

Pontificia Universidad Católica de Chile

Escuela de Ingeniería

Departamento de Ciencia de la Computación

|  |  |
| --- | --- |
| **Nombre del proyecto:** | **SCIENCENIA V2** |
| **Organización** | **CENIA** |
| **Fecha:** | **05/12/2024** |
| **Versión:** | **0.2** |
|  |  |
|  |  |

Documento Manual de Instalación

|  |  |
| --- | --- |
| **Nombre del proyecto:** | **…** |
| **Organización postulante:** | **…** |
| **Contraparte oficial del proyecto:** | **…** |
| **Dirección:** | **…** |
| **Teléfono:** | **…** |
| **Email:** | **…** |

Versión: 2010.v1

Historia del Documento

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Autor(es) | Razón del Cambio |
| 0.1 | 29/11/2024 | María Pía Vega | Pre- Entrega ciclo 3 |
| 0.2 | 05/12/2024 | Pablo Bustis | Entrega ciclo 3 |

Equipo de Desarrollo

|  |  |  |
| --- | --- | --- |
| Nombres y Apellidos | Rol | Contacto |
| María Pía Vega | Administradora del Proyecto | [mariapia.vega.f](mailto:jcampanella@puc.cl)@uc.cl  (+56 9) 9404 1471 |
| Ignacio Muñoz | Analista / Backend | ignaciomunoz@uc.cl  (+56 9) 5681 8618 |
| Sofía Larraín | Implementador / Backend | sofialarraínv@uc.cl  (+56 9) 8729 6649 |
| Felipe Contreras | Tester / Backend | ficontreras1@uc.cl  (+56 9) 7775 9418 |
| Pablo Bustos | Implementador / Backend | pablo.bustos@uc.cl  (+56 9) 5659 2795 |
| Alexandra San Martín | Analista / Frontend | apsanmartin@uc.cl  (+56 9) 9074 4671 |
| Trinidad Bravo | Implementador / Frontend | tsbravo@uc.cl  (+56 9) 5169 0895 |
| Cristóbal Martínez | Implementador / Frontend | crisms@uc.cl  (+56 9) 5728 8303 |
| Franco Anfossi | Analista / Mobile | franco.anfossi@uc.cl  (+56 9) 4216 2519 |
| Trinidad Schwarzenberg | Implementador / Mobile | tschwarzenberg@uc.cl  (+56 9) 9122 9252 |
| Martin Armstrong | Tester / Mobile | marmstrongleon@gmail.com  (+56 9) 7574 5584 |

Contraparte del Proyecto

|  |  |  |
| --- | --- | --- |
| Nombres y Apellidos | Rol | Contacto |
| ​Sebastián Martínez Figueroa | ​Coordinador de Investigación  ​y Gestión | ​Sebastian.martinez@cenia.cl  ​+56994040745 |
| ​Juan Ignacio Stark | ​Project Leader | ​Juanignacio.stark@cenia.cl  ​+56985004798 |
| ​Cristian Buc | ​Jefe Investigación Cenia | ​Cristian.buc@cenia.cl  ​+56933555811 |

Tabla de Contenidos

[Historia del Documento i](#_Toc1205557371)

[Equipo de Desarrollo ii](#_Toc181750760)

[Contraparte del Proyecto ii](#_Toc124690855)

[1. Descripción General 2](#_Toc507542464)

[1.1 Propósito del Sistema 2](#_Toc210189075)

[1.2 Alcance del Proyecto 3](#_Toc766049766)

[1.3 Ambiente Operacional de la Solución 4](#_Toc606739575)

[1.4 Definiciones, Acrónimos y Abreviaturas 5](#_Toc483117918)

[1.5 Referencias 5](#_Toc331899849)

[1.6 Estructura del Documento 5](#_Toc255254035)

[1.6.1 Sección 2: Instalación Backend 5](#_Toc1094975631)

[1.6.2 Sección 3: Instalación Frontend 5](#_Toc280728995)

[1.6.3 Sección 4: Instalación Mobile 5](#_Toc718960648)

[2. Instalación Backend 5](#_Toc1333880254)

[2.1 Setup 6](#_Toc272948071)

[2.1.1 Docker Compose 6](#_Toc845854034)

[2.2 Development 6](#_Toc1343424441)

[2.2.1 Linting 6](#_Toc2141788949)

[2.2.2 Type checking 7](#_Toc2085122335)

[2.2.3 Hot reloading 7](#_Toc2099557319)

[2.2.4 Debugging 7](#_Toc1737011177)

[2.3 Deployment 7](#_Toc381964636)

[2.4 Mailer 8](#_Toc331962687)

[3. Instalación Frontend 8](#_Toc1070094074)

[4. Instalación Mobile 9](#_Toc1949695798)

[4.1 Setup 9](#_Toc890355147)

[4.2 Configuring a developer Auth0 App 10](#_Toc1098292586)

# Descripción General

Se comenzará dando una descripción breve del contexto, objetivos y alcance del sistema a desarrollar, así como la documentación relativa al mismo. El Centro Nacional de Inteligencia Artificial (CENIA) es una corporación público-privada que está financiada por la Agencia Nacional de Investigación y Desarrollo (ANID) desde noviembre del año 2021. CENIA tiene como objetivo poner a la inteligencia artificial al servicio de las personas a través de tres líneas de acción: investigación científica de vanguardia, transferencia tecnológica y vinculación con la sociedad mediante proyectos sobre Inteligencia Artificial (IA). Además, el centro cuenta con cuatro universidades fundadoras y con más de 50 investigadoras e investigadores asociados. CENIA representa una oportunidad en el desarrollo de la IA a nivel nacional, ya que no solo es un espacio para estudiar diversos contenidos de la IA, sino que también es una clave para la expansión de todo el ecosistema nacional de investigación y desarrollo en este campo.

El proyecto Sciencenia Hub surge como una respuesta a la necesidad crítica de optimizar la gestión y el seguimiento de la producción científica dentro de CENIA. El objetivo del proyecto Sciencenia en su segunda versión es desarrollar una plataforma avanzada para la gestión y seguimiento de la producción científica dentro de CENIA. La necesidad de optimizar la gestión de la información científica, superar la fragmentación de los sistemas actuales y proporcionar análisis profundos para la toma de decisiones estratégicas impulsa este desarrollo. Sciencenia Hub centralizará la información dispersa en múltiples fuentes, ofreciendo herramientas analíticas avanzadas para mejorar la planificación y asignación de recursos, promoviendo así la eficiencia y efectividad en la investigación científica.

El principal objetivo del sistema es desarrollar una plataforma de análisis científico avanzado que

facilite la síntesis y el análisis de la producción científica, ofreciendo *insights* clave para la toma de decisiones estratégicas. El sistema también busca generar informes automatizados, asegurar la seguridad y privacidad de los datos, y promover una cultura de innovación dentro de la organización.

En relación con el alcance del sistema, el sistema abarcará desde el diseño inicial hasta el lanzamiento de aplicaciones web y móviles, asegurando calidad y funcionalidad mediante pruebas exhaustivas. Incluirá la implementación de protocolos de seguridad rigurosos para proteger la información y cumplir con las normativas de privacidad de datos. Además, se contemplará el monitoreo y la mejora continua del rendimiento de la plataforma.

La documentación relativa al proyecto incluye los detalles acerca de las restricciones, exigencias y dificultades relevantes del proyecto, además de las soluciones propuestas para cada una de ellas. Se incluyen los protocolos de seguridad implementados y las métricas para la evaluación de la aplicación. Toda esta información es obtenida a partir de las necesidades del cliente a satisfacer, en donde se asegura que se cumplan las expectativas que este tenga.

## Propósito del Sistema

La aplicación proporcionará una plataforma que se encargará de la recopilación y análisis de documentos científicos de los investigadores de CENIA. El enfoque en esta iteración es centrado en el estudiante, poniendo especial énfasis en mejorar el flujo de solicitudes de incentivos y fondos para tener un seguimiento detallado del estado de la solicitud. Adicionalmente, contará con la funcionalidad de tener un perfil, donde podrán editar sus datos personales y actualizar datos de su tesis. También se espera implementar un sistema de notificaciones, donde el estudiante puede recibir alertas sobre su tesis, resolución de solicitudes de becas y recursos y eventos, esto se espera lograr tanto en la aplicación web como móvil.

Por otro lado, el observatorio científico estará manejado por los ejecutivos que cargarán grandes volúmenes de producción científica mediante un archivo .csv y así evitar la introducción manual. Además, se espera lograr un filtrado avanzado con palabras clave, fechas, áreas de investigación y autores.

En lo que respecta al manejo de usuarios, el ejecutivo tiene la responsabilidad de gestionar a los usuarios, incluyendo la creación y eliminación de cuentas. Este puede crear cuentas de usuarios mediante la carga de un archivo .csv con la información correspondiente, lo que permite generar las cuentas de manera automática con una contraseña provisional. Posteriormente, los estudiantes pueden actualizar su contraseña por una propia y modificar sus datos personales, a excepción de su rol de "estudiante", que no es editable. Finalmente, la aplicación debe ser intuitiva y fácil de usar para los usuarios generando una mayor adaptación a la aplicación.

## Alcance del Proyecto

El proyecto consiste en la continuación de la creación de una página web y también de la mejora y completo funcionamiento de la aplicación móvil. Los usuarios de Sciencenia van a poder acceder a las producciones científicas de manera sencilla y ordenada, además de poder generar un análisis de estas producciones. En relación con los usuarios, existen tres tipos de usuarios: estudiantes, ejecutivos y usuario público (incluye al investigador).

Los estudiantes podrán tener un perfil, el cual podrán editar para mantenerlo actualizado, tendrán su tesis asociada en la página y podrán editar datos no sensibles sobre esta. También podrán realizar solicitudes de becas y recursos para conferencias y/o viajes mediante el envío de un formulario que contiene toda la información necesaria para la postulación. Finalmente, serán capaces de recibir notificaciones en tiempo real de la resolución de su solicitud y también sobre eventos de CENIA.

Por otro lado, el ejecutivo podrá recibir las solicitudes de los estudiantes y analizar la información enviada para aceptar o rechazar la solicitud. Puede subir y actualizar los datos de la tesis del estudiante. Está encargado de la carga masiva de datos de la aplicación, los que son de investigaciones y de usuarios, donde crean los perfiles del estudiante y también generan la información de los investigadores. Además, puede publicar anuncios de eventos de CENIA para que lo puedan ver los estudiantes y mandar notificaciones acerca de estos mismos.

Cada uno de los usuarios mencionados anteriormente pasa por un proceso de autenticación lo que asegura la información personal.

El tipo de usuario público tiene acceso al observatorio que es donde están las investigaciones de CENIA y también tiene acceso a la sección de Investigadores, donde puede ver la información de los investigadores asociados. El perfil de “investigador” del proyecto inicial, por decisión del cliente pasa a ser un usuario público, es decir, no tiene ni perfil ni actualizará información de sus investigaciones.

## Ambiente Operacional de la Solución

El sistema funcionará en un entorno de nube proporcionado por AWS y Cloudflare con las siguientes características:

El ambiente está configurado para trabajar en la plataforma de AWS, utilizando los servicios VPC (Virtual Private Cloud), RDS (Amazon Relational Database Service) y ECS (Elastic Container Service). La base de datos está alojada en un servidor PostgreSQL administrado dentro de la VPC.

**Infraestructura de la Aplicación**

* **VPC:** La aplicación estará alojada en una VPC (Amazon Virtual Private Cloud) personalizada, que incluye soporte para una instancia bastión para acceso seguro.
* **Base de Datos:** La base de datos principal es un servidor PostgreSQL en RDS (Amazon Relational Database Service) con alta disponibilidad. El acceso a la base de datos se realiza a través de conexiones TLS, asegurando la privacidad de los datos en tránsito.
* **Cluster:** La aplicación web utiliza un clúster de ECS (Amazon Elastic Container Service) para ejecutar los contenedores de la API, que se gestionan dinámicamente dentro de la infraestructura de AWS.
* **Servicio Web:** La API está expuesta en un puerto seguro (443) para conexiones HTTPS y redirige las solicitudes a la aplicación alojada en el puerto 3000.
* **Cloudflare:** En el entorno de producción, se utiliza un dominio personalizado que está configurado con Cloudflare para la gestión del DNS.

Para el buen funcionamiento del sistema, los usuarios deben tener acceso a un navegador moderno que soporte HTTPS. La comunicación entre los componentes se realiza a través de redes virtuales seguras en AWS, utilizando protocolos de seguridad avanzados como TLS y SSM para la creación de túneles entre la aplicación y la base de datos.

**SST (Serverless Stack Toolkit)**

La infraestructura del sistema está definida como código utilizando SST (Serverless Stack Toolkit), aplicando el concepto de IaC (Infrastructure as Code). Esto permite automatizar y gestionar fácilmente los recursos en AWS, como VPC, RDS y ECS, a través de código, garantizando consistencia, facilidad de despliegue, y soporte para integración continua. IaC facilita el escalado automático y la administración eficiente del entorno en la nube.

## Definiciones, Acrónimos y Abreviaturas

ANID: Agencia Nacional de Investigación y Desarrollo

AWS: Amazon Web Services

CENIA: Centro Nacional de Inteligencia Artificial

IA: Inteligencia Artificial

## Referencias

En esta sección se enumeran la documentación y bibliografía utilizada como apoyo para construir este documento:

1. ESA Software Engineering Standards, PSS-05-0, URD (User Requirement Document) and SRD (Software Requirement Document). ESA Board for Software Standardization and Control (BSSC) - European Space Agency, 2006, URL: <https://goo.gl/RHfZ06>
2. User Stories Applied: For agile software development. Cohn, M., Addison-Wesley Professional, 2004.
3. Agile Estimating and Planning. Cohn, M., Prentice Hall, 2005.
4. Scrum and XP from the Trenches: How we do Scrum. Kniberg, H., InfoQ Enterprise Software Development Series, 2006, URL: <https://goo.gl/bbhcFk>
5. Implementing Lean Software Development: From Concept to Cash. Poppendieck, M., Poppendieck, T., Pearson Education, 2007.
6. Perfil Proyecto plantilla 2024 - Sebastián Martínez. URL: [P24 - [SCIENCENIA V2] IIC2154.PerfilProyecto.Plantilla.2024 - Sebastián Martínez](https://uccl0-my.sharepoint.com/:b:/r/personal/aneyem_uc_cl/Documents/OneDrive.2023/DCC-UC/Cursos/IIC2154/2024/2024-2/IIC2154.2024-2.CoordinacionGrupos/Seccion-4/IIC2154.2024-2.S4.Grupo1/Repositorio/Proyecto.DocumentoPerfil/P24%20-%20%5BSCIENCENIA%20V2%5D%20IIC2154.PerfilProyecto.Plantilla.2024%20-%20Sebasti%C3%A1n%20Mart%C3%ADnez.pdf?csf=1&web=1&e=pzmQhZ)

## Estructura del Documento

### Sección 2: Instalación Backend

**Público Objetivo:** Desarrolladores

**Área:** Explicación de cómo instalar backend.

### Sección 3: Instalación Frontend

**Público Objetivo:** Desarrolladores, analistas, clientes

**Área:** Establecen, desde el punto de vista de un usuario, qué funcionalidades se implementarán, el alcance de la implementación, y sus plazos.

### Sección 4: Instalación Mobile

**Público Objetivo:** Desarrolladores

**Área:** Explicación de cómo instalar mobile.

# Instalación Backend

## Setup

Este repositorio usa Bun como el entorno de ejecución de JavaScript, gestor de paquetes y empaquetador. Para instalarlo, ejecuta el siguiente comando:

* Para Linux y macOS:

curl -fsSL https://bun.sh/install | bash

* Para Windows:

powershell -c "irm bun.sh/install.ps1 | iex"

Después de instalar Bun, ejecuta el siguiente comando para instalar las dependencias del proyecto:

bun install

A continuación, crea un archivo .env en la raíz del proyecto y copia el contenido del archivo .env.template. Los valores predeterminados deberían funcionar para el desarrollo local si usas *Docker Compose* para configurar la base de datos.

Se recomienda instalar Bun en tu entorno de desarrollo local, incluso si usas *Docker Compose*, ya que te permitirá ejecutar scripts fácilmente.

### Docker Compose

Para configurar un entorno de desarrollo completo, puedes usar Docker Compose. Primero, crea una red:

docker network create app-network

Luego, puedes iniciar la base de datos y la API:

docker compose up

La API estará disponible en <http://localhost:3000>.

## Development

Para el *development*, se recomienda habilitar la función de Watch de Docker Compose, que sincroniza los cambios en el contenedor de la API, donde Bun reiniciará automáticamente el servidor. Para usar esta función, solo agrega la bandera --watch al comando up:

docker compose up --watch

### Linting

El proyecto usa ESLint para el análisis de código. Para ejecutar el linter, usa el siguiente comando:

bun lint

### Type checking

El proyecto usa TypeScript para la verificación de tipos. Si usas VSCode, esto debería funcionar automáticamente. Si deseas ejecutar el verificador de tipos manualmente, usa el siguiente comando:

bun type-check

### Hot reloading

Usar el --watch permite a Bun reiniciar automáticamente el servidor cuando se realizan cambios, pero si prefieres usar la recarga en caliente de Bun para obtener actualizaciones casi instantáneas cada vez que actualizas el código, puedes sobrecargar el script de Bun que Compose ejecutará utilizando la variable de entorno BUN\_SCRIPT (el valor predeterminado es dev):

BUN\_SCRIPT=dev:hot docker compose up –watch

La única desventaja es que esto es menos útil para cambios que afectan el ciclo de inicio de la aplicación (por ejemplo, cambios en los esquemas de los modelos), ya que las actualizaciones no se reflejarán a menos que reinicies manualmente el servidor.

### Debugging

**Advertencia:** La depuración a través de VSCode mientras se ejecuta la aplicación en Docker Compose no es compatible debido a limitaciones en la extensión Bun para VSCode. Consulta esta [issue](https://github.com/oven-sh/bun/issues/7490).

Si deseas depurar la aplicación mientras usas Docker Compose, puedes usar el depurador basado en la web de Bun. Para hacerlo, solo inicia la aplicación y navega a <https://debug.bun.sh/#localhost:6499/api>. Puedes encontrar un tutorial sobre cómo usar el depurador [aquí.](https://bun.sh/docs/runtime/debugger#debuggers)

Ten en cuenta que los enlaces que aparecen al iniciar la aplicación no funcionarán, por lo que debes usar el enlace proporcionado en su lugar.

## Deployment

Este proyecto define su infraestructura como código usando SST. La especificación de esta infraestructura se encuentra en el archivo sst.config.ts.

Para desplegar el proyecto, necesitas configurar tus credenciales de AWS, así como establecer una variable de entorno CLOUDFLARE\_API\_TOKEN con el valor de tu token de API de Cloudflare. Este token debe tener al menos permisos para editar DNS y acceso de visualización a la configuración de la cuenta.

Puedes usar tu propia cuenta de AWS si lo deseas, aunque debes asegurarte de que el dominio especificado en sst.config.ts esté disponible en tu cuenta de Cloudflare. Si vas a redeplegar la aplicación en un nuevo dominio, solo cambia el valor al dominio o subdominio que esté bajo tu cuenta de Cloudflare.

Después de esto, puedes desplegar la aplicación (a producción) usando:

bun run deploy

Si prefieres desplegar a un entorno personal:

bun run deploy:personal

Y si deseas usar la infraestructura real para respaldar tu proceso de desarrollo, puedes usar:

bun run dev:sst

Esto desplegará temporalmente la base de datos de la aplicación, pero cargará una versión local de la API (y vinculará las credenciales automáticamente). Esto puede ser útil para depurar problemas que solo aparecen con la base de datos real de AWS RDS

## Mailer

Se implementó la funcionalidad de envío de correos electrónicos para notificar a los usuarios sobre eventos específicos que ocurren dentro de la plataforma. Para ello, se configuró el correo sciencenia@cenia.cl como remitente para todos los envíos y se utilizó una contraseña de aplicación para acceder a esta.

Durante el desarrollo y pruebas del proyecto, se implementó un filtro que restringe el envío de correos únicamente a los integrantes del grupo de desarrollo. Esto asegura que los correos no lleguen a los usuarios reales mientras la plataforma se encuentra en una etapa de pruebas. Por lo que una vez que se encuentre finalizado el proyecto, se debe de eliminar dicho filtro para que los correos sean entregados correctamente a los usuarios finales según los eventos configurados.

# Instalación Frontend

El proyecto funciona bajo Node.js y usando yarn para ejecutar. Para instalar ambos, ejecuta el siguiente comando:

Node en Linux:

curl -fsSL https://deb.nodesource.com/setup\_lts.x | sudo -E bash -  
sudo apt install -y nodejs

Yarn en Linux:

curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -  
sudo apt update  
sudo apt install yarn

Node en MacOS:

curl -o node.pkg <https://nodejs.org/dist/v18.19.0/node-v18.19.0.pkg>

Yarn en MacOs:  
curl -o- -L https://yarnpkg.com/install.sh | ba

Node en Windows:  
Invoke-WebRequest -Uri https://nodejs.org/dist/v18.19.0/node-v18.19.0-x64.msi -OutFile nodejs.msi  
Yarn en Windows:

npm install -g yarn

Una vez instalado ambos, se debe correr Yarn para instalar depedencias y Yarn dev para ejecutar y ver el proyecto en <http://localhost:5175/>

# Instalación Mobile

## Setup

Copiar repositorio

Cuando inicialices un nuevo entorno de desarrollo, debes hacer lo siguiente:

* Instalar [Node.js LTS](https://nodejs.org/en/download/package-manager/all) (Node 20.x en el momento de escribir este texto) y [Bun](https://bun.sh/docs/installation).
* Seguir la guía de [Set up your environment](https://docs.expo.dev/get-started/set-up-your-environment/) en la documentación de Expo.
* Deberás seleccionar " Development build ", ya que estamos utilizando CNG y no Expo Go.
* Después de seleccionar " Development build", deberás deshabilitar "Build with Expo Application Services (EAS)".
* Al seguir las instrucciones, sustituye cualquier comando npm por bun (recomendado).
* Ingresa al repositorio y ejecuta bun install para instalar las dependencias.
* Completa el archivo .env con las variables de entorno necesarias. Consulta .env.template como referencia.
* Por ahora, solo se necesita el client ID y el dominio de Auth0.
* Ejecuta bun start para iniciar el servidor de desarrollo.
* Si tienes algún problema y necesitas reiniciar la aplicación, se recomienda ejecutar bun run prebuild:clean para realizar nuevamente el build de desarrollo.

Opcionalmente, también puedes instalar [Reactotron](https://docs.infinite.red/reactotron/) como herramienta de depuración. Una vez abierto, deberías recargar la aplicación a través de Reactotron para conectarla con una aplicación en ejecución. Para rastrear cambios en el estado, puedes comenzar suscribiéndote a \*, lo cual te mostrará el estado raíz de Zustand.

## Configuring a developer Auth0 App

* Crea una nueva aplicación de Auth0, con el tipo de aplicación "Native".
* En la pestaña Settings (Configuración), copia el dominio y el client ID al archivo .env.
* En las secciones Allowed Callback URLs y Allowed Logout URLs, agrega lo siguiente:

com.cenia.scienceniahub.auth0://YOUR\_DOMAIN/ios/com.cenia.scienceniahub/callback, com.cenia.scienceniahub.auth0://YOUR\_DOMAIN/android/com.cenia.scienceniahub/callback

Reemplaza ambas instancias de YOUR\_DOMAIN con el dominio de tu Auth0.