

Desarrollo de una red social para fomentar la conciencia ambiental en los individuos de una sociedad moderna

Development of a social network to promote environmental awareness among individuals in a modern society

Alejandro J. Ávila Amesty

Universidad Rafael Urdaneta, Facultad de Ingeniería, Escuela de Ingeniería de Computación.
Maracaibo, Venezuela.

 <https://orcid.org/0009-0001-2437-0298> / Correo electrónico: avilaalejandro177@gmail.com

José A. Chacón Piña

Universidad Rafael Urdaneta, Facultad de Ingeniería, Escuela de Ingeniería de Computación.
Maracaibo, Venezuela.

 <https://orcid.org/0009-0003-1784-252X> / Correo electrónico: joseandres102003@gmail.com

Recibido: 05/10/2024 Admitido: 10/01/2025 Aceptado: 08/06/2025

Resumen

El presente trabajo tuvo como objetivo desarrollar un sistema basado en esquemas de red social para fomentar y canalizar la conciencia ambiental en la sociedad, atendiendo la necesidad de reducir la contaminación y mejorar la salud pública en el estado Zulia. La investigación combinó elementos aplicados y descriptivos con un enfoque proyectivo, utilizando técnicas no experimentales y documentales para el diseño del sistema, complementadas con la revisión documental y otras fuentes de datos para recolectar la información necesaria. El sistema se desarrolló siguiendo el ciclo de vida del desarrollo de software (SDLC) con modelos incremental e iterativo, empleando frameworks como Vue e Ionic para la interfaz, Deno para el servidor HTTP, y PostgreSQL para la base de datos. Culminó con la creación de una herramienta funcional que promueve la participación comunitaria en reciclaje y reducción de contaminación, impactando positivamente en la sociedad.

Palabras clave: Red social, conciencia ambiental, contaminación, reciclaje, desarrollo de software, Vue, Ionic, Deno, PostgreSQL.

Abstract

This special undergraduate thesis aimed to develop a system based on social network models to raise and channel environmental awareness in society, addressing the need to reduce pollution and improve public health in the state of Zulia. The research combined applied and descriptive elements with a projective approach, using non-experimental and documentary techniques for system design, complemented by documentary review and other data sources to gather the necessary information. The system was developed following the software development life cycle (SDLC) with incremental and iterative models, employing frameworks such as Vue and Ionic for the user interface, Deno for the HTTP server, and PostgreSQL for the database. It culminated in the creation of a functional tool that promotes community participation in recycling and pollution reduction, making a positive impact on society.

Keywords: Social network, environmental awareness, pollution, recycling, software development, Vue, Ionic, Deno, PostgreSQL.

Introducción

En la actualidad, los desafíos ambientales globales, especialmente en relación con el reciclaje y la contaminación, han alcanzado niveles críticos que demandan intervenciones efectivas y sostenibles. La creciente acumulación de residuos sólidos y la falta de conciencia ambiental han exacerbado la degradación del

medio ambiente, afectando tanto la calidad de vida de las comunidades como la salud pública. Estos problemas son evidentes en el estado Zulia, Venezuela, donde la insuficiente infraestructura y las prácticas inadecuadas de gestión de residuos contribuyen a una crisis ambiental preocupante.

Una red social es una estructura social formada por actores interconectados mediante relaciones definidas. Las redes sociales en línea han transformado la comunicación e interacción humana, facilitando el intercambio de información y la creación de comunidades en torno a intereses compartidos. Estas plataformas han revolucionado la difusión de contenido debido a su alcance masivo, características interactivas y su impacto sociocultural [1].

Este estudio tuvo como objetivo desarrollar una red social innovadora orientada a promover la conciencia ambiental y facilitar la participación activa de la comunidad zuliana en prácticas de reciclaje y reducción de la contaminación. La investigación se sustenta en principios de desarrollo de software y modelos de redes sociales, adoptando un enfoque metodológico que integra diseño, implementación y evaluación de la plataforma.

La relevancia de esta investigación radica en su potencial para abordar problemas ambientales locales a través de la tecnología y la educación comunitaria. Se espera que la red social no solo sensibilice a los usuarios sobre la importancia del reciclaje y la reducción de la contaminación, sino que también promueva un cambio cultural hacia prácticas más sostenibles y responsables. Además, este proyecto contribuirá al cuerpo de conocimiento en ingeniería de computación al explorar nuevas aplicaciones tecnológicas para la gestión ambiental y el desarrollo comunitario.

Metodología

Este estudio se enmarcó en una investigación de tipo aplicada y descriptiva, con un enfoque proyectivo. La investigación aplicada se centró en la aplicación práctica de conocimientos para abordar problemas específicos, en este caso, mejorar la conciencia ambiental a través de la tecnología. Por otro lado, la investigación descriptiva permitió comprender y describir las características técnicas y funcionales del sistema propuesto, utilizando análisis detallados y modelos [2, 3].

El diseño de la investigación adoptado combinó elementos del diseño no experimental y documental. El enfoque no experimental se empleó para observar el fenómeno del desarrollo del sistema en su entorno natural, sin manipular variables intencionalmente. Esto garantizó una comprensión auténtica de su funcionamiento y eficacia. Por otro lado, el diseño documental implicó una extensa búsqueda, análisis e interpretación de datos provenientes de fuentes documentales, como estudios previos y recursos técnicos relevantes [2, 3].

Para el desarrollo de la red social se emplearon técnicas y herramientas de recolección de datos como la revisión documental y la observación [3]. La observación se basó en experiencias personales e indagaciones realizadas a través de la investigación. Estas técnicas y herramientas fueron seleccionadas cuidadosamente para asegurar que la información recopilada fuera relevante, precisa y útil para el diseño e implementación de la red social.

El desarrollo del proyecto se orientó en un ciclo incremental e iterativo. Este modelo permitió definir fases secuenciales como la definición de alcance y requisitos, análisis detallado de requerimientos, diseño del sistema, desarrollo e implementación, pruebas exhaustivas y despliegue del prototipo en un entorno real, sin limitar que en algún momento se pueda devolver a una fase anterior para ajustar, adaptar y mejorar continuamente el sistema durante su desarrollo. Cada fase se completó de manera organizada, asegurando que el sistema fuera robusto, eficiente y alineado con los requisitos establecidos.

Resultados y discusión

En esta sección se presenta el análisis detallado del proceso de desarrollo de la plataforma de red social orientada a la conciencia ambiental en el estado Zulia. Se describen las etapas clave del proceso de desarrollo iterativo, destacando la contribución de cada ciclo al logro del objetivo principal de la investigación.

Fase 1. Declaración del alcance y requisitos

El desarrollo inicial del alcance y requisitos para la plataforma de red social, reveló una estructura integral y funcional diseñada para promover la participación activa y la educación en prácticas sostenibles.

El primer paso consistió en definir el público objetivo de la plataforma, considerando diversos factores demográficos, socioeconómicos y de interés ambiental. Este público estuvo conformado por individuos y organizaciones locales que buscan mejorar el medio ambiente. Para ello, la plataforma debe ofrecer una interfaz fácil de usar, accesible y atractiva, de modo que los usuarios puedan interactuar con eficiencia y encontrar la información de manera rápida y sencilla.

En términos de funcionalidad, la interfaz intuitiva y accesible asegura una experiencia de usuario optimizada, facilitando la navegación y la interacción sin complicaciones. La inclusión de herramientas como el registro de usuarios, la personalización de perfiles y la capacidad de establecer conexiones entre usuarios fortalece el sentido de comunidad y colaboración, fundamentales para el éxito de la red social.

Destacaron también características específicas como la calculadora de huella de carbono y el medidor del buen impacto ambiental, herramientas educativas que permiten a los usuarios evaluar y visualizar el impacto ambiental de sus acciones cotidianas. Estas funciones no solo fomentan la conciencia ambiental, sino que también motivan a los usuarios a adoptar comportamientos más sostenibles.

La capacidad de reportar casos de contaminación y el desarrollo de un mapa interactivo de contaminación y botaderos de basura, constituyeron elementos clave para la vigilancia ambiental comunitaria. Estas herramientas promueven la identificación y resolución de problemas ambientales locales, incentivando la acción colectiva. Además, la facilitación de chats personales y grupales refuerzan la interacción social y la coordinación de esfuerzos ambientales, contribuyendo al fortalecimiento de la comunidad virtual.

La fase inicial del desarrollo reflejó una plataforma robusta y multifacética, diseñada para educar, motivar y movilizar a la comunidad hacia prácticas más sostenibles y la protección del medio ambiente en el estado Zulia. Estos hallazgos sentaron las bases para futuras etapas de implementación y evaluación del impacto de la red social en la conciencia y comportamiento ambiental de sus usuarios.

Fase 2. Análisis de requerimientos

En esta fase se llevó a cabo un análisis detallado de los requisitos funcionales y no funcionales necesarios para el desarrollo de la plataforma de red social. Estos requisitos funcionales incluyeron el registro de usuarios, creación de perfiles, seguimiento de usuarios, mensajería y publicación de contenido. Estas funcionalidades permitieron a los usuarios registrarse, compartir contenido, interactuar con otros y comunicarse de manera privada y grupal.

Para fomentar la conciencia ambiental, se incorporaron herramientas como una calculadora de huella de carbono, un medidor del buen impacto y la capacidad de reportar casos de contaminación. Estas herramientas permiten a los usuarios evaluar y reducir su impacto ambiental, así como reportar y visualizar problemas locales a través de un mapa interactivo de contaminación y botaderos de basura [4, 5].

Además de los requisitos funcionales, se identificaron los requisitos no funcionales esenciales para la eficacia del sistema, como la seguridad, el rendimiento, la escalabilidad, la usabilidad y la compatibilidad. La plataforma garantizó la protección de datos personales mediante autenticación segura y encriptación de datos, respondió de manera rápida y eficiente, y se adaptó a un creciente número de usuarios sin pérdida de rendimiento. La interfaz es intuitiva y accesible para todos los niveles de habilidad tecnológica y compatible, con una amplia gama de dispositivos y navegadores modernos.

La plataforma utilizó Deno con Express.js para el backend, Vue con Ionic para el frontend, y PostgreSQL como base de datos, aprovechando la eficiencia de estas tecnologías modernas. El entorno de desarrollo incluyó herramientas como Visual Studio Code y Git para el control de versiones. Los servidores de desarrollo y producción se configuraron con especificaciones adecuadas para asegurar un rendimiento óptimo;

mientras que, se implementaron soluciones de seguridad robustas y servicios en la nube para escalabilidad y disponibilidad.

Fase 3. Elaboración del diseño de la red social

En esta fase, se definieron las funcionalidades clave y se diseñaron los módulos necesarios para el desarrollo de la aplicación móvil [6], enfocados en promover la conciencia ambiental y fomentar la interacción social entre los usuarios. Dado que la aplicación es una red social, se priorizó la creación de herramientas que permitieran a los usuarios comprender y reducir su impacto ambiental, mientras interactúan con comunidades afines.

Un componente destacado es el cálculo de la huella de carbono [6], diseñado para proporcionar a los usuarios una estimación personalizada de sus emisiones de gases de efecto invernadero (GEI). Este cálculo se basó en un formulario que recogió información sobre hábitos cotidianos del usuario, como transporte, consumo de energía y patrones de compra.

Los datos fueron procesados utilizando factores de emisión específicos, categorizados en tres alcances principales:

- Emisiones directas (Alcance 1):

Incluyen el uso de vehículos personales, transporte público y viajes aéreos.

- Emisiones indirectas (Alcance 2):

Asociadas al consumo energético en el hogar, como el uso de aires acondicionados y cocinas.

- Otras emisiones indirectas (Alcance 3):

Relacionadas con el consumo de alimentos y bienes, como carne, productos enlatados y plástico de un solo uso.

Esta funcionalidad no solo buscó crear conciencia sobre el impacto ambiental, sino que también motivó a los usuarios a adoptar prácticas más sostenibles mediante la comparación de sus huellas con promedios globales o locales.

La arquitectura del sistema fue concebida con un enfoque por capas, permitiendo una organización clara de responsabilidades que simplificó tanto el mantenimiento como la evolución del software. En la Figura 1 se ilustra un esquema general de la arquitectura, destacando las tecnologías clave empleadas en el proyecto. Para la interfaz de usuario, se optó por Vue e Ionic, herramientas que garantizan un diseño adaptable y una experiencia interactiva.

En cuanto al backend, desarrollado con Deno, se implementaron subsistemas especializados para manejar funcionalidades esenciales como la autenticación, la gestión de usuarios y el soporte de chats en tiempo real a través de sockets. La seguridad en las sesiones fue respaldada por el uso de JWT (JSON Web Tokens), lo que reforzó la protección de los datos sensibles. Asimismo, se resaltó la integración con la base de datos PostgreSQL, gestionada a través del Query Builder Kysely, lo que aseguró un manejo de datos eficiente, confiable y escalable.

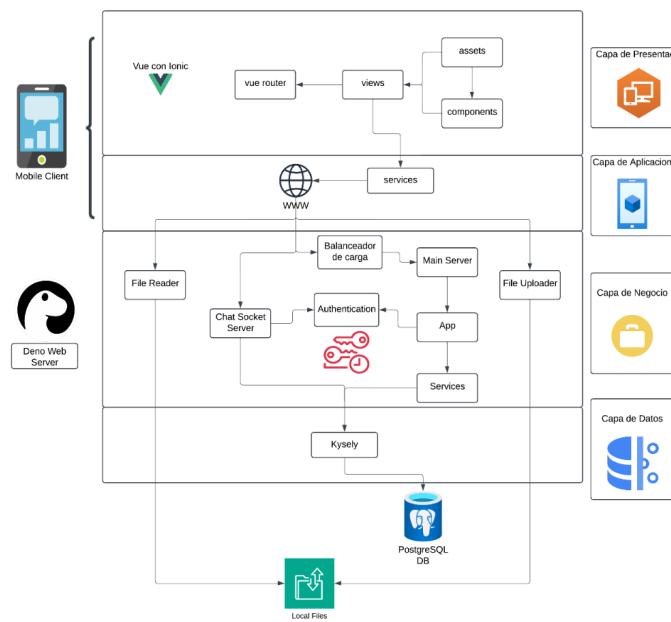


Figura 1. Diagrama de Arquitectura del sistema.

La representación de los componentes y entidades del proyecto a través del diagrama de arquitectura, sirvió como punto de partida para el diseño del modelo de datos del sistema (Figura 2). Se utilizó un enfoque relacional para estructurar los datos, asegurando tanto la integridad como la eficiencia en el manejo de la información. Este diseño se desarrolló con especial énfasis en la gestión de posts.

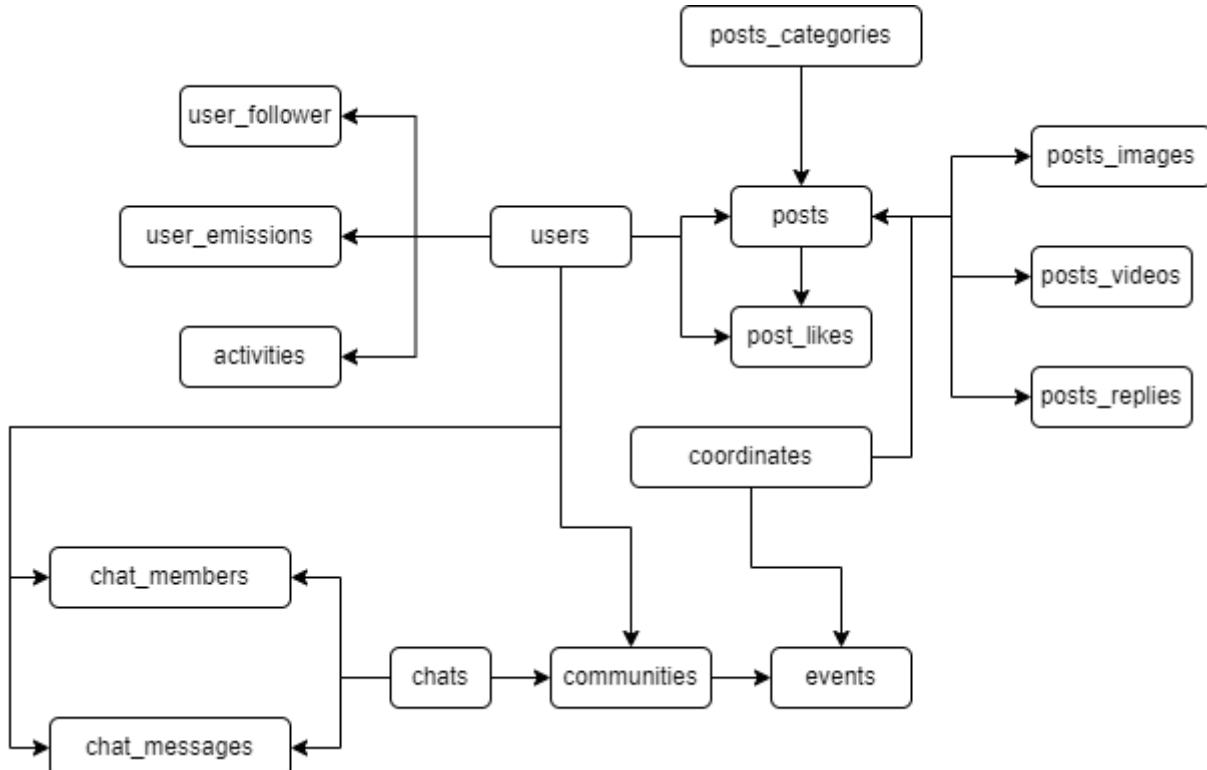


Figura 2. Diagrama de Modelo de Datos.

Este modelo de datos relacional se complementa y detalla en el diagrama de entidad-relación (DER) representado en la Figura 3. En dicho diagrama se especificaron cuidadosamente los campos de cada tabla, incluyen información como el tipo de dato. Además, se destacaron las llaves primarias y foráneas, definiendo con precisión las relaciones y dependencias entre las diferentes entidades del sistema.

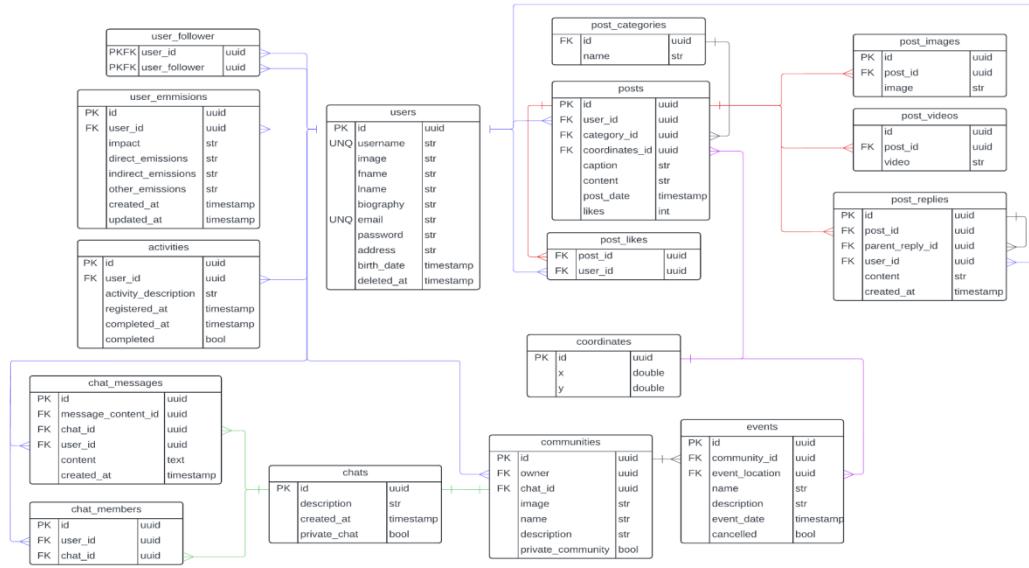


Figura 3. Diagrama de Entidad Relación.

En el diseño de la aplicación, los mockups (Figura 4) se agruparon en módulos que abarcaron las principales funcionalidades de la plataforma. Cada módulo buscó proporcionar una experiencia de usuario intuitiva y eficiente, enfocada en promover prácticas sostenibles y la interacción comunitaria.

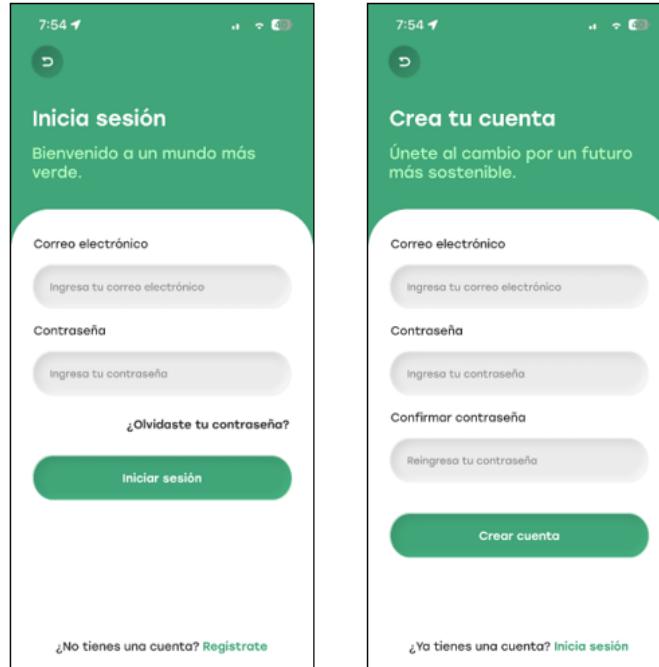


Figura 4. Mockups del módulo de Autorización

El módulo de autorización incluyó las pantallas de inicio de sesión y registro. La pantalla de inicio de sesión se centró en ofrecer un acceso seguro y eficiente a los usuarios, con un encabezado que da la bienvenida

mediante un mensaje motivador, seguido de campos de entrada para el correo electrónico y la contraseña. En la parte inferior, se encuentran opciones clave como la recuperación de contraseña y un botón destacado para iniciar sesión. Además, se incluyó un mensaje que invita al registro para nuevos usuarios. Por su parte, la pantalla de registro permite a los usuarios crear una cuenta con facilidad. El formulario incluyó campos para ingresar el correo electrónico, la contraseña y su confirmación, acompañados de placeholders que orientaron sobre la información requerida.

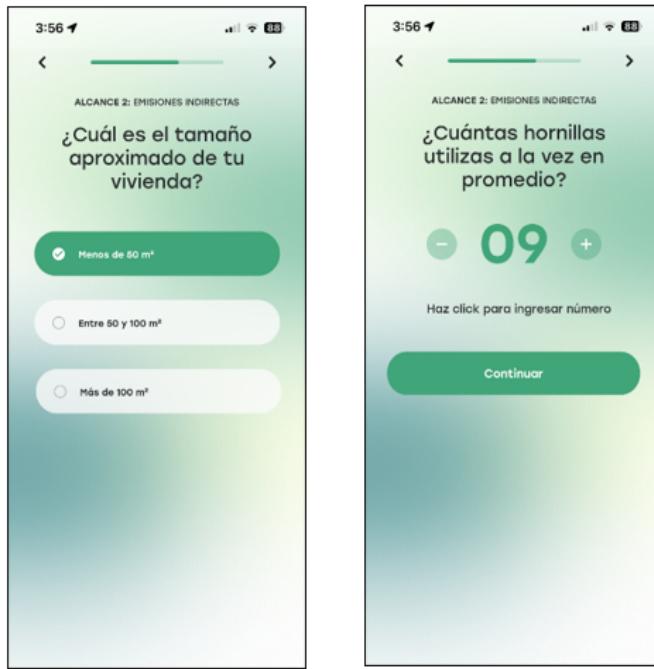


Figura 5. Mockups del módulo de Cálculo de huella de carbono

Este módulo ayudó a los usuarios a calcular su impacto ambiental (Figura 5) a través de un cuestionario dividido en tres alcances: emisiones directas, indirectas y otras. Cada sección incluyó una barra de progreso que indicó el avance, el alcance actual y la pregunta correspondiente. Las opciones de respuesta incluyeron botones de selección única o campos numéricos. La navegación se realizó mediante flechas o botones de continuar, dependiendo del tipo de pregunta. Este diseño garantizó una experiencia clara y enfocada para calcular la huella de carbono personal.



Figura 6. Mockups del módulo de Home

El módulo de inicio (Figura 6) fue diseñado para brindar una experiencia personalizada, con un encabezado que incluyó el nombre de la aplicación y un acceso rápido a la búsqueda. Los usuarios pudieron alternar entre dos feeds: "Descubre", que muestra contenido reciente y popular, y "Siguiendo", enfocado en publicaciones de amigos. Cada publicación resaltó el nombre del creador, su ícono, interacciones disponibles y una vista previa atractiva, asegurando una navegación atractiva y funcional.

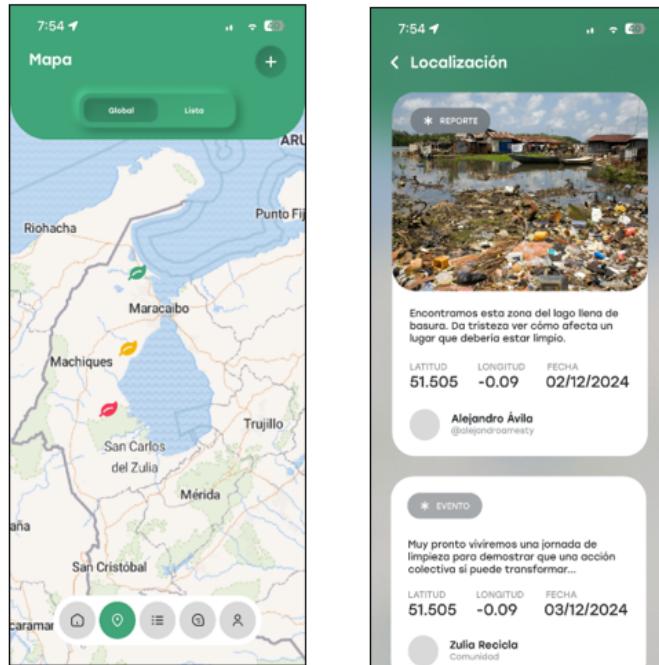


Figura 7. Mockups del módulo de Mapa

El módulo de mapa (Figura 7) permitió explorar y gestionar ubicaciones relacionadas con reportes y eventos. En la vista de mapa global, los usuarios pueden alternar entre las opciones de "Mapa global" y "Lista", visualizando marcadores que indican eventos, reportes o ambos. Los colores de los marcadores permitieron

identificar rápidamente la información. Al hacer clic en un marcador, se accedió a la pantalla de localización, donde se detalló la información asociada a esa ubicación, como reportes y eventos registrados. Además, la creación de eventos se simplificó al interactuar con el mapa o directamente desde la pantalla de localización.

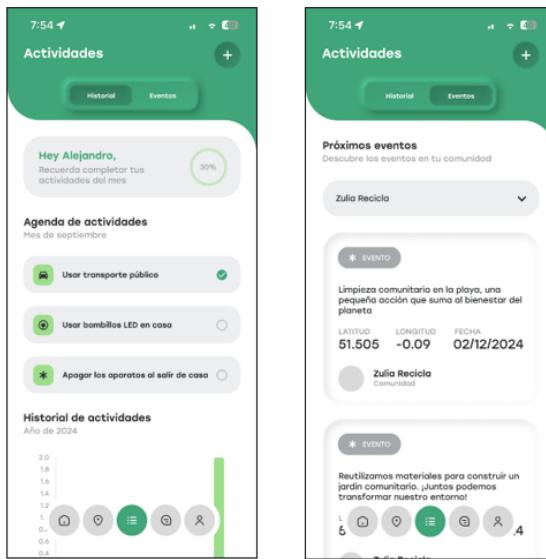


Figura 8. Mockups del módulo de Actividades

El módulo de actividades (Figura 8) organizó el progreso del usuario en prácticas ambientales a través de dos apartados: historial y eventos. En el historial, se mostró el progreso mensual con gráficos y una lista de actividades registradas, que los usuarios pueden marcar como completadas. Por otro lado, los eventos futuros fueron organizados según las comunidades, permitiendo explorar actividades relevantes con información clara sobre su ubicación, fecha y organizadores.

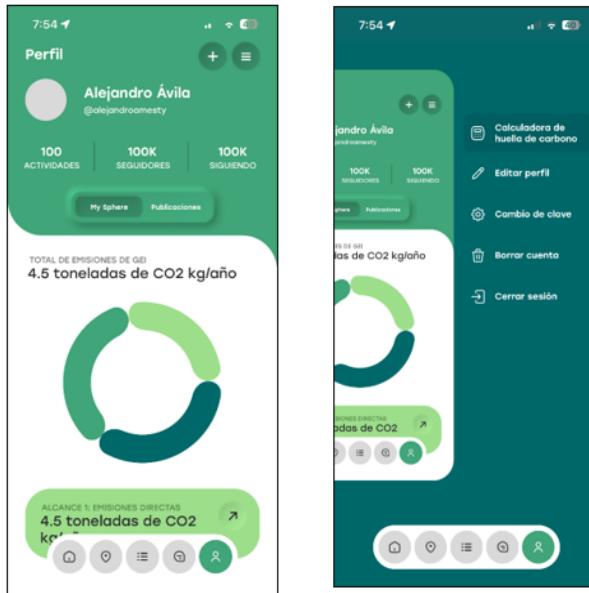


Figura 10. Mockups del módulo de Perfil.

El módulo de chats (Figura 9) facilitó la comunicación entre los usuarios con dos apartados principales: amigos y comunidades. En cada apartado, se presentaron vistas previas de conversaciones, que incluyeron íconos, nombres y últimos mensajes enviados, manteniendo una estructura consistente. Al acceder

a una conversación específica, la pantalla de chat mostró los mensajes de forma cronológica y permitió enviar mensajes mediante un campo de entrada ubicado en la parte inferior, optimizado para su uso con el teclado.

El módulo de perfil (Figura 10) ofreció una vista detallada de la información personal y las estadísticas del usuario. Incluyó un gráfico circular que mostró la distribución de las emisiones de gases de efecto invernadero y una sección de publicaciones propias. El menú lateral del perfil proporcionó accesos rápidos a funcionalidades clave como recalcular la huella de carbono, editar el perfil o gestionar la cuenta. Este diseño buscó centralizar la información y ofrecer un acceso rápido a herramientas de personalización.

Fase 4. Desarrollo de la red social

Durante la fase de desarrollo, se lograron hitos importantes al cumplir con las metas establecidas en las etapas de análisis y diseño. Este período fue esencial para convertir las ideas conceptuales y los diseños en elementos funcionales. Las tareas se abordaron de manera ordenada, siguiendo una secuencia lógica que permitió una ejecución fluida y sin contratiempos.

Para asegurar un manejo organizado y fluido de las tareas, se utilizó Trello como herramienta principal. Esto permitió una asignación efectiva de responsabilidades, brindando transparencia en el progreso de cada tarea y promoviendo la colaboración entre los miembros del equipo. Gracias a este enfoque, todas las actividades se completaron dentro del tiempo establecido.

Se adoptó una estructura de carpetas bien definida para el desarrollo del frontend y del backend, lo cual favoreció la organización del código, la mantenibilidad y el trabajo colaborativo. La aplicación de enfoques como el Modelo-Vista-Controlador y el Diseño Guiado por el Dominio, permitió establecer una arquitectura modular y escalable, alineada con los objetivos del proyecto.

Fase 5. Realización de pruebas para asegurar el funcionamiento de la aplicación

Esta fase tuvo un mayor enfoque en el backend, ya que aquí es donde la lógica y la gestión de datos ocurre. Las pruebas del backend tuvieron su propia clasificación para fácilmente poder filtrar entre las pruebas correspondientes. El formato de las pruebas automatizadas del backend fue:

{tipo del test}: {modulo} - {método (http o función)}:{/ruta} - {acción}

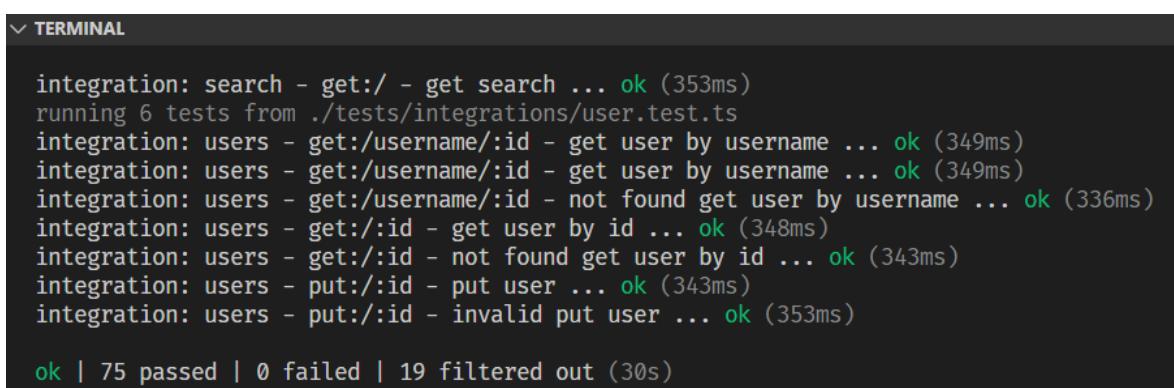
Ejemplos de pruebas fueron:

integration: auth - post:/login – valid

unit: typeChecker - verifyTypes - verify that password is incorrect

Pruebas de integración

En el desarrollo del proyecto, las pruebas de integración fueron esenciales para validar la interacción entre diferentes componentes del sistema. Estas pruebas buscaron detectar problemas relacionados con la comunicación entre módulos y aseguraron el cumplimiento de los flujos de trabajo completos (Figura 13).



```

▼ TERMINAL

integration: search - get:/ - get search ... ok (353ms)
running 6 tests from ./tests/integrations/user.test.ts
integration: users - get:/username/:id - get user by username ... ok (349ms)
integration: users - get:/username/:id - get user by username ... ok (349ms)
integration: users - get:/username/:id - not found get user by username ... ok (336ms)
integration: users - get:/:id - get user by id ... ok (348ms)
integration: users - get:/:id - not found get user by id ... ok (343ms)
integration: users - put:/:id - put user ... ok (343ms)
integration: users - put:/:id - invalid put user ... ok (353ms)

ok | 75 passed | 0 failed | 19 filtered out (30s)

```

Figura 11. Log de pruebas de integración.

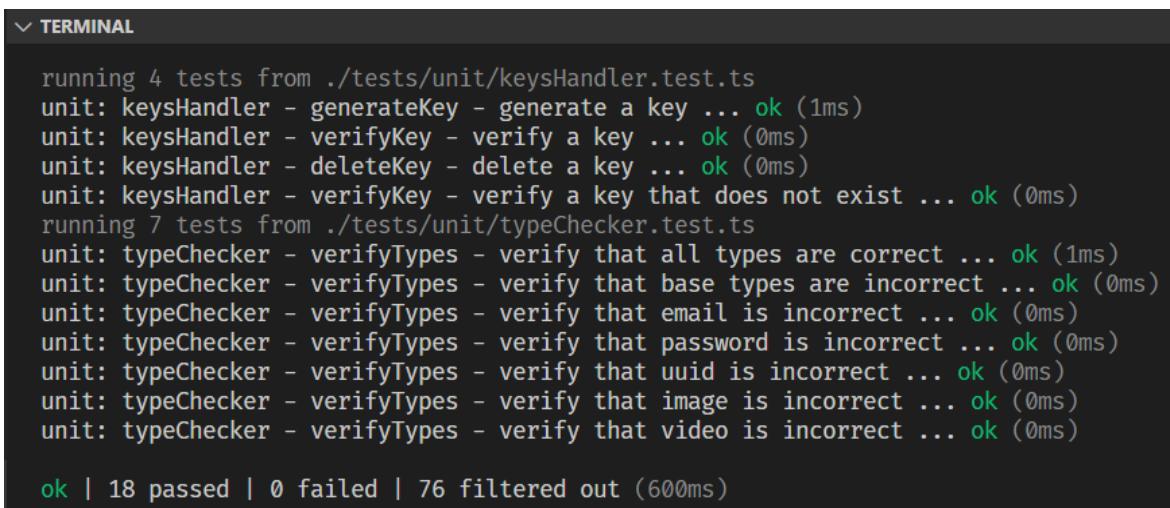
En el log presentado en la figura anterior, se evidenció que se ejecutaron pruebas bajo el prefijo “integration.”, lo cual dejó claro que están diseñadas específicamente para evaluar las interacciones entre diferentes endpoints de la API.

Flujo probado: Cada prueba ejecutó un flujo que involucró la interacción de varias capas del backend, como:

- Rutas que reciben solicitudes HTTP.
- Validación de datos entrantes.
- Comunicación con servicios internos y bases de datos.

Pruebas de Unidad

En el desarrollo del proyecto, las pruebas unitarias constituyeron una parte esencial para garantizar que cada componente del sistema funcionara de manera independiente y según lo esperado. Estas pruebas se centraron en validar el comportamiento de funciones o módulos individuales, asegurando que la lógica interna de cada unidad fuera robusta y cumpla con los requisitos especificados.



```

TERMINAL

running 4 tests from ./tests/unit/keysHandler.test.ts
unit: keysHandler - generateKey - generate a key ... ok (1ms)
unit: keysHandler - verifyKey - verify a key ... ok (0ms)
unit: keysHandler - deleteKey - delete a key ... ok (0ms)
unit: keysHandler - verifyKey - verify a key that does not exist ... ok (0ms)
running 7 tests from ./tests/unit/typeChecker.test.ts
unit: typeChecker - verifyTypes - verify that all types are correct ... ok (1ms)
unit: typeChecker - verifyTypes - verify that base types are incorrect ... ok (0ms)
unit: typeChecker - verifyTypes - verify that email is incorrect ... ok (0ms)
unit: typeChecker - verifyTypes - verify that password is incorrect ... ok (0ms)
unit: typeChecker - verifyTypes - verify that uuid is incorrect ... ok (0ms)
unit: typeChecker - verifyTypes - verify that image is incorrect ... ok (0ms)
unit: typeChecker - verifyTypes - verify that video is incorrect ... ok (0ms)

ok | 18 passed | 0 failed | 76 filtered out (600ms)

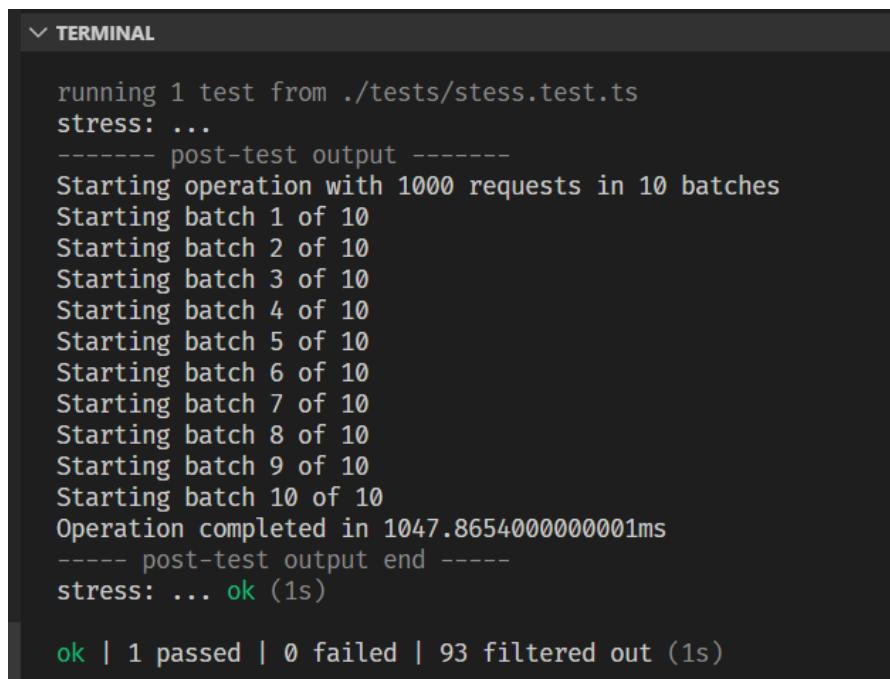
```

Figura 12. Log de pruebas de unidad.

El log proporcionado en la Figura 14 mostró la ejecución de un total de 18 pruebas unitarias, organizadas en diferentes archivos de prueba, cada uno enfocado en un módulo o función específica del backend.

En la fase de pruebas de estrés, el objetivo principal fue evaluar el rendimiento del sistema bajo condiciones de carga elevada. Estas pruebas fueron fundamentales para identificar cuellos de botella, límites de capacidad y puntos de fallo potenciales. En este caso, se ejecutó una prueba que simuló múltiples solicitudes concurrentes para realizar un login sencillo, evaluando la estabilidad y respuesta del sistema frente a una carga considerable.

El log mostró la ejecución de una prueba de estrés específica, identificada por el prefijo “stress:”. La prueba realizó un total de 1000 solicitudes, divididas en 10 lotes, lo que permitió observar el comportamiento del sistema al procesar cargas significativas de manera simultánea.



```

running 1 test from ./tests/stess.test.ts
stress: ...
----- post-test output -----
Starting operation with 1000 requests in 10 batches
Starting batch 1 of 10
Starting batch 2 of 10
Starting batch 3 of 10
Starting batch 4 of 10
Starting batch 5 of 10
Starting batch 6 of 10
Starting batch 7 of 10
Starting batch 8 of 10
Starting batch 9 of 10
Starting batch 10 of 10
Operation completed in 1047.8654000000001ms
----- post-test output end -----
stress: ... ok (1s)

ok | 1 passed | 0 failed | 93 filtered out (1s)
  
```

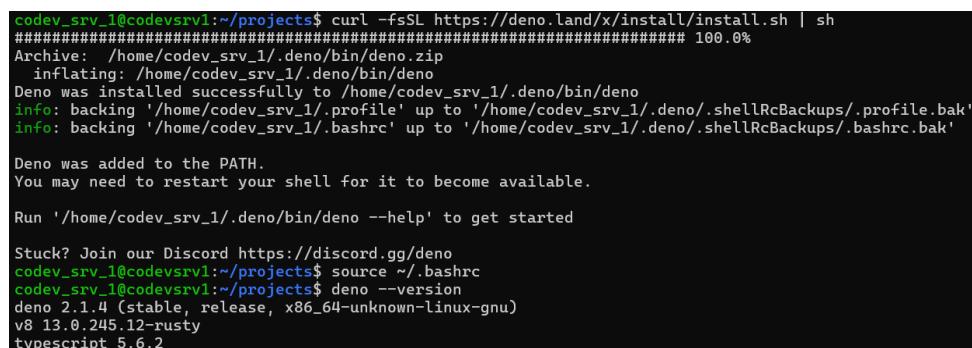
Figura 13. Log de pruebas de estrés.

El sistema completó las 1000 solicitudes en 1047.0865 ms (poco más de un segundo), lo cual reflejó un desempeño eficiente bajo esta carga específica.

Fase 6. Despliegue de prototipo

El despliegue del prototipo se llevó a cabo en un servidor de producción modelo HP ProLiant DL360p Gen8, equipado con dos procesadores Intel® Xeon® E5-2670 de ocho núcleos cada uno, soportando hasta 32 hilos en total. Este servidor contó con una memoria RAM de 94 GiB, de los cuales 6,2 GiB están en uso y 87 GiB están disponibles, lo que garantizó un rendimiento adecuado para la ejecución del sistema. En cuanto al almacenamiento, contuvo un disco principal de 98 GiB, con 15 GiB utilizados y 78 GiB disponibles. El servidor operó bajo el sistema Ubuntu Server, una plataforma segura, estable y optimizada para aplicaciones de producción. Con esta infraestructura, se aseguró la capacidad necesaria para soportar de manera eficiente el sistema implementado, así como las pruebas de rendimiento y estrés realizadas.

El proceso comenzó con la actualización del servidor, asegurando que todos los paquetes esenciales estuvieran al día, lo cual fue crucial para evitar vulnerabilidades y garantizar la estabilidad del entorno. A continuación, se instaló el software necesario, como el descomprimidor Unzip y el entorno de ejecución Deno, que fue fundamental para la ejecución del backend, así como los paquetes adicionales como PostgreSQL y Git, los cuales se validaron para confirmar su correcta instalación (Figura 16).



```

codev_srv_1@codevsrv1:~/projects$ curl -fsSL https://deno.land/x/install/install.sh | sh
#####
Archive: /home/codev_srv_1/.deno/bin/deno.zip
  inflating: /home/codev_srv_1/.deno/bin/deno
Deno was installed successfully to /home/codev_srv_1/.deno/bin/deno
info: backing '/home/codev_srv_1/.profile' up to '/home/codev_srv_1/.deno/.shellRcBackups/.profile.bak'
info: backing '/home/codev_srv_1/.bashrc' up to '/home/codev_srv_1/.deno/.shellRcBackups/.bashrc.bak'

Deno was added to the PATH.
You may need to restart your shell for it to become available.

Run '/home/codev_srv_1/.deno/bin/deno --help' to get started

Stuck? Join our Discord https://discord.gg/deno
codev_srv_1@codevsrv1:~/projects$ source ~/.bashrc
codev_srv_1@codevsrv1:~/projects$ deno --version
deno 2.1.4 (stable, release, x86_64-unknown-linux-gnu)
v8 13.0.245.12-rusty
typescript 5.6.2
  
```

Figura 14. Instalación de Deno desde la terminal.

La base de datos fue configurada en PostgreSQL, creando una nueva base de datos que sirve de soporte para el sistema. Con el entorno preparado, se clonó el repositorio del proyecto desde GitHub, y posteriormente se instalaron las dependencias necesarias para que la aplicación pudiera ejecutarse en el servidor de producción. La estructura de la base de datos se migró, asegurando que todos los cambios estructurales estuvieran alineados con los requerimientos del sistema.

Para asegurar un entorno seguro y eficiente, se configuró un archivo de variables de entorno donde se incluyeron las credenciales y configuraciones necesarias para la operación del backend en el servidor. El siguiente paso fue la creación de un servicio con systemd que gestionó el backend, permitiendo que la aplicación se ejecutara de manera continua y eficiente incluso después de reinicios del servidor. Después de habilitar y verificar el servicio, se configuraron los puertos necesarios para permitir el acceso a la aplicación desde redes externas.

Una vez configurado el entorno de producción, se realizaron varias pruebas para verificar su correcto funcionamiento. Estas pruebas incluyeron simulaciones en la misma red, pruebas desde una IP externa y una comparación de rendimiento entre el entorno local y el servidor desplegado. En cuanto a las pruebas de estrés, se compararon los tiempos de respuesta entre ambos entornos, demostrando una mejora significativa en la eficiencia del servidor desplegado. El resultado de la prueba realizada en el entorno local, mostró un tiempo de operación de 1037.0501 ms; mientras que, en el servidor desplegado, el tiempo fue considerablemente menor, con un resultado de 818.027995 ms, lo que reflejó una mejora del 21% en el tiempo de respuesta. Esto confirmó que el servidor estaba preparado para manejar la carga esperada de tráfico de manera eficiente (Figura 17).

```
● deno-app.service - Deno Backend App
   Loaded: loaded (/etc/systemd/system/deno-app.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-01-02 21:58:29 UTC; 1h 16min ago
     Main PID: 30921 (deno)
        Tasks: 47 (Limit: 115834)
       Memory: 1.0G (peak: 1.0G)
          CPU: 59.165s
         CGroup: /system.slice/deno-app.service
                 └─30921 /home/codev_srv_1/.deno/bin/deno run -A --unstable-worker-options /home/codev_srv_1/projects/ecosp>

ene 02 21:58:30 codevsrv1 deno[30921]: Upload file server running on http://localhost:4002/
ene 02 21:58:30 codevsrv1 deno[30921]: Main server worker listening on port 8019
ene 02 21:58:30 codevsrv1 deno[30921]: Main server worker listening on port 8020
ene 02 21:58:30 codevsrv1 deno[30921]: Main server worker listening on port 8021
ene 02 21:58:30 codevsrv1 deno[30921]: Main server worker listening on port 8022
ene 02 21:58:30 codevsrv1 deno[30921]: Main server worker listening on port 8023
ene 02 21:58:30 codevsrv1 deno[30921]: Socket server listening on localhost:4003
ene 02 21:58:30 codevsrv1 deno[30921]: Main server worker listening on port 8024
ene 02 21:58:30 codevsrv1 deno[30921]: Main server worker listening on port 8026
ene 02 21:58:30 codevsrv1 deno[30921]: Main server worker listening on port 8025
lines 1-20/20 (END)
```

Figura 15. Instalación de Deno desde la terminal.

En cuanto al despliegue de la aplicación móvil, el proceso comenzó con la configuración de un entorno en el Ionic Dashboard (Figura 18), donde se registró la aplicación y se vincularon las variables necesarias para su funcionamiento. Posteriormente, se configuró el entorno de compilación para generar el archivo APK, seleccionando la plataforma Android y el build stack adecuado. Una vez configurado, el proceso de generación del APK se realizó en tiempo real, lo que permitió obtener el archivo final que podría ser instalado en dispositivos o subido a la tienda de aplicaciones.

En resumen, el despliegue del prototipo fue un proceso meticuloso que involucró la actualización y configuración del servidor de producción, la validación y creación de la base de datos, la instalación de dependencias, y la ejecución de pruebas que validaron la estabilidad y optimización del sistema en el entorno de producción. Además, la generación del archivo APK permitió que la aplicación móvil estuviera lista para ser distribuida.

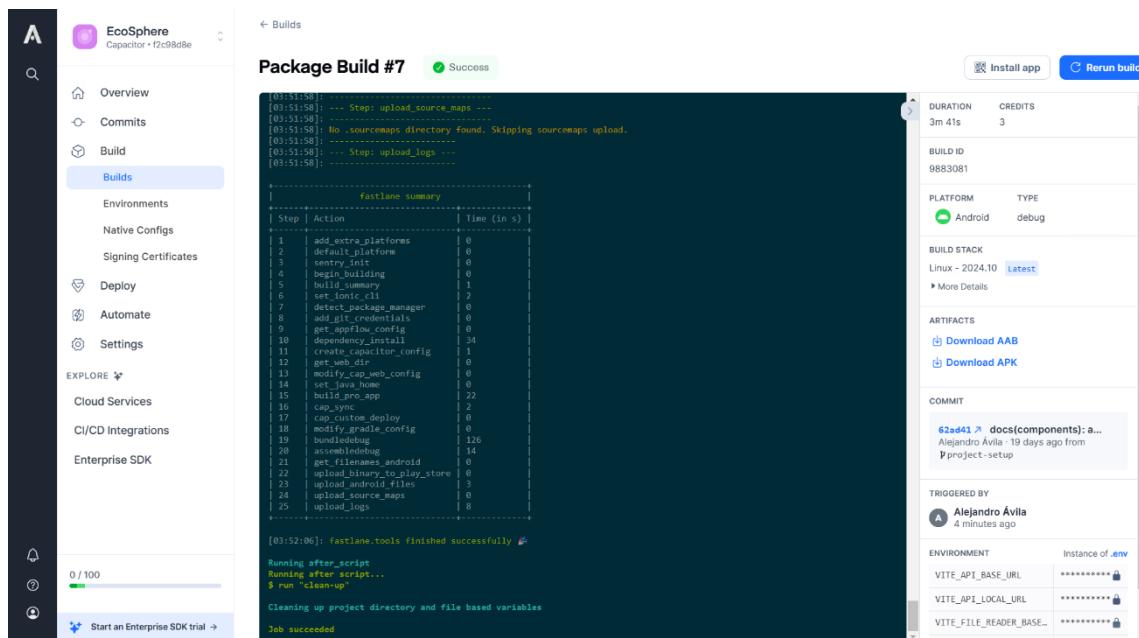


Figura 16. Generación de APK a través de Ionic Dashboard.

Conclusiones

El desarrollo de la red social orientada a fomentar la conciencia ambiental representa una propuesta innovadora que combina tecnología y sostenibilidad para generar impacto social positivo. A través de un diseño intuitivo, la aplicación permitió a los usuarios calcular su huella de carbono y visualizar cómo sus hábitos afectan al medio ambiente, promoviendo un cambio hacia prácticas más responsables. Además, la plataforma fomentó la interacción entre personas con intereses comunes en la sostenibilidad, fortaleciendo una comunidad comprometida con el cambio positivo.

Desde el aspecto técnico, se adoptaron herramientas modernas para garantizar una arquitectura robusta, escalable y eficiente. La integración de un frontend ágil, un backend bien estructurado y una base de datos sólida, complementada con pruebas exhaustivas, aseguró la calidad y la estabilidad del sistema. Este enfoque no solo facilita la experiencia del usuario, sino que también establece una base sólida para futuras mejoras y expansiones.

Para maximizar el potencial de la aplicación, se recomiendan varias actualizaciones orientadas a enriquecer la experiencia del usuario y ampliar su funcionalidad. Entre estas, destaca la implementación de un sistema de logros y badges que motive la participación activa, la incorporación de noticias ambientales y consejos prácticos, y el desarrollo de una herramienta para medir la reducción de la huella de carbono basada en actividades realizadas. Asimismo, una sección que recopile visualmente las actividades sostenibles de los usuarios a lo largo del tiempo podría fomentar un compromiso continuo, mientras que un espacio dedicado a la promoción de productos ecológicos conectará a los usuarios con marcas responsables.

Estas recomendaciones buscan fortalecer la conexión entre tecnología y sostenibilidad, garantizando que la aplicación evolucione para satisfacer las necesidades de sus usuarios y ampliar su impacto positivo. Con este enfoque, la red social no solo facilita un cambio individual, sino que contribuye a construir una sociedad más consciente y comprometida con el cuidado del planeta.

Referencias bibliográficas

- [1] Cambridge University Press. (n.d.). Mobile application. Cambridge Dictionary. [En línea]. Disponible en: <https://dictionary.cambridge.org/us/dictionary/english/mobile-application>

- [2] H. Hütt Herrera, “Las redes sociales: una nueva herramienta de difusión”. Reflexiones, Vol. 91, No. 2, pp. 121–128. 2012. [En línea]. Disponible en: <https://www.redalyc.org/pdf/729/72923962008.pdf>
- [3] J. Hurtado de Barrera, Metodología de la investigación holística, Tercera edición: Fundación Sypal, Caracas, Venezuela. 2000. [En línea]. Disponible en: <https://ayudacontextos.wordpress.com/wp-content/uploads/2018/04/jacqueline-hurtado-de-barrera-metodologia-de-investigacion-holistica.pdf>
- [4] F. G. Arias, El Proyecto de investigación, Sexta edición: Editorial Episteme, Caracas, Venezuela. 2012. [En línea]. Disponible en: <https://abacoenred.org/wp-content/uploads/2019/02/El-proyecto-de-investigaci%C3%B3n-F.G.-Arias-2012-pdf-1.pdf>
- [5] Ministerio del Medio Ambiente (MMA) de Chile. (n.d.). Huella de carbono. MMA. [En línea]. Disponible en: <https://mma.gob.cl/cambio-climatico/cc-02-7-huella-de-carbono/>
- [6] ClimateTrade. (n.d.). ¿Cómo calcular y compensar su propia huella de carbono? ClimateTrade. [En línea]. Disponible en: <https://climatetrade.com/es/como-calcular-y-compensar-su-propia-huella-de-carbono/>