



Universidad Austral de Chile



---

## Manual de lanzamiento de la aplicación del proyecto CEHUM Humedata

---

Líder del proyecto: Dr. Cristian Correa

Administrador del proyecto Humedata app: Matías Adaury

Revisor técnico: Matías Soto Lomboy

Fecha: 04/2025

## 0. Índice

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Requisitos previos.....</b>	<b>5</b>
<b>3. Configuración Inicial del Proyecto.....</b>	<b>6</b>
<b>4. Creación y Configuración de la Cuenta MongoDB Atlas.....</b>	<b>8</b>
<b>5. Configuración del Servicio Sentry.....</b>	<b>10</b>
<b>6. Creación de la Cuenta Clerk.....</b>	<b>12</b>
<b>7. Configuración de las Variables de Entorno.....</b>	<b>13</b>
<b>8. Setup del Frontend.....</b>	<b>14</b>
<b>9. Setup del Backend.....</b>	<b>16</b>
9.1. Conexión del backend a MongoDB en modo producción.....	19
<b>10. Ejecución en Modo Desarrollo.....</b>	<b>20</b>
<b>11. Requisitos previos de la aplicación móvil.....</b>	<b>21</b>
<b>12. Configuración inicial de la Aplicación Móvil.....</b>	<b>22</b>
12.1. Construir la aplicación para Android.....	22
<b>13. Ejecución de la Aplicación Móvil en Modo Desarrollo.....</b>	<b>23</b>

# 1. Introducción

El siguiente documento es un manual detallado de instalación y configuración del proyecto *CEHUM Humedata* que contiene la información necesaria para realizar su lanzamiento. El proyecto consta de tres aplicaciones principales:

1. Una aplicación web desarrollada con Next.js
2. Una API construida con Express
3. Una aplicación móvil desarrollada con Expo y React Native

El proyecto utiliza una arquitectura *mono-repositorio*, esto significa que las tres aplicaciones conviven en un mismo repositorio de código, donde cada una puede ser lanzada independientemente. Para gestionar este proyecto de manera eficiente se utilizó Turborepo, una herramienta creada por Vercel que facilita la coordinación, ejecución y construcción de proyectos dentro de un entorno *monorepo*.

Este manual incluye información previa necesaria para entender cada fase del lanzamiento del proyecto y una guía paso a paso a través del proceso de configuración. Se hablará desde los requisitos iniciales y el entorno de desarrollo hasta la puesta en marcha de cada componente del proyecto. El proceso de lanzamiento en este manual está basado en su mayoría en Windows, por lo que en Linux y MacOS, algunos pasos pueden variar.

La aplicación utiliza múltiples plataformas para distintos principios, estas son:

- Vercel para el lanzamiento del frontend.
- Amazon Web Services para el lanzamiento del backend.
- MongoDB como base de datos.
- Sentry para monitorear la aplicación en caso de errores.
- Clerk para manejar usuarios.

Esto se puede ver reflejado en el siguiente diagrama:

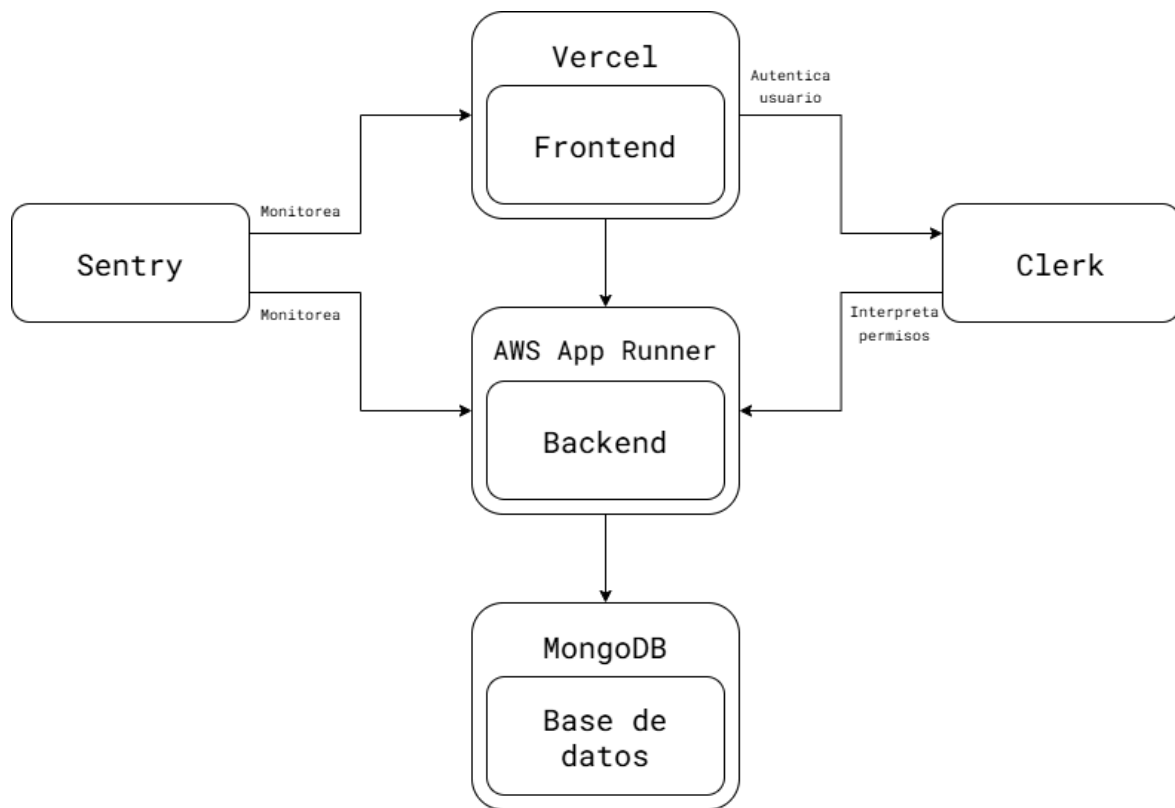


Figura 1: Diagrama de plataformas proyecto humedata

## 2. Requisitos previos

Antes de comenzar con la instalación y configuración del proyecto, necesita asegurarse de preparar su entorno de desarrollo. Esta sección incluye solo los requisitos previos para trabajar con la aplicación web, la aplicación móvil tiene su propia sección más adelante.

Necesita cumplir con los siguientes requisitos:

1. Instale la última versión estable de Node.js desde el sitio web [nodejs.org](https://nodejs.org).  
Para asegurarse de que node fue instalado correctamente, puede abrir una terminal e ingresar el comando: `node --version`
2. Instale pnpm globalmente desde la terminal utilizando el siguiente comando:  
(dependiendo del sistema operativo que esté usando, su terminal puede ser distinta. En windows puede ser CMD o Powershell y en linux o macOS puede ser bash)

```
npm install -g pnpm
```

Para comprobar que pnpm fue instalado correctamente, puede ingresar el siguiente comando en la terminal: `pnpm --version`

3. Cree una cuenta de [GitHub](https://github.com) si aún no tiene una.
4. Instale Git desde su [página oficial](https://git-scm.com) y desde una terminal configúralo con sus credenciales mediante los siguientes comandos:

```
git config --global user.name "Su nombre"  
git config --global user.email "su-email@ejemplo.com"
```

Para comprobar que sus credenciales son correctas puede escribir el comando:  
`git config --global --list`

5. Como recomendación, puede instalar [Visual Studio Code](https://code.visualstudio.com), un entorno de desarrollo integrado (IDE) que le permitirá una mejor gestión de los archivos dentro del proyecto y que también incluye su propio terminal. En esta guía está explicado cómo utilizar VSCode para realizar el lanzamiento de la aplicación.
- **RECOMENDACIÓN:** En el proceso de lanzamiento necesitará tomar nota de datos de varias plataformas y guardar varias credenciales, por lo que se recomienda crear un archivo Excel para organizar todos estos datos, de esta manera los mismos estarán disponibles en el caso de que los necesite.

### 3. Configuración Inicial del Proyecto

1. Descomprima el zip donde se encuentra empaquetado el proyecto y guárdelo en la carpeta que desee. Puede guardar el contenido de esta carpeta como una copia de seguridad, puesto que luego, el código fuente tendrá que ir a la carpeta que le asigne al repositorio.
2. Cree un nuevo repositorio en GitHub, para ello:
  - a. Vaya a la [página principal](#) e inicie sesión en su cuenta.
  - b. Haga clic en el botón “+” en la esquina superior derecha y seleccione “New Repository”.
  - c. Asigne un nombre al repositorio y haga clic en “Create Repository”  
Ejemplo de nombre: cehum-humedata
3. Suba el código fuente del proyecto a su nuevo repositorio GitHub, para ello tiene dos métodos.
  - a. Método por interfaz gráfica:
    - i. Descargue e instale [GitHub Desktop](#), la interfaz gráfica de GitHub donde usted puede manejar sus repositorios de una manera más intuitiva.
    - ii. Inicie sesión con su cuenta de GitHub y haga clic en el botón “Clone a repository from the internet”
    - iii. Seleccione el repositorio que creó en su proyecto y designe una carpeta para almacenar su repositorio local.  
NOTA: Tenga precaución con guardarlo en un directorio que tenga como nombre caracteres especiales, prefiera una ruta donde no haya carpetas con un tilde en el nombre, esto puede dar problemas más adelante. Ejemplo:  
C:\Users\<su user>\OneDrive\Desktop\Código
    - iv. Una vez clonado el repositorio, haga clic derecho en el botón “Current repository” en la esquina superior izquierda y seleccione la opción “Show in explorer” para abrir su repositorio local desde su explorador de archivos.
    - v. Puede descomprimir o hacer una copia del proyecto en esta carpeta. Esta es la versión del código fuente que lanzará al internet.
    - vi. En la esquina inferior izquierda asígnele un nombre a su primer commit y haga clic en el botón “Commit to main”.  
Ejemplo de nombre: Commit inicial del proyecto CEHUM Humedata
    - vii. En la barra superior, haga clic en el botón “Publish branch”.
  - b. Método por la terminal de su computadora:
    - i. Abra una terminal y navegue hasta el directorio del proyecto descomprimido.  
Esto lo puede lograr mediante el comando `cd <Directorio donde guardó su código descomprimido>`.  
Ejemplo: `cd C:\Users\<su-user>\OneDrive\Desktop\cehum-humedata-app`
    - ii. Inicialice un nuevo repositorio git con el comando:  
`git init`

- iii. Agregue todos los archivos al staging con el comando:  
`git add`
  - iv. Realice el primer commit con el comando:  
`git commit -m "Commit inicial del proyecto CEHUM Humedata"`
  - v. Añada el repositorio remoto de GitHub con el comando:  
`git remote add origin`  
`https://github.com/<su-usuario>/<nombre-del-repositorio>`
  - vi. Suba los cambios a GitHub:  
`git push -u origin main`
4. Una vez subido el proyecto al repositorio a GitHub necesita instalar las dependencias del mismo, para ello puede usar la terminal integrada de Visual Studio Code o abrir la raíz del proyecto dentro de la terminal de su sistema operativo. Para ello:
- a. Si tiene GitHub Desktop, haga clic derecho en el botón "Current repository" y seleccione la opción "Open in Visual Studio Code". En su defecto, puede abrir VSCode, hacer clic en "File" en la esquina superior izquierda y seleccionar "Open folder", donde puede seleccionar el directorio raíz del proyecto.
  - b. En la barra superior de VSCode, navegue a la sección "Terminal" y abra un nuevo terminal.
  - c. Asegúrese de encontrarse en el directorio raíz del proyecto y ejecute el comando:  
`pnpm install`
  - d. Si no cuenta con Visual Studio Code, puede hacer lo mismo desde su terminal de su sistema operativo, navegando hasta el directorio raíz del proyecto e ingresando el comando.
- El proceso de instalación de dependencias puede durar varios minutos dependiendo de la velocidad de su conexión a internet y el rendimiento de su computadora.

## 4. Creación y Configuración de la Cuenta MongoDB Atlas

MongoDB Atlas es un servicio de base de datos totalmente gestionado que cuenta múltiples sistemas para facilitarle la vida a sus usuarios. En este proyecto, los datos a almacenar son tanto la ubicación de los humedata y la información que estos recolectan como la información de los usuarios que utilizan la plataforma.

Si desea analizar el modelo de base de datos más a detalle, puede hacerlo entrando a los archivos `cehum.prisma` y `mongo.prisma` desde la siguiente ruta dentro del proyecto: `apps\backend\prisma`. Para inicializar la base de datos en MongoDB Atlas y conectarla al proyecto deberá seguir los siguientes pasos:

1. Cree una cuenta en [MongoDB Atlas](#), rellene el formulario de registro con su información personal y verifique su dirección de correo electrónico.
2. Una vez creada la cuenta y estando en el panel de control de Atlas, haga clic en el botón de proyectos y entre a “New Project”.
3. Asígnele un nombre al proyecto y complete el proceso de creación.  
Ejemplo de nombre: CEHUM\_Humedata.
4. A continuación, cree un nuevo cluster. Para ello:
  - a. En la pestaña de Overview, haga clic en “Create” bajo la sección de “Create a cluster”.
  - b. Seleccione la opción M0, es gratis y ofrece 512MB de almacenamiento gratuito.  
Usar esta opción cuando aún la aplicación no entre en el ámbito de producción.
  - c. Asígnele un nombre al cluster y elija el proveedor de nube y la región más cercana a su ubicación.  
Ejemplo de nombre: HumedataCluster
  - d. Haga clic en “Create Deployment” y espere a que se complete la creación.  
Este proceso puede tardar unos minutos.
5. Ahora debe crear un acceso a su base de datos:
  - a. En el panel lateral izquierdo, vaya a “Database Access” bajo la sección “Security”
  - b. Haga clic en “Add New Database User”
  - c. Elija un nombre y una contraseña fuerte. Guarde estos datos en un lugar seguro.
  - d. Haga clic en “Add built in role” y seleccione “Read and write to any database”.
  - e. Luego, en el panel izquierdo nuevamente, vaya a “Network access” bajo la sección “Security”.
  - f. Asegúrese de que su dirección IP aparezca en la lista, esto le servirá para que el backend se pueda conectar a la base de datos al lanzar la aplicación de manera local.  
Si desea permitir el acceso desde cualquier IP, haga clic en “Add IP address” e ingrese 0.0.0.0/0 en el campo de IP, de otra manera, el backend al ser lanzado en



AWS App Runner no podrá leer ni escribir en la base de datos. **PRECAUCIÓN**, no haga esto en el entorno de producción.

6. Regrese a la pestaña de “Overview” y haga clic en “Connect” para el cluster que creó anteriormente.
7. Seleccione “Drivers” bajo la categoría “Connect to your application”.
8. Tome nota de la cadena de conexión, puesto que la necesitará más adelante. Esta cadena tendrá la siguiente estructura:  
`mongodb+srv://<username>:<db_password>@cluster0.1n5ux.mongodb.net/<database-name>?retryWrites=true&w=majority&appName=Cluster0`  
Nota: Lo más probable es que MongoDB le entregue una cadena sin el campo <database-name>, si es que le llega a ocurrir un problema de conexión con mongo, pruebe añadir el nombre de la base de datos a su cadena (Esto también puede dar fallos si el nombre de su base de datos contiene espacios).

## 5. Configuración del Servicio Sentry

Sentry es una plataforma de monitoreo de errores. Su tarea principal es identificar los errores que puedan aparecer después del lanzamiento de la aplicación y recopilar toda la información posible para hacer el proceso de *bug fixing* más fácil para el programador. Para configurar Sentry debe seguir los siguientes pasos:

1. Ingrese a la página principal de [Sentry](#) y cree una cuenta.
2. Asígnele un nombre a su organización y seleccione la región más cercana a usted en “Data Store Location”.  
Ejemplo de nombre de organización: CEHUM
3. Una vez creada su cuenta, haga clic en “Skip Onboarding” bajo los recuadros “Install Sentry” y “Set up my team”.
4. Diríjase a “Projects” y cree un nuevo proyecto.
5. Seleccione Next.js como la plataforma a monitorear.
6. Asígnele un nombre al proyecto.  
Ejemplo de nombre: cehum-humedata
7. A continuación se le hará entrega de un comando con el siguiente formato:  

```
npx @sentry/wizard@latest -i nextjs --saas --org <Su organización> --project <Nombre de su proyecto>
```

  
Una vez tenga el comando, debe hacer lo siguiente:
  - a. Localice el archivo `next.config.mjs` dentro de `apps/frontend` y elimínelo. Este contiene información que se va a sobrescribir y el no eliminarlo puede causar problemas.
  - b. Abra una terminal en `apps/frontend` y ejecute el comando entregado por Sentry.
  - c. Una vez ejecutado el comando, se le hará una serie de preguntas respecto a la configuración de su aplicación, las cuales están listadas en la siguiente tabla:

Instrucción	Respuesta	Explicación
Actualizar los paquetes de Sentry a la última versión	No recomendado	Las actualizaciones pueden causar conflictos, por lo que si no está seguro si es que estos puedan ocurrir, es mejor no actualizar.
Enrutar las solicitudes de Sentry a través del servidor Next.js	Si	Esto se hace para evitar problemas que puedan ser causados por AD Blockers.
Habilitar el rastreo para seguir el rendimiento de la aplicación	Si	Facilita el debug.

Habilitar "Session Replay" para obtener una reproducción de los errores que se produzcan durante la sesión de un usuario	Si	Facilita el debug.
Modificar el archivo <code>sentry.server.config.ts</code> existente	Si	Se asegura de que no haya información desactualizada.
Modificar el archivo <code>sentry.client.config.ts</code> existente	Si	Se asegura de que no haya información desactualizada.
Modificar el archivo <code>sentry.edge.config.ts</code> existente	Si	Se asegura de que no haya información desactualizada.
Agregar un bloque de código al archivo <code>instrumentation.ts</code>	No es necesario, continúe sin cambios	El archivo <code>instrumentation.ts</code> ya tiene la información que necesita.
Añadir la configuración de Sentry al archivo <code>next.config.mjs</code>	Si	Este archivo contiene información de su organización, por lo que es mejor que Sentry lo construya desde cero para no hacer cambios manualmente.
¿Usa una herramienta CI/CD para lanzar la aplicación?	Si, seleccione vercel	Vercel es la plataforma por donde estará operando el frontend.
Añadir la variable <code>SENTRY_AUTH_TOKEN</code> como variable de entorno	Copie la clave, guárdela en un lugar seguro y proceda	Esta variable le permitirá a su aplicación acceder a Sentry.
Seleccionar el administrador de paquetes del proyecto	Seleccione npm	Es el administrador de paquetes que utiliza el proyecto.

8. Asegúrese de guardar la variable `SENTRY_AUTH_TOKEN` en un lugar seguro, pues esta clave solo estará disponible en el momento de la creación de la misma.

- En el caso de que no haya guardado la variable, la haya perdido o se haya corupto, puede crear una nueva siguiendo los pasos a continuación:
- En el dashboard de Sentry haga clic en su organización e ingrese en "User auth tokens".
  - Haga clic en "Create new token".
  - Asígnele un nombre al token, otorgue permisos de administrador a todas las opciones que lo permitan y seleccione "Read and write" para "Alerts".  
Ejemplo de nombre: "humedata-token"
  - Copie el token creado y guárdelo en un lugar seguro, puesto que será indispensable más adelante.

## 6. Creación de la Cuenta Clerk

Clerk es la plataforma encargada de la gestión de usuarios de la aplicación humedata. Gracias a clerk, el proyecto cuenta con un sistema de autenticación y gestión de usuarios robusto, permitiendo funcionalidades como el inicio de sesión, registro, recuperación de contraseña y manejo de sesiones de forma segura y sencilla. Para conectar Clerk con el proyecto, debe seguir los siguientes pasos:

1. Ingrese a la página principal de [Clerk](#) y cree una cuenta.
2. Una vez creada su cuenta, cree una aplicación en la plataforma y asígnele un nombre acorde.  
Ejemplo de nombre: Humedata
3. Asegúrese de que en la pestaña llamada "Sign in options" estén seleccionadas las opciones "Email" y "Google", estas serán las opciones que tendrán los usuarios para iniciar sesión en la plataforma.
4. Haga clic en "Create application".
5. En estos momentos usted debería estar viendo un tutorial para instalar clerk en su aplicación. Como la plataforma ya se encuentra instalada en el proyecto, debe seguir los siguientes pasos:
  - a. Asegúrese de que el framework seleccionado sea "Next.js".
  - b. Tome nota de las variables `NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY` y `CLERK_SECRET_KEY`, ya que las necesitará más adelante.

→ **Información importante:** En total, Clerk requiere cuatro variables de entorno, estas son:

```
NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY
EXPO_PUBLIC_CLERK_PUBLISHABLE_KEY
CLERK_PUBLISHABLE_KEY
CLERK_SECRET_KEY
```

De las cuales solo `CLERK_SECRET_KEY` tiene un valor diferente, el resto de variables comparten un único valor, por lo que si ya cuenta con `NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY`, no necesita buscar el valor de las demás.

6. En el caso de que no aparezca el tutorial o haya perdido el valor de las variables, siga los siguientes pasos
  - a. En el dashboard de Clerk, en la parte superior izquierda, diríjase a la pestaña de configuración de la aplicación.
  - b. Bajo la sección "Development", seleccione la opción "API Keys".
  - c. Copie las variables anteriormente mencionadas y guárdelas en un lugar seguro.

## 7. Configuración de las Variables de Entorno

Las variables de entorno están especialmente pensadas para contener información que pueda cambiar dependiendo del entorno en el que se ejecute una aplicación. En este caso, las variables de entorno en el proyecto humedata contienen principalmente claves y vínculos para poder realizar las conexiones con las plataformas que utiliza. Para configurar las variables de entorno, debe seguir los siguientes pasos:

1. En el directorio raíz del proyecto, cree un archivo “.env.local” y ábralo en su editor de texto de preferencia.

Recomendación: Visual Studio Code

2. Utilice la siguiente plantilla para ingresar las variables que guardó en los capítulos anteriores:

```
# Frontend
SENTRY_AUTH_TOKEN="<Cadena de autorización de sentry>"
SENTRY_ORG="<Nombre de su organización de Sentry>"
SENTRY_PROJECT="react-native" # Backend

NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY="<Cadena pública de clerk>"
NEXT_PUBLIC_CLERK_SIGN_IN_URL=/sign-in
NEXT_PUBLIC_CLERK_SIGN_UP_URL=/sign-up
NEXT_PUBLIC_CLERK_AFTER_SIGN_IN_URL=/
NEXT_PUBLIC_CLERK_AFTER_SIGN_UP_URL=/ # Mobile
EXPO_PUBLIC_CLERK_PUBLISHABLE_KEY="<Cadena pública de clerk>"
CLERK_PUBLISHABLE_KEY="<Cadena pública de clerk>"
CLERK_SECRET_KEY="<Cadena secreta de clerk>"

EXPO_USE_METRO_WORKSPACE_ROOT=1

MONGO_URL="<Cadena de conexión MongoDB>"

CEHUM_URL="mysql://grafanareader:Viewer_1@168.232.165.73:3306/mqtt?host_user=root
&host_password=pP0V6Nk15mn7ZFW2id" # Frontend y Backend
```

3. Reemplace los valores que obtuvo de las plataformas anteriores en los campos correspondientes y asegúrese de guardar los cambios en su archivo “.env.local”.  
**Nunca comparta o suba este archivo a repositorios públicos, puesto que tiene información sensible.**

## 8. Setup del Frontend

Vercel es la plataforma que utiliza Humedata para lanzar el frontend de la aplicación. Vercel es útil para desplegar aplicaciones web construidas con frameworks como Next.js, React, Vue y Svelte, y a su vez está integrada con GitHub para volver a desplegar la aplicación una vez se haya realizado un commit en el repositorio.

Para configurar y desplegar el frontend, siga los siguientes pasos:

1. Inicie sesión con su cuenta de GitHub en la página principal de [Vercel](#).
2. En el dashboard de Vercel, cree un nuevo proyecto haciendo clic en el botón “Add new...” y luego “Project”.
3. Configure su nuevo proyecto de la siguiente manera:
  - a. Elija el repositorio de GitHub donde anteriormente subió el proyecto CEHUM Humedata.
  - b. Como “Root directory” configúralo como “apps/frontend”.  
Vercel detectará automáticamente que es un proyecto Next.js, confirme que este sea el caso en el campo “Framework preset”.
  - c. En “Build command” seleccione “Override” e introduzca el siguiente comando:

```
cd ../../ && pnpm web:build
```
  - d. En “Install command” seleccione “Override” e introduzca el siguiente comando:

```
pnpm install
```
  - e. Incluya todas las variables de entorno que agregó anteriormente en su archivo “.env.local”.  
Esto lo puede hacer de una manera muy sencilla copiando todo el contenido del archivo ([ctrl + a] + [ctrl + c] en VSCode) y pegarlo en el campo “Key” de una variable de entorno con [ctrl + v] dentro de Vercel. Haciendo esto, quedarán configuradas todas las variables automáticamente.
  - f. Para la variable MONGO\_URL, asegúrese de usar la URL de producción si está configurando para un entorno de producción.
4. Haga clic en “Deploy” para desplegar la aplicación. Vercel la construirá y desplegará automáticamente.  
Este proceso puede demorar varios minutos.
5. Visite la URL para asegurarse de que la aplicación se haya desplegado correctamente.
6. Si desea utilizar un dominio personalizado para su aplicación, puede configurarlo de la siguiente manera:
  - a. Ingrese a su proyecto en Vercel.
  - b. Haga clic en el botón “Domains”.
  - c. Una vez en domains, haga clic en “Add”.

- d. Asegúrese de ingresar un dominio válido, el ambiente correcto y la rama principal del proyecto.

Puede encontrar información más completa con respecto a la creación de dominios personalizados desde la [documentación oficial de Vercel](#).

## 9. Setup del Backend

AWS App Runner es un servicio que proporciona Amazon que, al igual que Vercel, te permite desplegar una aplicación directamente desde el código fuente. El proyecto Humedata utiliza App Runner para correr el backend de la aplicación. A continuación, tendrá que crear una cuenta de Amazon Web Services, guardar las variables de entorno usando Secrets Manager, crear un acceso a las mismas con IAM y lanzar la aplicación mediante la creación de un servicio de App Runner. Siga los pasos a continuación:

1. Cree una cuenta de Amazon Web Services. Para esto, necesitará vincular una tarjeta de crédito/débito puesto que el servicio produce costos mensuales, que por lo general, si tan solo tiene la aplicación lanzada, los costos mensuales suelen ser menores a 1 USD. Para comprobar todo lo que tenga que ver con costos en AWS, usted debe:
  - a. Ingresar a la consola de AWS.
  - b. Haga clic en su nombre de usuario y luego “cuenta”.
  - c. Todas las opciones que necesita para conocer los costos de los servicios que está utilizando en su cuenta de AWS se pueden ver en la sección “Facturación y pagos”.
2. Para guardar las variables de entorno utilizadas por el backend, utilizaremos Secrets Manager. Este servicio se encargará de que las claves se encuentren encriptadas y seguras. Para crear un secreto, haga lo siguiente:
  - a. Navegue a AWS Secrets Manager.
  - b. Haga clic en “Store a new secret”.
  - c. Seleccione “Other type of secret”.
  - d. Ingrese las siguientes variables de entorno desde su archivo .env.local:  
MONGO\_URL  
CEHUM\_URL  
CLERK\_SECRET\_KEY  
CLERK\_PUBLISHABLE\_KEY
  - e. Asigne un nombre al secreto  
Ejemplo: “cehum-humedata-secrets”
  - f. Complete el proceso de creación del secreto
  - g. Tome nota del ARN del secreto creado.
3. Una vez creados los secretos, la aplicación necesita una manera para acceder a ellos, por lo que necesita crear una política de IAM (Identity and Access Manager). Siga los pasos a continuación:
  - a. En la consola de AWS, diríjase a IAM.
  - b. Diríjase a la sección “Políticas”.
  - c. Cree una nueva política que permita el acceso al secreto creado.
  - d. Use el siguiente JSON como base para su política, reemplazando <ARN-del-secreto> con el ARN del secreto que creó anteriormente.



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["secretsmanager:GetSecretValue"],
      "Resource": ["<ARN-del-secreto*>"]
    }
  ]
}
```

IMPORTANTE: Reemplace únicamente el campo <ARN-del-secreto> sin eliminar el asterisco ni las comillas. El asterisco indica que el secreto tiene múltiples valores y las comillas indica que es una cadena. La falta de uno de los dos va a causar errores de lanzamiento.

- e. Asígnele un nombre a la política  
Ejemplo: CehumHumedataSecretsAccess

4. El backend, para acceder a las variables de entorno, necesita utilizar la política de acceso creada anteriormente, puesto que esta le permitirá a la aplicación ingresar al secreto que creó anteriormente. Esto lo puede lograr mediante la creación de un rol de IAM, el cual será utilizado por el servicio lanzado de App Runner para acceder a Secrets Manager. Cree un rol mediante los siguientes pasos.
  - a. En IAM, diríjase a la sección “Roles”.
  - b. Cree un nuevo rol.
  - c. Seleccione “Política de confianza personalizada” y escriba la siguiente política de confianza:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "tasks.apprunner.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- d. Adjunte la política “CehumHumedataSecretsAccess” que creó en el paso anterior.
  - e. Asígnele un nombre al rol.  
Ejemplo: CehumHumedataAppRunnerRole
5. Antes de lanzar el servicio de App Runner, al igual que hicimos en Vercel con el frontend, necesita asignar las configuraciones para que la aplicación pueda ser construida correctamente. Esto lo logramos mediante el archivo “apprunner.yaml”, el cual tiene tanto la información de construcción y lanzamiento de la app como las direcciones de los secretos para que desde el mismo código fuente se pueda acceder a ellos. Haga lo siguiente:

- a. En su repositorio local, abra el archivo ubicado en la siguiente dirección:  
apps/backend/apprunner.yaml
- b. Modifique la sección de secrets para que coincida con los ARNs de sus secretos:

```
secrets:
  - name: MONGO_URL
    value-from: "<ARN-del-secreto>:MONGO_URL::"
  - name: CEHUM_URL
    value-from: "<ARN-del-secreto>:CEHUM_URL::"
  - name: CLERK_SECRET_KEY
    value-from: "<ARN-del-secreto>:CLERK_SECRET_KEY::"
  - name: CLERK_PUBLISHABLE_KEY
    value-from: "<ARN-del-secreto>:CLERK_PUBLISHABLE_KEY::"
```

Reemplace el campo <ARN-del-secreto> con el ARN del secreto que creó anteriormente.

- c. Guarde los cambios y haga un commit al repositorio.  
Ejemplo de nombre para el commit: Actualización apprunner.yaml

6. Una vez hechas las configuraciones para las variables de entorno y las instrucciones de construcción del proyecto, puede proceder a lanzar el servicio de App Runner realizando lo siguiente:

- a. En la consola, navegue hasta AWS App Runner.
- b. Haga clic en "Create an App Runner service".
- c. En "Source and deployment", seleccione:
  - Source: GitHub.
  - Cree una conexión con su cuenta de GitHub.
  - Seleccione el repositorio del proyecto CEHUM Humedata.
  - Branch: main (o la rama que desee desplegar).
  - Deployment trigger: Automatic.
  - Directory: apps/backend.
- d. En "Build settings":
  - Compilation configuration: use configuration file (apprunner.yaml).
- e. En "Service settings":
  - Service name: cehum-humedata-backend.
  - Environment variables: No agregue ninguna aquí, las obtendremos de Secrets Manager.
- f. En "Security":
  - Instance role: Seleccione el rol IAM que creó anteriormente.
- g. Una vez completadas las configuraciones, termine con el proceso de creación de servicio.

El lanzamiento del backend puede durar varios minutos.

7. Una vez completado el despliegue, verifique que todo está funcionando correctamente probando la URL que le proporcionará App Runner.

8. Dentro del proyecto, localice el archivo `config.ts` dentro de la dirección `apps/backend/src/config.ts`
9. Abra el archivo y reemplace el valor de la variable `PRODUCTION_URL` por la URL que designó App Runner al crear su proyecto.  
Asegúrese de que la URL no termine con un símbolo `/`, esto puede causar problemas de conexión. Ejemplo de `PRODUCTION_URL` válida:  
`PRODUCTION_URL = "https://<id>.<region>.awsapprunner.com"`
10. Realice un commit a su repositorio de GitHub  
Ejemplo de nombre para el commit: Actualización a `PRODUCTION_URL`

Puede encontrar más información detallada sobre el lanzamiento de aplicaciones con AWS App Runner mediante los siguientes vínculos de la documentación oficial de App Runner:

- [Documentación oficial de AWS App Runner.](#)
- [Documentación de Secrets Manager.](#)
- [Documentación de IAM.](#)
- [Documentación de los archivos de configuración `apprunner.yaml`.](#)

## 9.1. Conexión del backend a MongoDB en modo producción

En un proyecto de desarrollo web, la fase de producción es cuando se lanza la aplicación desarrollada a un ámbito real, de cara a usuarios reales. El backend lanzado en AWS App Runner se puede comunicar con la base de datos MongoDB de dos maneras diferentes dependiendo del ámbito en el que el proyecto se encuentre. Si se utiliza en un ámbito no dirigido a clientes, es decir, la aplicación como herramienta personal de gestión de boyas o durante el desarrollo de la misma, puede hacer lo siguiente:

1. Diríjase al dashboard de su proyecto en MongoDB Atlas.
2. Bajo la pestaña "Security" haga clic en "Network Access".
3. Añada una nueva dirección IP haciendo clic en "Add IP address".
4. Añada la dirección IP `0.0.0.0/0` (Allow access from anywhere).

Pero si desea establecer una conexión en un ámbito de producción, donde se requiera una mayor integridad de los datos y seguridad entre conexiones, puede comunicar el backend con MongoDB a través de una Virtual Private Cloud, la cual es una red privada que ofrece Amazon para manejar el tráfico de salida de su aplicación de forma segura.

Debe tener en cuenta que al utilizar este servicio, los costos subirán no solo por parte de Amazon, si no que por parte de MongoDB también. El uso de conexiones VPC solo está disponible para clusters dedicados (Plan M10 o superior, en el cual se paga \$0.12 USD a la

hora), por lo que antes de configurar su VPC, debe actualizar el plan de su cluster. Puede hacerlo de la siguiente manera:

1. En el dashboard de su proyecto, en la barra izquierda bajo la sección “Database” ingrese a “Clusters”.
2. En el cluster que desea mejorar, busque la opción “Edit Configuration”.
3. Seleccione la opción M10 y aplique los cambios.  
Aparte de poder establecer una conexión más segura, también obtendrá 10GB de almacenamiento, memoria RAM y CPU dedicadas a su base de datos, la capacidad de guardar copias de seguridad de sus datos, entre otras ventajas.
4. Añada un método de pago y lance su cluster.

Una vez hecho esto, puede configurar la VPC en la consola de AWS App Runner. Para ello, haga lo siguiente:

1. Diríjase a la consola de App Runner e ingrese a la página de su servicio.
2. Baje hasta el registro de eventos de su servicio e ingrese a configuración.
3. Haga clic en el botón “editar” en la sección llamada “Configurar servicio” y busque la categoría “Redes”.
4. En “Redes”, seleccione las opciones “Punto de conexión público” y “VPC personalizada”.
5. Cree un nuevo conector de VPC, para ello haga clic en “Agregar nueva” en la parte inferior de la sección.
6. Asígnele un nombre a su conector, seleccione la VPC por defecto de Amazon, escoja al menos 3 subredes, seleccione el grupo de seguridad por defecto de Amazon y guarde los cambios hechos en su servicio.  
Si no ha tocado la consola de AWS VPC de antemano, debería tener únicamente disponibles las opciones por defecto que crea Amazon.  
Ejemplo de nombre: HumedataVPCConnector.
7. Diríjase al dashboard de su proyecto MongoDB Atlas y bajo la sección “Security” seleccione “Network access”.
8. Vaya a la pestaña “Peering” y haga clic en “Add outbound connection”, de esta manera conectará su servicio App Runner con MongoDB Atlas.
9. En otra pestaña diríjase a la consola de AWS e ingrese a “AWS Virtual Private Network” mediante el buscador.

10. Bajo la sección “Virtual Private Network”, ingrese a “Your VPCs” y seleccione la VPC por defecto que crea Amazon. Necesitará la información de la tabla que se encuentra en la parte inferior de la página.
11. Regrese a MongoDB Atlas, seleccione AWS como su proveedor de nube y a partir de los datos de la tabla, rellene el formulario y haga clic en “Initiate peering”.  
Necesitará: Account ID (Owner ID en la tabla), VPC ID, IPv4 CIDR y la región en la que lanzó su servicio.
12. Mongo enviará una solicitud de autorización de conexión a su servicio de AWS. Para aceptarla, diríjase a la consola de VPC, y bajo la sección “Virtual Private Network” ingrese a “Peering connections”. Una vez allí, puede aceptar la solicitud.
13. Una vez aceptada la solicitud, diríjase a “Route tables” en la consola de VPC. Seleccione la *route table* que esté asociada a las subredes que seleccionó anteriormente.  
Puede revisar las asociaciones de su route table en la sección “subnet associations”. Basta con que sea una asociación no explícita para que funcione.
14. Haga clic en “Routes” y en “Edit routes”.
15. Diríjase a MongoDB Atlas y de la *peering connection* que creó anteriormente, copie el Atlas CIDR.  
Por lo general, el Atlas CIDR será de la forma 192.168.248.0/21.
16. De vuelta en la consola de VPC, haga clic en “Add route”, pegue el Atlas CIDR, seleccione la opción “Peering connection” y seleccione la conexión con MongoDB Atlas que creó anteriormente. Guarde los cambios.
17. Por último, regrese a la consola de App Runner y reinicie el lanzamiento de su servicio haciendo clic en “Deployment”.

Puede encontrar más información detallada sobre el uso de VPC en AWS App Runner para el manejo seguro de conexiones outbound en el entorno de producción mediante los siguientes vínculos de la documentación oficial de AWS Virtual Private Cloud y App Runner.

- [Configuración de la VPC en el servicio de App Runner](#)
- [Uso de App Runner con endpoints de VPC](#)
- [Documentación oficial AWS Virtual Private Cloud](#)
- [Conectar la VPC a otras redes](#)
- [Documentación de las subredes y tablas de enrutamiento](#)

## 10. Ejecución en Modo Desarrollo

Cuando usted desee realizar cambios experimentales en el código, para poder probarlos de manera rápida en un ambiente local, puede correr la aplicación en modo desarrollo. De esta manera, correrá la aplicación en su propia máquina por lo que cualquier cambio que haga no tomará efecto en la versión que usted lanzó mientras no haga un commit de los cambios en GitHub.

1. El primer paso para ejecutar la aplicación en modo desarrollo es instalar las dependencias del proyecto, esto ya lo realizó en la configuración inicial del proyecto. Para instalar las dependencias debe ejecutar el siguiente comando desde la raíz del proyecto:  
`pnpm install`
2. Para ejecutar el frontend en modo desarrollo, haga lo siguiente:
  - a. Desde el directorio raíz del proyecto, en una terminal que no vaya a utilizar para otra cosa, ejecute el comando:  
`pnpm web:dev`
  - b. Esto iniciará el servidor de desarrollo de Next.js. Para ver la aplicación web, desde su navegador ingrese a:  
<https://localhost:3000>
3. Para ejecutar el backend en modo desarrollo, haga lo siguiente:
  - a. Desde el directorio raíz del proyecto, ejecute el comando:  
`pnpm backend:dev`
  - b. Esto iniciará el servidor de desarrollo de Express. Para acceder al backend ingrese a  
<https://localhost:4000>  
Asegúrese de que en MongoDB Atlas, en la sección de Network access se encuentre añadida su dirección IP.

## 11. Requisitos previos de la aplicación móvil

La aplicación móvil cuenta con soporte para Android y para iOS, los requisitos previos para preparar la aplicación son los siguientes:

1. Instale Java Development Kit 17, para ello:
  - a. Ingrese a la página oficial de Oracle para descargar el instalador de [JDK 17](#) para su sistema operativo.
  - b. Una vez instalado, ingrese en el buscador de windows “Editar las variables de entorno del sistema”.
  - c. Ingrese en la sección “Opciones avanzadas” y haga clic en variables de entorno.
  - d. En “Variables de sistema”, cree una nueva variable de nombre JAVA\_HOME y de clave el directorio donde se encuentra instalado JDK 17.  
Si instaló todo por defecto, este será similar a C:\Program Files\Java\jdk-17.  
Compruebe esto ingresando la dirección en su explorador de archivos.
  - e. Ingrese en la variable “Path” que se encuentra en variables de sistema y agregue lo siguiente como un nuevo valor:  
%JAVA\_HOME%\bin
2. Si desea lanzar en modo desarrollo la aplicación, instale Android Studio. Para ello:
  - a. Visite la página oficial de [Android Studio](#) y descargue el instalador.
  - b. Busque nuevamente “Editar las variables de entorno del sistema”.
  - c. Ingrese en la sección “Opciones avanzadas” y haga clic en variables de entorno.
  - d. En “Variables de sistema”, cree una nueva variable de nombre ANDROID\_HOME y de clave el directorio donde se encuentra instalado Android Studio.  
Si instaló todo por defecto, este será similar a  
C:\Users\<su-usuario>\AppData\Local\Android\Sdk. Compruebe esto ingresando la dirección en su explorador de archivos.
  - e. Ingrese en la variable “Path” que se encuentra en variables de sistema y agregue los siguientes nuevos valores:  
%ANDROID\_HOME%\platform-tools  
%ANDROID\_HOME%\emulator  
%ANDROID\_HOME%\tools  
%ANDROID\_HOME%\tools\bin
3. Si se encuentra en macOS y desea desarrollar para iOS, instale XCode desde la App Store.  
Para más información acerca del desarrollo de aplicaciones móviles para iOS en XCode, puede acceder a la [página web de documentación oficial](#).

## 12. Configuración inicial de la Aplicación Móvil

1. Copie el archivo `.env.local` que creó anteriormente en la raíz del proyecto a la carpeta `/apps/mobile`.
2. Asegúrese de que las variables específicas de mobile estén correctamente configuradas con los valores vistos en la plantilla anterior, estas son:  
`EXPO_USE_METRO_WORKSPACE_ROOT`  
`EXPO_PUBLIC_CLERK_PUBLISHABLE_KEY`  
`SENTRY_ORG`  
`SENTRY_PROJECT`
3. Instale las dependencias del proyecto. Para ello abra una terminal en la raíz del proyecto mobile (`/apps/mobile`) e ingrese el siguiente comando:  
`pnpm prebuild`

### 12.1. Construir la aplicación para Android

Para construir la versión de producción para la aplicación de android debe realizar lo siguiente:

1. Abra una terminal en la raíz del proyecto mobile.
2. Ejecute el siguiente comando:  
`pnpm build:production:android`
3. Ingrese las credenciales de Expo de los autores del proyecto y proceda con el proceso de construcción.  
Este proceso puede tardar varios minutos y requiere tener Android Studio correctamente configurado.



## 13. Ejecución de la Aplicación Móvil en Modo Desarrollo

Para poder ejecutar la aplicación móvil, debido a que se debe desarrollar en un ambiente de windows totalmente diferente al ambiente de un teléfono celular, se requiere un emulador de ambiente Android y iOS respectivamente. Tanto en Android como en iOS tiene dos opciones, utilizar el emulador incluido en Android Studio y XCode respectivamente o conectar un dispositivo físico.

Esta guía se fijará más a detalle en el proceso para realizarlo en un ambiente Android, pero más adelante podrá encontrar documentación para el emulador de iOS incluido en XCode. Siga los siguientes pasos:

1. Si desea correr la aplicación en un dispositivo virtual creado en Android Studio haga lo siguiente:
  - a. Entre a Android Studio.
  - b. Al lado del buscador de proyectos, ingrese a “Virtual Device Manager” luego de hacer clic en los tres puntos.
  - c. Haga clic en el símbolo “+” para crear un nuevo dispositivo virtual.
  - d. Seleccione el dispositivo donde quiera correr la aplicación.  
Recomendación: Crear un dispositivo virtual desde Pixel 6 en adelante.
  - e. Asígnele un nombre a su dispositivo y haga clic en “Finish”.  
Por lo general, al hacer clic en finish necesitará descargar algunas dependencias dependiendo del dispositivo, este proceso puede tardar unos minutos.
  - f. Una vez creado, lance su dispositivo virtual haciendo clic en el botón “Start”.
  - g. Abra una terminal en la raíz del proyecto mobile e ingrese el siguiente comando:  
`pnpm android`
2. Si desea correr la aplicación desde un dispositivo físico haga lo siguiente:
  - a. Desde su dispositivo android, ingrese a configuraciones, luego seleccione a la opción “Acerca del teléfono” e ingrese a “Información de software”.
  - b. Active las opciones de desarrollador presionando múltiples veces seguidas la opción “Número de compilación”.
  - c. Regrese a la sección principal de las configuraciones y como última opción aparecerá “Opciones de desarrollador”, ingrese a ellas.
  - d. Busque la opción “Depuración por USB” y actívela.
  - e. Conecte su dispositivo android a su computadora con un cable USB.
  - f. Abra una terminal en la raíz del proyecto mobile e ingrese el siguiente comando:  
`pnpm android`
3. Si desea correr la aplicación para iOS, en una terminal en la raíz del proyecto mobile ingrese el siguiente comando:  
`pnpm ios`  
Puede conocer información sobre la emulación de iOS desde la [documentación oficial de iOS Simulator](#).

**Nota del revisor técnico:** Aún hay problemas de lanzamiento que falta por resolver, estos son:

- ☐ Asegurarse de conectar correctamente el backend a la base de datos MySQL para que este obtenga la información de las boyas.
- ☐ El componente de inicio de sesión de Clerk todavía muestra el label "Development mode".
- ☐ Existe un error no resuelto que marca Sentry con respecto a clerk, es el siguiente:  
"auth() was called but Clerk can't detect the usage of clerkMiddleware()".