

## HINTS

### CÓMO ALOJAR TUS PROYECTOS FRONT END

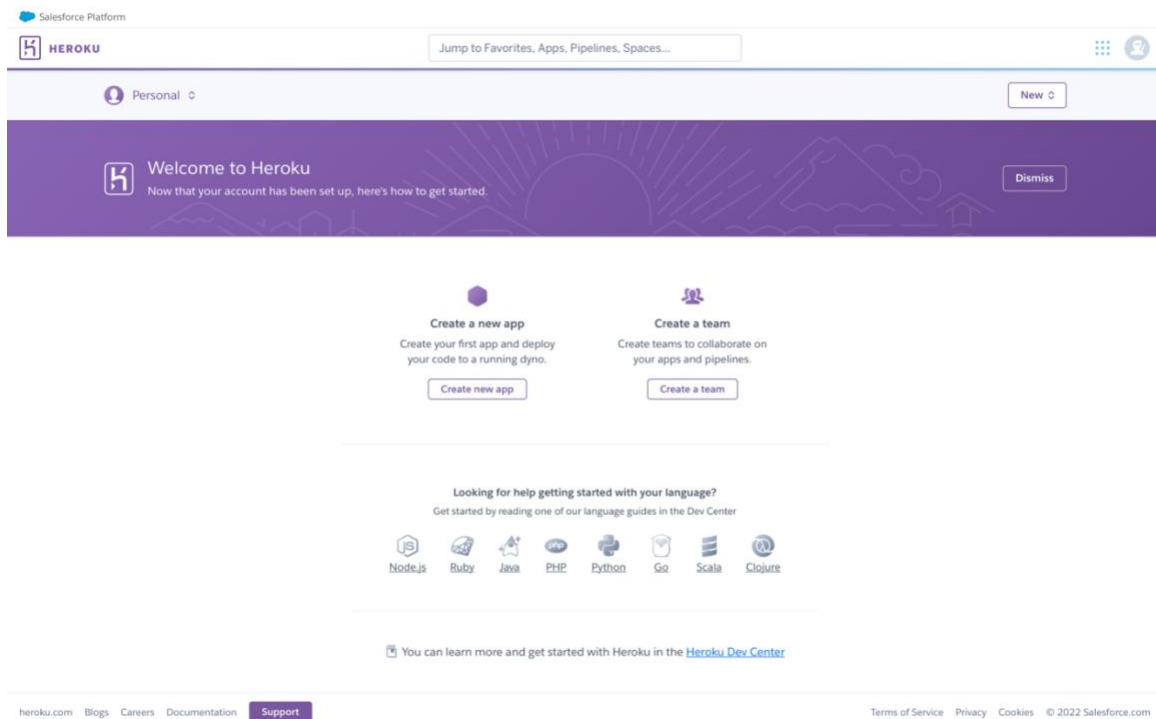
Para cargar proyectos front-end a un servidor, solo necesita cargar sus archivos estáticos a tu proveedor de alojamiento.

### CÓMO ALOJAR TUS PROYECTOS FULLSTACK JAVA, JAVASCRIPT Y PYTHON

Para cargar proyectos fullstack de cualquier tipo, primero debes encontrar un lugar para almacenar tu interfaz frontal y otro lugar para almacenar tu backend en un servidor. La mayoría de los servicios de hosting te permiten almacenar estas dos piezas en sus servidores.

Un buen servicio para alojar nuestras aplicaciones Full Stack, ya sea Full Stack Java, Full Stack JavaScript o Full Stack Python es Heroku. Existen otras alternativas como Fireship o AWS pero en este Hints veremos cómo alojar un proyecto Full Stack Java en Heroku (puedes ver la documentación de Heroku [aquí](#)).

Para usar Heroku, lo primero que debemos hacer es crear una cuenta gratuita de Heroku. Puedes registrarte [aquí](#). Una vez que confirmes tu correo electrónico, serás recibido en la siguiente página de inicio.



Antes de continuar, asegúrate de tener Java 8, Maven 3, Git y Postgres instalados localmente.

Aquí podrás seguir los pasos que demostraremos de manera gráfica en el sitio de Heroku pero en este ejemplo seguiremos usando los comandos respectivos desde la línea de comandos de nuestro computador.

Después de instalar las herramientas necesarias, todo lo que nos queda por hacer es instalar heroku, que puede descargar [aquí](#). Ahora seguiremos el ejemplo en la documentación de Heroku clonando el proyecto ejemplo Full Stack Java de Heroku y cambiando nuestro directorio usando los siguientes comandos.

```
1 git clone https://github.com/heroku/java-getting-started
2 cd Java-primeros pasos
```

Ahora podemos implementar la aplicación. Para hacer esto, debemos crear una aplicación en Heroku para prepararla para recibir tu código fuente. Así se podrá después implementar el código en línea. El comando que hay que usar para crear una aplicación es la siguiente.

```
1 Heroku create
```

Heroku elige un nombre generado aleatoriamente de forma predeterminada. Al elegir el nombre de una aplicación, ten en cuenta que Heroku solo acepta nombres que comienzan con una letra, terminan con una letra, o un dígito, y solo pueden contener letras minúsculas, dígitos y guiones.

En este caso, Heroku creó un repositorio git para alojar nuestra aplicación y, al mismo tiempo, creó un dominio de alojamiento gratuito para mantener la aplicación en línea. En este caso, <https://arcane-oasis-04688.herokuapp.com> es el dominio. Heroku eligió el nombre arcane-oasis-04688. Si visitamos la url de la aplicación podemos ver la siguiente página.

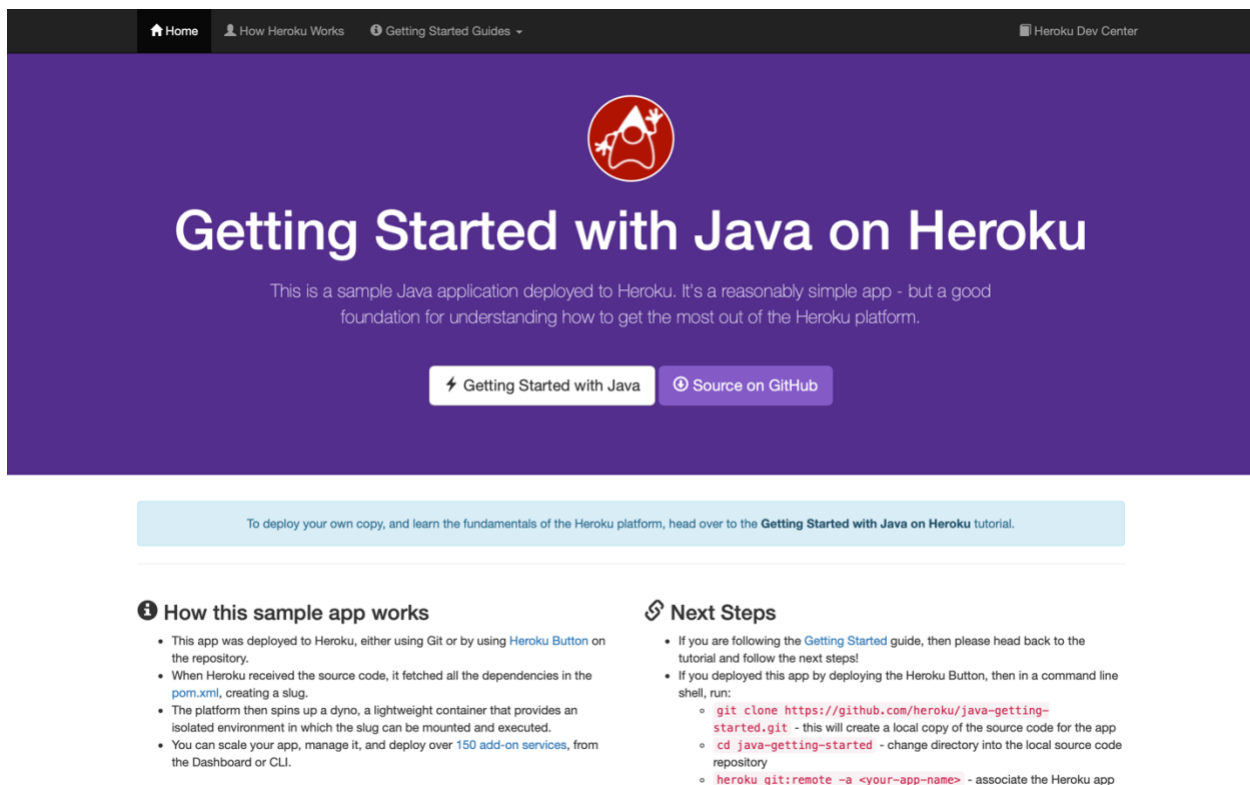
## Heroku | Welcome to your new app!

Refer to the [documentation](#) if you need help deploying.

Ahora tenemos un espacio para alojar nuestra aplicación, pero esta no tiene código fuente. Para cargar nuestra aplicación Java de ejemplo en Heroku, debemos usar el siguiente comando:

```
1 git push heroku main
```

Esto cargará la aplicación, o el “código fuente”, en Heroku y la implementará en línea (esta acción también es conocida como un “deploy” de la aplicación). Si ahora volvemos a verificar el dominio de nuestra aplicación, podremos ver que nuestra aplicación está completamente implementada.



The screenshot shows the Heroku 'Getting Started with Java' page. At the top, there's a navigation bar with links: Home, How Heroku Works, Getting Started Guides, and Heroku Dev Center. The main heading is 'Getting Started with Java on Heroku' with a red rocket icon. Below the heading, it says: 'This is a sample Java application deployed to Heroku. It's a reasonably simple app - but a good foundation for understanding how to get the most out of the Heroku platform.' There are two buttons: 'Getting Started with Java' and 'Source on GitHub'. A light blue banner below the buttons says: 'To deploy your own copy, and learn the fundamentals of the Heroku platform, head over to the [Getting Started with Java on Heroku](#) tutorial.' The page is divided into two columns. The left column is titled 'How this sample app works' and lists four bullet points: 1. The app was deployed to Heroku, either using Git or by using [Heroku Button](#) on the repository. 2. When Heroku received the source code, it fetched all the dependencies in the [pom.xml](#), creating a slug. 3. The platform then spins up a dyno, a lightweight container that provides an isolated environment in which the slug can be mounted and executed. 4. You can scale your app, manage it, and deploy over [150 add-on services](#), from the Dashboard or CLI. The right column is titled 'Next Steps' and lists two bullet points: 1. If you are following the [Getting Started](#) guide, then please head back to the tutorial and follow the next steps! 2. If you deployed this app by deploying the Heroku Button, then in a command line shell, run: 

- `git clone https://github.com/heroku/java-getting-started.git` - this will create a local copy of the source code for the app
- `cd java-getting-started` - change directory into the local source code repository
- `heroku git:remote -a <your-app-name>` - associate the Heroku app

¡Estupendo! Acabamos de implementar nuestra primera aplicación Full Stack Java.

Quizás te preguntes si la aplicación ya está conectada a su propia base de datos y la respuesta es sí. En Heroku, al hacer un deploy de una aplicación, se le añade una base de datos de Postgres por defecto, lo cual es un complemento gratuito que podemos comprobar mediante el comando `heroku addons`.

Para ver la URL exacta de la base de datos que usa nuestra aplicación, podemos usar el comando de configuración de heroku. Usar el comando `heroku pg` también nos permitirá ver otros detalles.

De esta forma se puede alojar un proyecto en un servicio hosting realizando un deploy de un ejemplo de código.

## **CÓMO ALOJAR TUS PROYECTOS ANDRIOD**

De acuerdo con la documentación para desarrolladores de Android, puedes alojar tus proyectos Android en un servidor lanzando o publicando tus aplicaciones. Esto se puede hacer de varias maneras. Por lo general, se lanzan aplicaciones a través de un mercado de aplicaciones como Google Play, pero también puede lanzar aplicaciones en tu propio sitio web o enviando una aplicación directamente a un usuario.