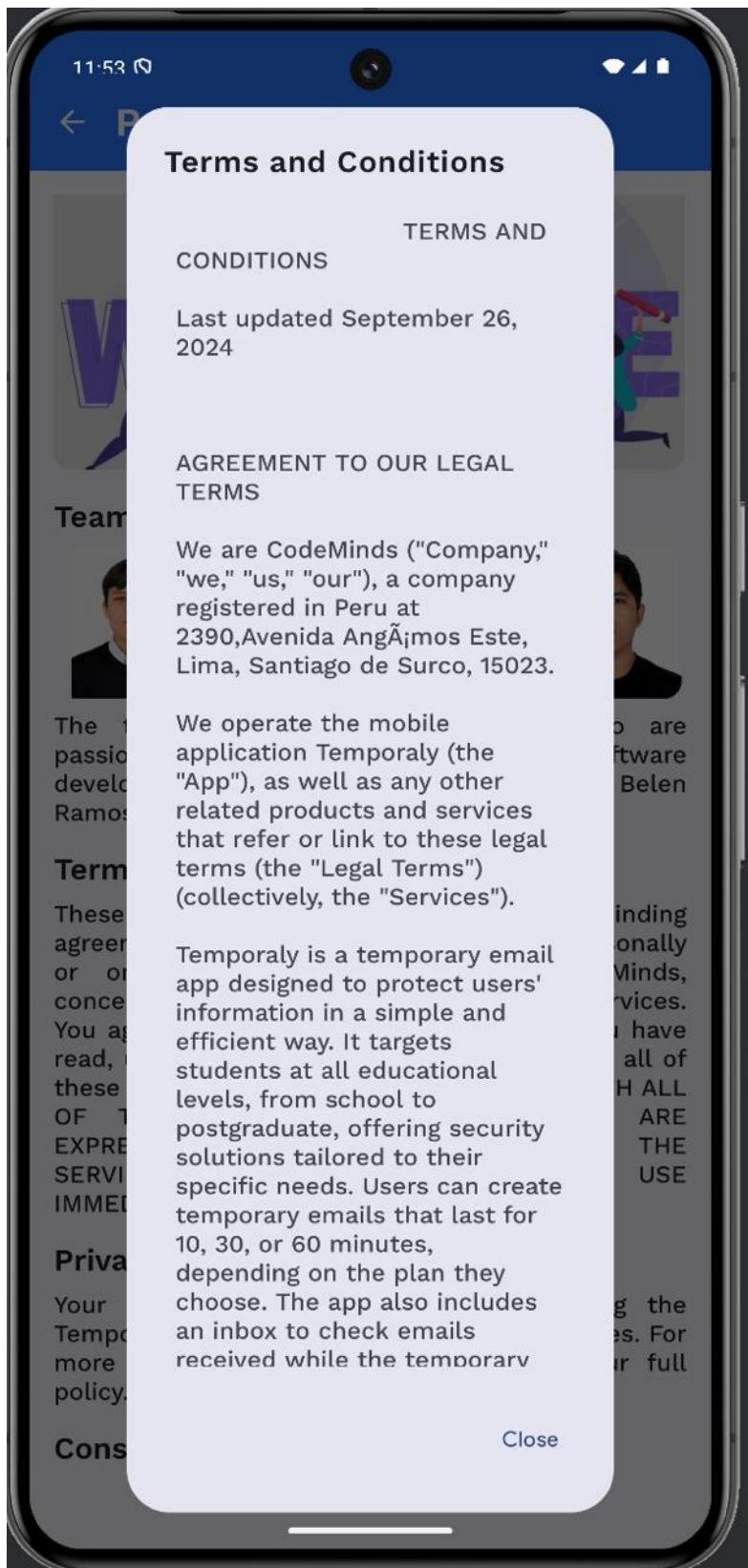
[Powered by Flasgger 0.9.7.1]

**Figura 1.17:** Backend: Imagen extraída del navegador

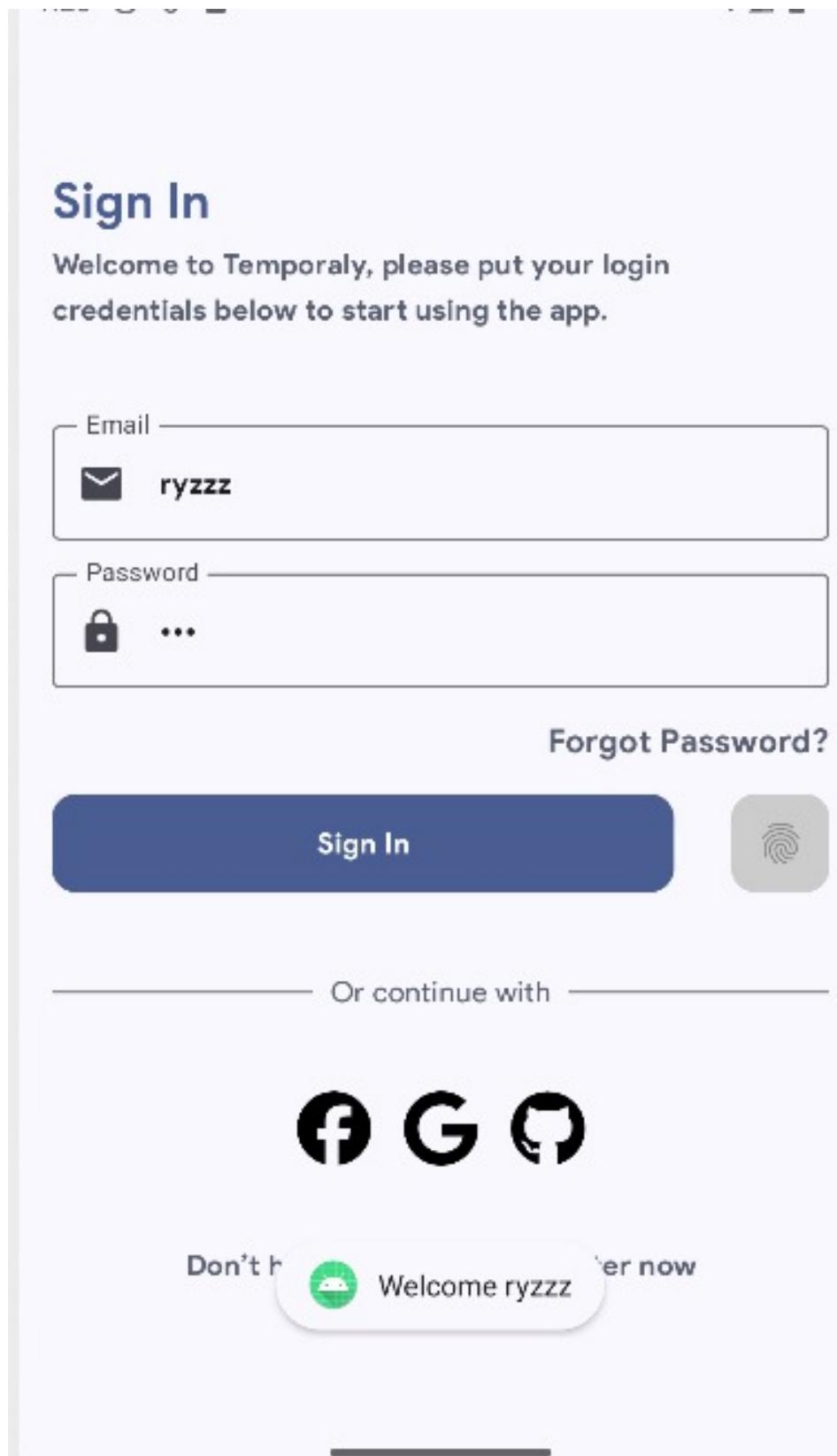
## Implementacion de las pantallas



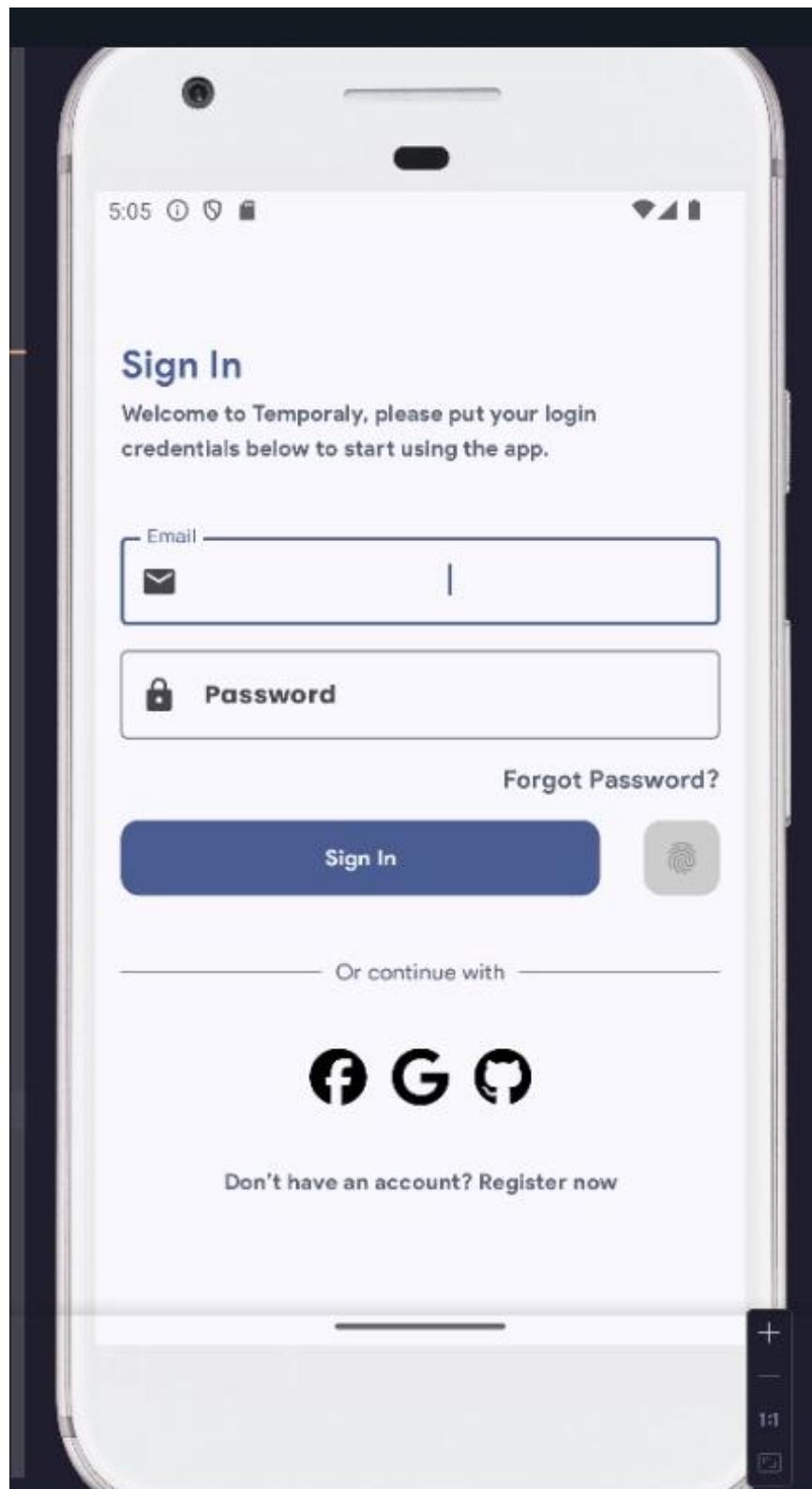
**Figura 1.18:** App móvil: Imagen extraída de Android Studio



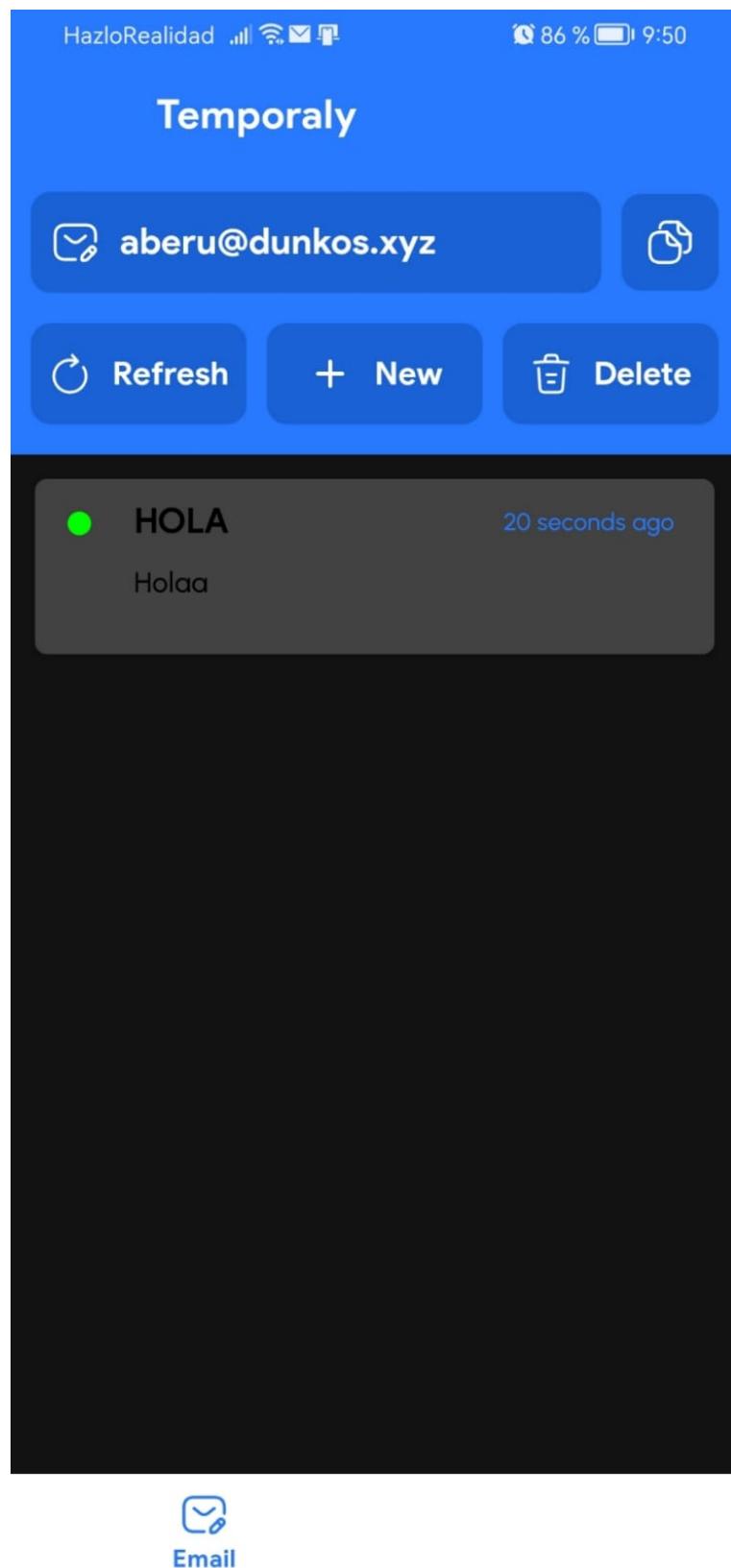
**Figura 1.19:** App móvil: Imagen extraída de Android Studio.



**Figura 1.20:** App móvil: Imagen extraída de Android Studio.



**Figura 1.21:** App móvil: Imagen extraída de Android Studio.



**Figura 1.22:** App móvil: Imagen extraída de Android Studio.

### 1.2.1.6 Services Documentation Evidence for Sprint Review

En esta sección, se muestra la documentación de los servicios del *backend* de la aplicación móvil, así como también se muestra el uso de Swagger para la documentación de los servicios.

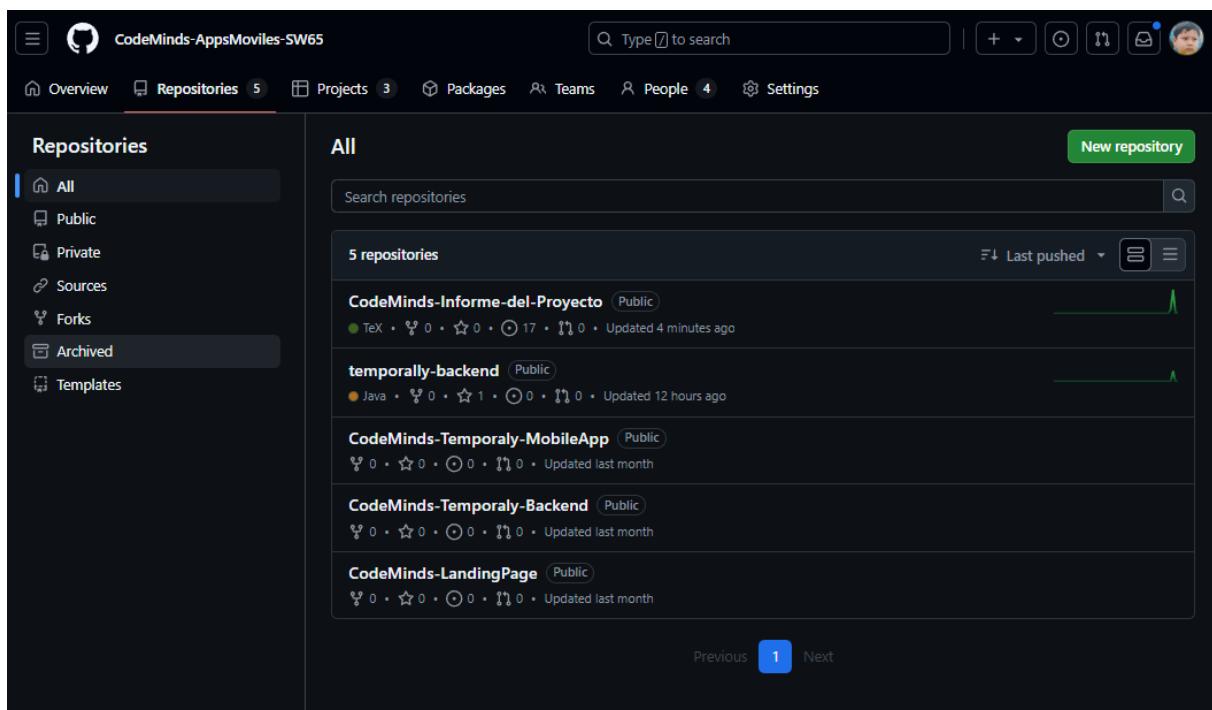
Utilizamos *SpringBoot*, *Spring Security*, *Spring Mongo*, *Spring Oauthclient*, *JWTToken*, *Spring Data* y *Flask*

- <https://docs.spring.io/spring-boot/index.html>
- <https://docs.spring.io/spring-boot/reference/web/spring-security.html#page-title>
- <https://www.npmjs.com/package/jsonwebtoken>
- <https://spring.io/projects/spring-data-jpa>
- <https://github.com/flasgger/flasgger>

### 1.2.1.7 Software Deployment Evidence for Sprint Review

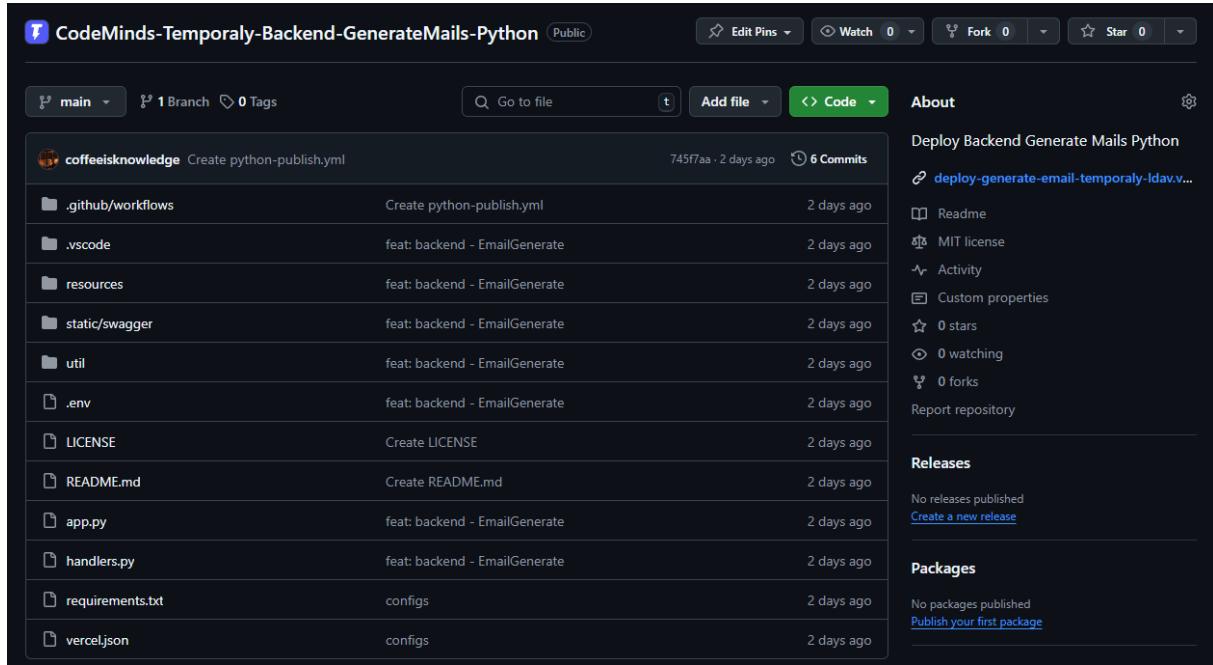
#### Desplegar Email Generate - Backend

1. Seleccionar el repositorio *CodeMinds-Temporaly-Backend-GenerateMails-Python*



**Figura 1.23:** Organización CodeMinds, imagen extraída de Github

## 2. Abrir, descargar y ejecutar en VSCode, Vim o su editor de preferencia



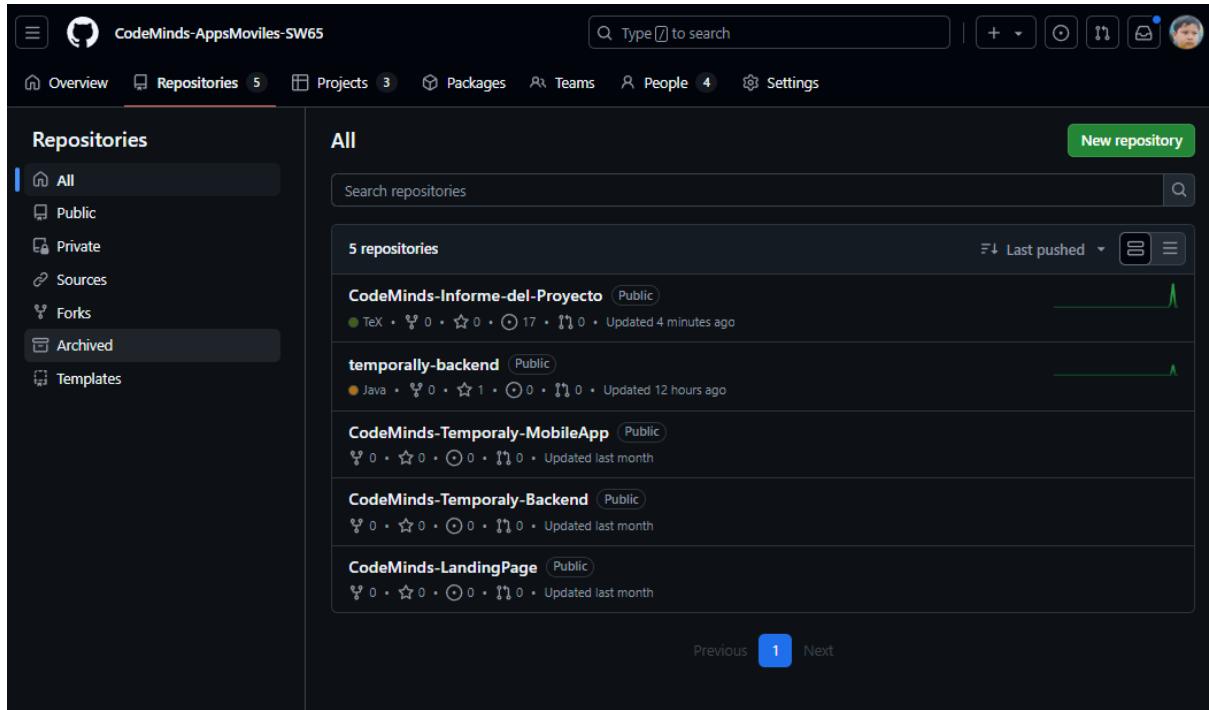
**Figura 1.24:** Repositorio del backend, imagen extraída de Github

## 3. Visualización de la documentación Flasgger

**Figura 1.25:** Flasgger, imagen extraída de Github

### Desplegar Mobile App - Temporaly

1. Seleccionar el repositorio *CodeMinds-Temporaly-MobileApp*



**Figura 1.26:** Organización CodeMinds, imagen extraída de Github

### 1.2.1.8 Team Collaboration Insights during Sprint

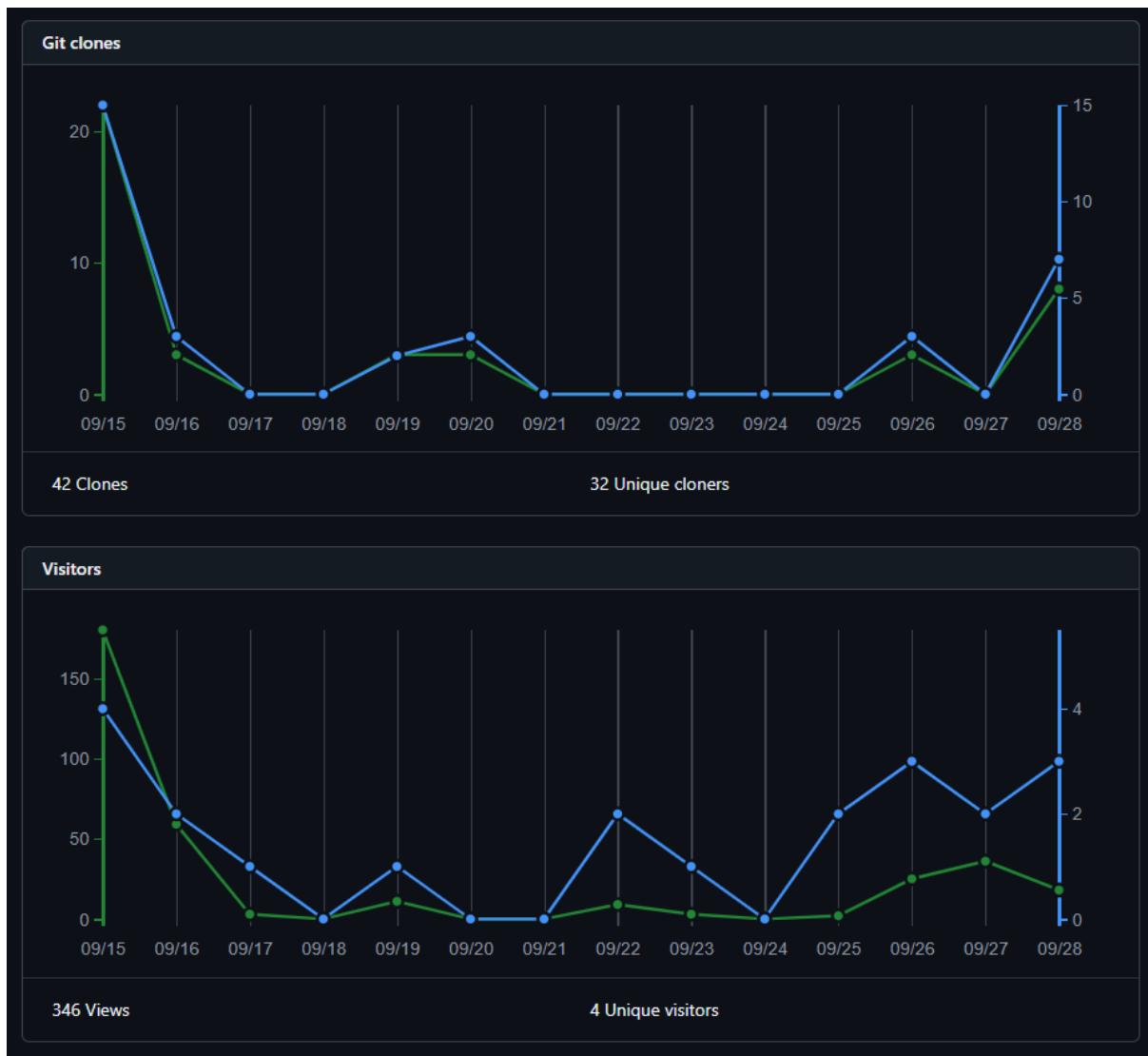
Para acceder a la visualización del flujo del trabajo, haga click en la [URL](#)

The screenshot shows a Kanban board interface with the following details:

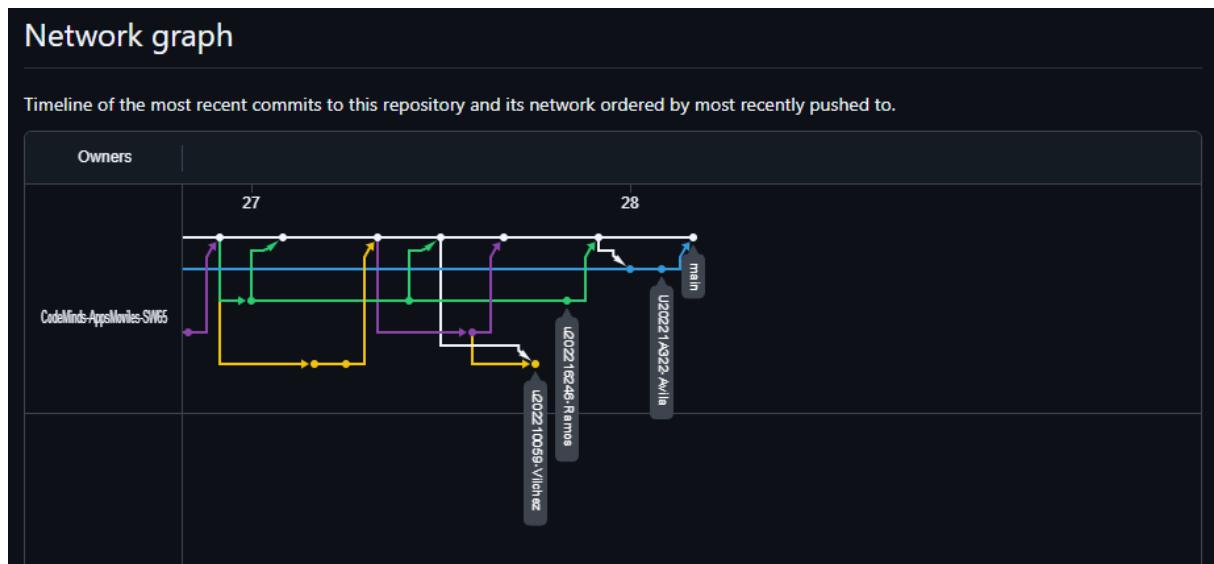
- Todo Column:** 0 items, Estimate: 0. Subtext: "This item hasn't been started".
- In Progress Column:** 0 items, Estimate: 0. Subtext: "This is actively being worked on".
- Done Column:** 55 items, Estimate: 648. Subtext: "This has been completed".
  - Item 1: CodeMinds-Informe-del-Proyecto #53, 3.1.4. Mobile Applications UX/UI Design, S, 72, P0, Iteration 2.
  - Item 2: CodeMinds-Informe-del-Proyecto #54, 3.1.4.1. Mobile Applications Wireframes, XL, 72, P2, Iteration 2.
  - Item 3: CodeMinds-Informe-del-Proyecto #55, 3.1.4.2. Mobile Applications Wireflow Diagrams, L, 72, P2, Iteration 2.
  - Item 4: CodeMinds-Informe-del-Proyecto #56, 3.1.4.3. Mobile Applications Mock-ups, L, 48, P1, Iteration 2.
  - Item 5: CodeMinds-Informe-del-Proyecto #57, 3.1.4.4. Mobile Applications User Flow Diagrams, XL, 48, P2, Iteration 2.
  - Item 6: CodeMinds-Informe-del-Proyecto #58, 3.1.4.5. Mobile Applications Prototyping, L, 48, P2, Iteration 2.

At the bottom of each column are "+ Add item" buttons.

**Figura 1.27:** Imagen extraída de Github - Codeminds



**Figura 1.28:** Imagen extraída de Github - Codeminds



**Figura 1.29:** Imagen extraída de Github - Codeminds

### 1.3 Validation Interviews

Las entrevistas de validación son una herramienta clave en el proceso de diseño centrado en el usuario. A través de estas entrevistas, se busca obtener información directa de los usuarios sobre cómo interactúan con el producto, lo que permite validar su usabilidad y funcionalidad. Este método es esencial para identificar problemas, detectar oportunidades de mejora y asegurar que el producto final cumpla con las expectativas del público objetivo. Las entrevistas de validación no solo ayudan a mejorar la experiencia del usuario, sino que también garantizan que las decisiones de diseño estén basadas en datos reales y no en suposiciones.

A continuación, exploraremos cómo se diseñaron las entrevistas, cómo se llevó a cabo el registro de las mismas y las evaluaciones realizadas basadas en principios heurísticos.

#### 1.3.1 Diseño de entrevistas

**Segmento Objetivo 1:** Estudiantes de nivel escolar

##### Evaluar la claridad del mensaje y propósito de la app:

- 1) ¿Qué te transmite el mensaje “Crea, Usa y Olvida ;)?”?
- 2) ¿Comprendes lo que significa proteger tu privacidad en línea con Temporaly? ¿Qué crees que hace la app al ver la primera sección?
- 3) Si tuvieras que explicar esta app a un amigo, ¿cómo lo harías?
- 4) ¿Cómo usarías Temporaly en tu vida diaria?

##### Encontrar fácilmente las funcionalidades clave:

- 1) ¿Sabes cómo crear un correo temporal rápidamente? ¿Te resulta claro cómo funciona?
- 2) ¿Te parece fácil entender los diferentes planes que ofrece la app (Básico, Pro, Master)?
- 3) Confianza en la seguridad que ofrece la app:
- 4) Después de ver la página, ¿sientes que tus datos estarían seguros si usas esta app? ¿Qué parte te hace sentir más seguro o inseguro?

##### Sugerencias para mejorar el diseño o la usabilidad:

- 1) ¿El diseño de la página es claro y fácil de navegar?
- 2) ¿Qué cambiarías para hacer la página más comprensible o atractiva para ti?

**Segmento Objetivo 2:** Estudiantes de nivel universitario y superior (pregrado - postgrado)**Evaluar la claridad del mensaje y propósito de la app:**

- 1) ¿El mensaje “Protege tu Privacidad en Línea con Temporaly” te da una idea clara de lo que hace la app? ¿Por qué sí o por qué no?
- 2) ¿Cómo relacionarías esta funcionalidad con situaciones como registrar correos en plataformas académicas o eventos?
- 3) ¿Qué tipo de problemas crees que resuelve Temporaly? ¿Es claro para ti cómo usarla?
- 4) ¿Cómo te imaginas utilizando Temporaly para proteger tu información personal en situaciones reales (registro de eventos, estudios, etc.)?

**Encontrar fácilmente las funcionalidades clave:**

- 1) ¿Es fácil para ti entender cómo crear un correo temporal y gestionar tu historial? ¿Consideras útil la barra lateral que menciona el historial de correos usados?
- 2) ¿Te parece clara la diferencia entre los planes de pago y el plan gratuito?

**Confianza en la seguridad que ofrece la app:**

- 1) Después de navegar por la página, ¿qué tan seguro te sientes respecto a la protección de tu información personal?
- 2) ¿Consideras que hay elementos adicionales que podrían incrementar tu confianza en la seguridad de la app?

**Sugerencias para mejorar el diseño o la usabilidad:**

- 1) ¿Cómo valorarías el diseño en cuanto a facilidad de uso y navegación?
- 2) ¿Aregarías o cambiarías algo en la estructura o la interfaz para hacerla más efectiva?

### 1.3.2 Registro de entrevistas



Para acceder al video de las entrevistas, haga click en la [URL](#)

#### Segmento Objetivo 1: Estudiantes de nivel escolar

##### Entrevista #1

<b>Nombre y Apellido</b>	Pol Adriano Ramos Rios
<b>Edad</b>	14 años
<b>Ubicación geográfica</b>	Trujillo, Perú
<b>Grado</b>	Secundaria
<b>Tiempo de entrevista</b>	inicio - fin

##### Resumen de la entrevista

En la entrevista con Adriano Ramos Ríos sobre la aplicación *Temporaly*, se le pidió que evaluara distintos aspectos clave de la landing page, centrada en la creación de correos electrónicos temporales para proteger la privacidad en línea. Durante la entrevista, Adriano expresó que el eslogan principal de la aplicación, “*Crea, Usa y Olvida*”, le transmitía la idea de generar correos temporales para usarlos en situaciones donde no se desea emplear el correo personal, lo que le resultaba útil para evitar el riesgo de utilizar su correo personal en sitios web de dudosa confianza. En su opinión, la idea de usar correos momentáneos para verificar cuentas era clara y directa.

Adriano manifestó que entendía bien el propósito de la aplicación y que estaba diseñada para proteger la privacidad en línea, algo que considera que la landing page comunica de manera efectiva. Sin embargo, sugirió que sería útil agregar una descripción más detallada para los usuarios que no estén familiarizados con conceptos relacionados a la privacidad digital, de manera que el mensaje sea aún más claro para todos los públicos. Al preguntársele cómo explicaría la aplicación a un amigo, mencionó que la describiría como una herramienta útil para acceder a sitios web sin comprometer la información personal. Específicamente, explicó que *Temporaly* permite la creación de correos momentáneos que ayudan a verificar cuentas sin necesidad de exponer el correo principal, algo que él mismo usaría para obtener descuentos o probar aplicaciones sin tener que proporcionar información personal.

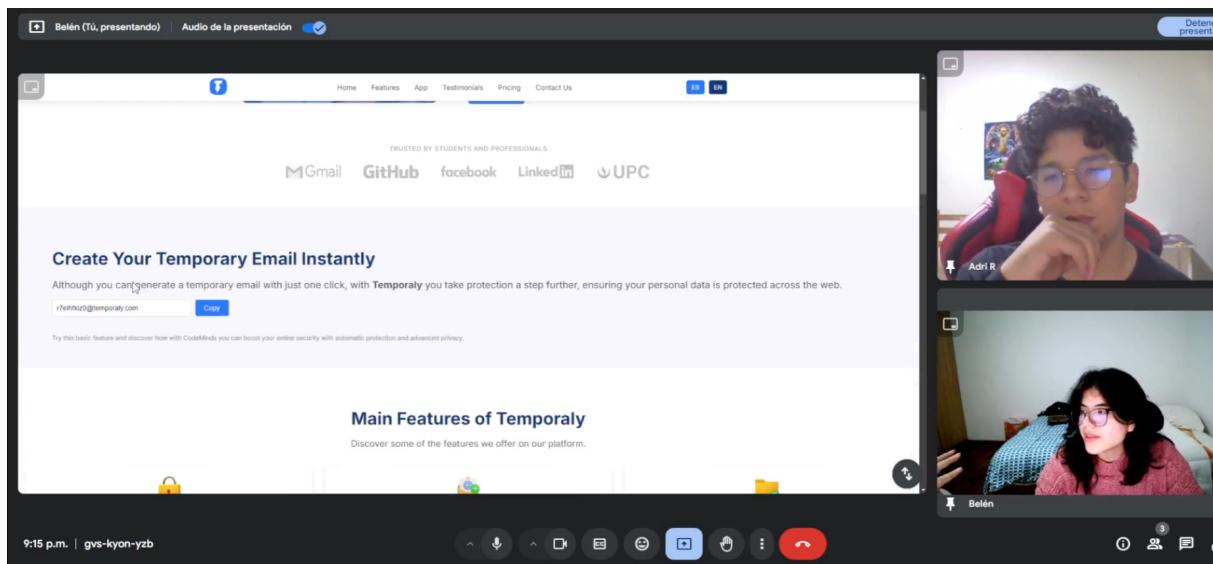
En cuanto a la usabilidad, Adriano indicó que le resultaría muy fácil usar la aplicación para crear correos temporales, pues ya tiene experiencia previa en el uso de servicios similares. No tuvo dificultades para comprender esta funcionalidad, lo que sugiere que el diseño es lo suficientemente intuitivo para

usuarios con conocimientos básicos. Respecto a los planes de pago (Básico, Pro y Máster), destacó que entendía bien las diferencias entre ellos, comentando que el plan Básico tiene un tiempo limitado de uso, mientras que los planes Pro y Máster ofrecen mayores ventajas como tiempos más largos y la posibilidad de crear un número ilimitado de correos. Esto le pareció claro y considera que la elección del plan depende de las necesidades particulares de cada usuario.

Cuando se abordó el tema de la seguridad, Adriano manifestó que, tras revisar la página, sentía un alto nivel de confianza en la aplicación, especialmente al ver que está asociada con entidades reconocidas como *Gmail*, *LinkedIn* y la Universidad Privada de Ciencias Aplicadas (UPC). Estas asociaciones le transmitieron una sensación de seguridad, ya que para él son indicativos de que *Temporaly* es una aplicación seria y confiable. A pesar de esta confianza inicial, comentó que preferiría probar la aplicación antes de confiar completamente en ella. En cuanto a los elementos que más le transmitieron seguridad, destacó la sección donde se mencionan las asociaciones con empresas y universidades, lo cual le generó una impresión positiva. No identificó ninguna parte de la página que le hiciera sentir inseguro.

Finalmente, en relación al diseño de la página, Adriano opinó que es claro y fácil de navegar, tanto en su versión para escritorio como en dispositivos móviles. Afirmó que no tuvo dificultades para encontrar las secciones clave, como los testimonios, los precios y el apartado de contacto. Sin embargo, sugirió incluir un video explicativo que guíe a los usuarios a través de las funcionalidades de la app, especialmente a aquellos que no son tan tecnológicos, ya que un recurso visual podría facilitar la comprensión de cómo utilizar *Temporaly* y sacarle el máximo provecho.

En resumen, Adriano considera que la aplicación tiene un buen nivel de claridad en cuanto a su mensaje y propósito, además de ser fácil de usar para aquellos con algo de experiencia. La confianza que genera la app es alta, aunque recomendó probarla para asegurarse de su eficacia. Su única sugerencia significativa fue agregar un video tutorial para mejorar la accesibilidad y la comprensión de la herramienta para todo tipo de usuarios.



**Figura 1.30:** Imagen extraída del video de entrevistas

**Segmento Objetivo 2:** Estudiantes de Pregrado/Postgrado**Entrevista #1**

<b>Nombre y Apellido</b>	Daniel Mateo del Castillo
<b>Edad</b>	19 años
<b>Ubicación geográfica</b>	Lima, Perú
<b>Grado</b>	Pregrado
<b>Tiempo de entrevista</b>	inicio - fin

### 1.3.3 Evaluaciones segun heurísticas

#### **UX Heuristics & Principles Evaluation Usability – Inclusive Design – Information Architecture**

<b>CARRERA:</b>	Ingeniería de Software
<b>CURSO:</b>	Aplicaciones para Dispositivos Móviles
<b>SECCIÓN:</b>	SW65
<b>PROFESORES:</b>	Jorge Luis Mayta Guillermo
<b>AUDITOR:</b>	CodeMinds
<b>CLIENTE(S):</b>	Estudiantes de nivel escolar y estudiantes de Pregrado/Postgrado

---

**SITE o APP A EVALUAR:** *Temporaly*

**TAREAS A EVALUAR:**

El alcance de esta evaluación incluye la revisión de la usabilidad de la **landing page** de Temporaly, con foco en los siguientes elementos:

1. **Comunicación del propósito de la app:** Claridad del mensaje en cuanto a la creación de correos temporales.
2. **Acceso a planes de pago:** Evaluación de la presentación y diferencias entre los planes (Básico, Pro y Máster).
3. **Confianza y seguridad:** Percepción de seguridad en la landing page.
4. **Accesibilidad para usuarios nuevos:** Comprensión de la información por parte de usuarios no técnicos.
5. **Diseño visual y navegación:** Facilidad para encontrar secciones clave.

## ESCALA DE SEVERIDAD

Los errores serán puntuados tomando en cuenta la siguiente escala de severidad:

Nivel	Descripción
1	Problema superficial: puede ser fácilmente superado por el usuario o ocurre con muy poca frecuencia. No necesita ser arreglado a no ser que exista disponibilidad de tiempo.
2	Problema menor: puede ocurrir un poco más frecuentemente o es un poco más difícil de superar para el usuario. Se debería asignar una prioridad baja resolverlo de cara al siguiente release.
3	Problema mayor: ocurre frecuentemente o los usuarios no son capaces de resolverlo. Es importante que sean corregidos y se les debe asignar una prioridad alta.
4	Problema muy grave: un error de gran impacto que impide al usuario continuar con el uso de la herramienta. Es imperativo que sea corregido antes del lanzamiento.

## TABLA RESUMEN

#	Problema	Escala de severidad	Heurística/Principio violado
1	Falta de descripción detallada sobre privacidad	2	Usabilidad: Concordancia entre el sistema y el mundo real
2	Dificultad para entender los planes de pago	3	Usabilidad: Simplicidad y naturalidad de uso
3	Ausencia de video tutorial en la landing page	2	Usabilidad: Ayuda y documentación
4	Falta de elementos de confianza visual	3	Usabilidad: Confianza y seguridad

## DESCRIPCIÓN DE PROBLEMAS

- **PROBLEMA #1: Falta de descripción detallada sobre privacidad**

**Severidad:** 2

**Heurística violada:** Usabilidad – Concordancia entre el sistema y el mundo real

**Problema:** Aunque el mensaje principal es claro, falta una explicación más detallada sobre cómo la app protege la privacidad de los usuarios, lo que puede generar dudas en personas menos familiarizadas con estos temas.

**Recomendación:** Añadir una sección que explique en detalle la importancia de la privacidad y cómo Temporaly la garantiza.

- **PROBLEMA #2: Dificultad para entender los planes de pago**

**Severidad:** 3

**Heurística violada:** Usabilidad – Simplicidad y naturalidad de uso

**Problema:** Las diferencias entre los planes de pago no son suficientemente claras para los usuarios nuevos, lo que puede llevar a confusión al elegir entre el plan Básico, Pro y Máster.

**Recomendación:** Clarificar las ventajas de cada plan mediante comparaciones visuales, destacando las diferencias clave de forma más evidente.

- **PROBLEMA #3: Ausencia de video tutorial en la landing page**

**Severidad:** 2

**Heurística violada:** Usabilidad – Ayuda y documentación

**Problema:** Adriano sugirió que un video tutorial ayudaría a usuarios menos experimentados a entender mejor cómo funciona Temporaly, especialmente para aquellos que no están familiarizados con la creación de correos temporales.

**Recomendación:** Incluir un breve video explicativo en la landing page que muestre cómo usar la app paso a paso.

- **PROBLEMA #4: Falta de elementos de confianza visual Severidad: 3**

**Heurística violada:** Usabilidad – Confianza y seguridad

**Problema:** Aunque se mencionan asociaciones con entidades como Gmail, LinkedIn y la UPC, falta un uso más prominente de estos elementos en la landing page para transmitir mayor confianza a usuarios nuevos.

**Recomendación:** Mostrar de manera más destacada los logos y asociaciones de seguridad con terceros en la landing page, asegurando que estos elementos visuales refuerzen la confianza del usuario.