



El futuro digital
es de todos

MinTIC



CICLO III: Desarrollo de software

Mision
TIC2022





El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

Sesión 17: Desarrollo Software

Administrar un servidor en la nube para albergar una aplicación WEB

**Mision
TIC2022**



Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Explicar y aplicar el servidores web y WSGI.
2. Explicar los conceptos básicos de Heroku.
3. Migrar su aplicación a un servidor web alojado en Heroku.
4. Configurar y administrar su aplicación en Heroku.

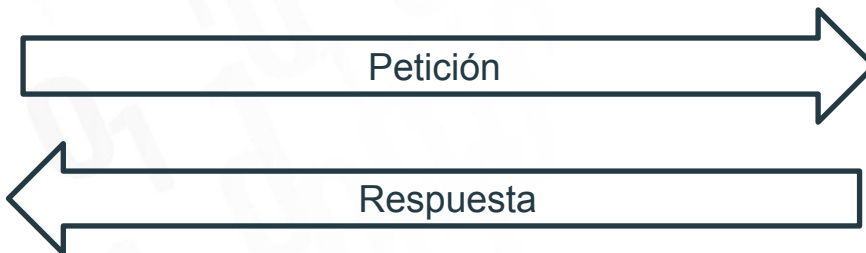


Servidores

Cliente



Servidor





Servidores

- El término de servidor en tecnología, hace referencia a un equipo de cómputo en el cual se procesan solicitudes y entregan las respuestas a otros equipos, a los que se les llama clientes.
- Más que un objeto, el servidor es un término que se le asocia al computador destinado a recibir y procesar solicitudes dentro de la arquitectura diseñada, por este motivo el servidor debe tener amplia memoria y capacidad de procesamiento para atender de manera oportuna las peticiones entrantes.
- Las solicitudes a un servidor pueden realizarse a través de una red local o de internet.



Servidores

Existen diferentes tipos de servidores, estos se pueden agrupar de acuerdo a su uso, o al servicio que ofrecen.

De acuerdo a su uso se pueden clasificar en dos tipo:

- **Servidor dedicado:** dedica todos sus recursos a atender solicitudes de los equipos cliente.
- **Servidor compartido:** además de utilizar sus recursos para servir las peticiones de los clientes, también es utilizado por un usuario para trabajar de forma local.



Servidores

Existen varios tipos servidores de acuerdo al servicio que ofrecen o la función que cumplen dentro de la arquitectura de la red, los más comunes son:

- **Servidor de archivos:** es aquel que almacena y provee archivos.
- **Servidor de Directorio Activo/Dominio:** guarda la información sobre los usuarios, equipos y grupos de una red.
- **Servidor de Correo:** envía, recibe y almacena los correos de una organización.
- **Servidor Proxy:** entre otras funciones, guarda en memoria caché las páginas web a las que acceden los usuarios de la red durante un cierto tiempo, haciendo más rápida la respuesta hacia el usuario las siguientes veces que este acceda.



Servidores

- **Servidor Web:** almacena contenido web y lo pone al servicio de los usuarios.
- **Servidor de Base de Datos:** provee servicios de base de datos a otros programas o equipos.
- **Servidor DNS:** establece la relación entre los nombres de dominio y las direcciones IP de los equipos de una red.
- **Servidor DHCP:** dispone de una rango de direcciones con el cual, asigna automáticamente los parámetros de configuración de red IP a los equipos cliente.
- **Servidor FTP:** permite el intercambio de ficheros entre equipos.



Servidores

Servidor Web

- Un servidor web es un software que se instala en el equipo de cómputo y cuyo objetivo es devolver información (páginas web) cuando los usuarios realizan las peticiones de acceso. La respuesta a estas peticiones se transmiten a los usuarios a través de los navegadores web mediante el protocolo HTTP (Hipertext Transfer Protocol).
- Cuando un usuario escribe una dirección web, está solicitando información al servidor, es por esto que el servidor web siempre debe estar a la espera de una solicitud de información, de lo contrario no será posible para el usuario visitar la página web deseada.



Servidores

En la actualidad, los servidores web más comunes son: Apache, IIS, Lighttpd, y Jagsaw.

Servidor Apache

Es el servidor web más popular del mundo, y fue desarrollado por Apache Software Foundation. Apache es un software de código abierto y puede instalarse en casi todos los sistemas operativos, incluyendo Linux, Unix, Windows, FreeBSD, Mac OS X entre otros. Alrededor del 60% de las máquinas de servidor web ejecutan Apache como servidor web para alojar las aplicaciones.

Internet Information Server (IIS)

Internet Information Server (IIS) es un servidor Web de alto rendimiento que pertenece a la compañía Microsoft. Este servidor se ejecuta en plataformas Windows y debido a que está integrado con el sistema operativo es más sencilla su configuración.



Servidores

Lighttpd

Lighttpd es un servidor web gratuito y de código abierto que se distribuye con el sistema operativo FreeBSD. Este servidor se destaca porque su rapidez, es seguro y consume mucho menos energía de la CPU. Lighttpd puede ejecutarse también en sistemas operativos Windows, Mac OS X, Linux y Solaris.

Jigsaw Server

Jigsaw (W3C Server, World Wide Web Consortium). Es un servidor web gratuito, de código abierto que puede ejecutarse en diferentes sistemas operativos como: Linux, Unix, Windows, Mac OS X Free BSD, entre otros. Jigsaw está desarrollado en Java y puede ejecutar secuencias de comandos CGI y programas PHP.



Servidores

- Antes de elegir un servidor web, se debe tener en cuenta qué tan bien funciona con el sistema operativo, la compatibilidad con otros servidores, las características de seguridad, y las herramientas necesarias para la publicación de información.
- Es importante mencionar que el término de servidor web, puede referirse tanto al software que permite la publicación de páginas web como al equipo de cómputo dónde se instala dicha aplicación. De acuerdo al contexto de su uso el lector deberá ser capaz de interpretar su significado.



Servidores - WSGI

- Web Server Gateway Interface (WSGI) es un estándar creado por la comunidad web Python que permite escribir programas, los cuales puedan comunicarse a través internet.
- WSGI permite a través de un llamado simple que los servidores web envíen solicitudes a aplicaciones web o frameworks escritos en el lenguaje de programación Python.
- La versión actual de WSGI (1.0.1), se especifica en [Python Enhancement Proposal \(PEP\) 3333](#).



Servidores - WSGI

- WSGI se especificó primeramente en PEP-333 en el año 2003, la actualización de esta versión se publicó en el año 2010 con el nombre de PEP-3333.
- La interfaz WSGI tiene dos lados, el lado del "servidor" o "puerta de enlace" y el lado de la "aplicación" o "framework".
- Del lado del servidor se invoca un objeto proporcionado por el lado de la aplicación. Dependerá del servidor cómo se proporciona ese objeto.



Servidores - WSGI

Del lado de la aplicación o framework:

- El objeto *aplicación* es simplemente un objeto invocable que acepta dos argumentos.
- El término "objeto" no debe malinterpretarse en el sentido de que requiere una instancia de objeto real: una función, método, clase o instancia con un método `__call__` son aceptables para su uso como objeto de aplicación.
- Los objetos de la aplicación deben poder invocarse más de una vez, ya que prácticamente todos los servidores / puertas de enlace (excepto CGI) realizarán solicitudes repetidas.



Servidores - WSGI

Ejemplo de objeto de aplicación como función:

```
def simple_app(environ, start_response):  
    """Simplest possible application object"""  
    status = '200 OK'  
    response_headers = [('Content-type', 'text/plain')]  
    start_response(status, response_headers)  
    return ['Hello world!\n']
```




Servidores - WSGI

Ejemplo de objeto de aplicación como clase:

```
class AppClass:
    def __init__(self, environ, start_response):
        self.environ = environ
        self.start = start_response

    def __iter__(self):
        status = '200 OK'
        response_headers = [('Content-type', 'text/plain')]
        self.start(status, response_headers)
        yield "Hello world!\n"
```



Servidores - WSGI

Del lado servidor:

- El servidor o la puerta de enlace invoca la aplicación una vez por cada solicitud que recibe de un cliente HTTP, que se dirige al aplicativo web.
- En el siguiente enlace se encuentra un ejemplo sobre puerta de enlace CGI simple, implementada como una función que toma un objeto de aplicación.

<https://www.python.org/dev/peps/pep-0333/#preface>



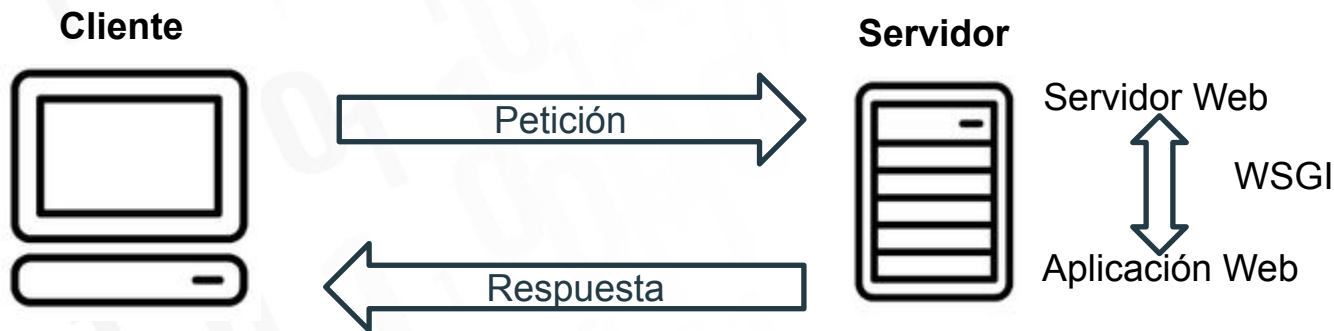
Servidores - WSGI

¿Por qué se necesitan ?

- WSGI no es más que un intermediario entre el servidor web y la aplicación desarrollada en Python que es capaz de interpretar las peticiones que recibe el servidor web (que están dirigidas al aplicativo web) y enviárselas al aplicativo de una manera que este las entienda.
- Es por lo anterior, que WSGI tiene la especificación del lado del cliente y del servidor, como se mostró anteriormente.
- Sin este “interprete” la aplicación podría no comprender todas las peticiones que recibe el servidor web por parte de los clientes.



Servidores - WSGI





El futuro digital
es de todos

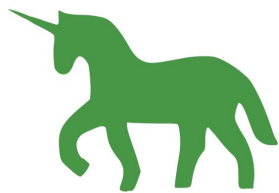
MinTIC



Vigilada Mineducación

Servidores - WSGI

Gunicorn (Green Unicorn)



gunicorn

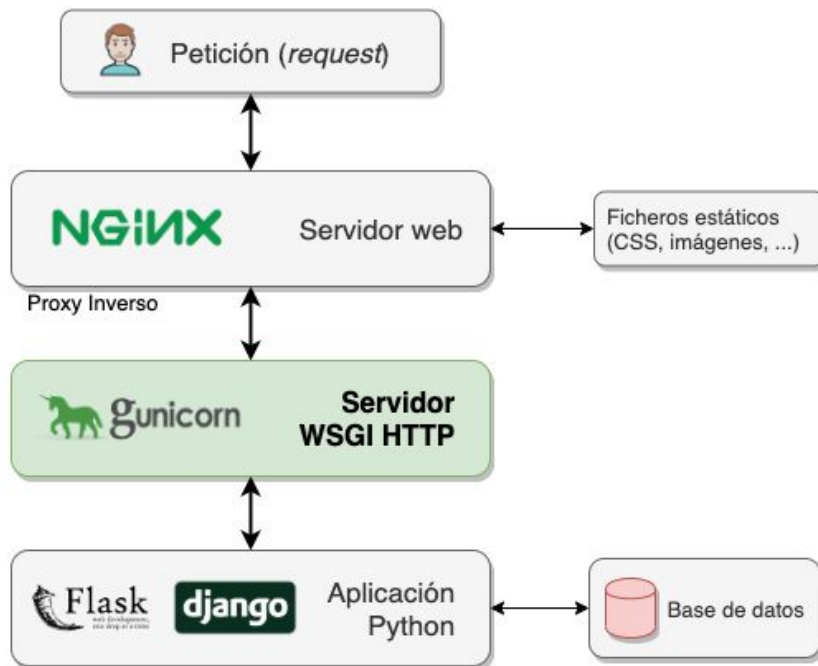
uWSGI

uWSGI





Servidores - WSGI





Servidores - WSGI

- Existen muchísimas formas por medio de las cuales una aplicación web se puede comunicar con el servidor web, por lo tanto la aplicación debe “aprender” todas esas formas para entender la información que reciba del servidor web.
- Esto hace que las aplicaciones python, tengan que adaptar el código para adaptarse a los requerimientos del servidor web. Sin embargo, implementando el intérprete WSGI esta tarea ya no es necesaria.
- Para mayor información puede dirigirse a la documentación de WSGI:
<https://wsgi.readthedocs.io/en/latest/learn.html>



Servidores - WSGI

PythonAnywhere configura automáticamente un uWSGI al momento de desplegar su aplicación web.

```
2021-10-14 22:23:29 *** Starting uWSGI 2.0.19.1 (64bit) on [Thu Oct 14 22:23:28 2021] ***
2021-10-14 22:23:29 compiled with version: 9.3.0 on 27 May 2021 21:02:35
2021-10-14 22:23:29 os: Linux-5.4.0-1038-aws #40 SMP Thu Mar 4 18:32:33 UTC 2021
2021-10-14 22:23:29 nodename: blue-liveweb10
2021-10-14 22:23:29 machine: x86_64
2021-10-14 22:23:29 clock source: unix
2021-10-14 22:23:29 pcre jit disabled
2021-10-14 22:23:29 detected number of CPU cores: 4
```




Heroku

- Heroku es una Plataforma como Servicio (PaaS) en la nube basada en contenedores gestionados.
- Los desarrolladores utilizan Heroku para desplegar, gestionar y escalar aplicaciones modernas.
- La experiencia del desarrollador de Heroku es un enfoque centrado en la aplicación para la entrega de software, integrado con las herramientas y los flujos de trabajo de desarrolladores más populares de la actualidad.



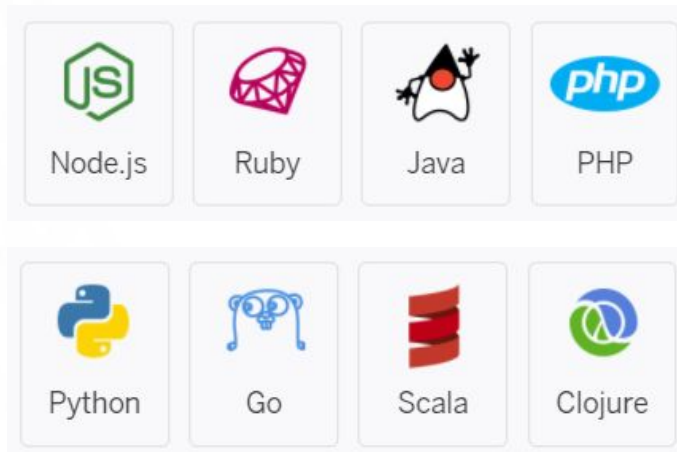
HEROKU



Heroku



- Lenguajes oficialmente soportados.
 - Node.js
 - Ruby
 - Java
 - PHP
 - Python
 - Go
 - Scala
 - Clojure





Heroku - Características

- **Heroku Runtime:**
 - Heroku ejecuta sus aplicaciones dentro de dynos, es decir, contenedores inteligentes en un entorno de ejecución fiable y totalmente gestionado.
 - Los desarrolladores despliegan su código escrito en Node, Ruby, Java, PHP, Python, Go, Scala o Clojure en un sistema de construcción que produce una aplicación lista para su ejecución.



Heroku - Características

- **Heroku Developer Experience (DX):**
 - Es un enfoque centrado en la aplicación para la entrega de software.
 - Los desarrolladores pueden centrarse en la creación y la entrega continua de aplicaciones, sin distraerse con los servidores o la infraestructura.
 - Los desarrolladores despliegan directamente desde herramientas populares como Git, GitHub o sistemas de integración continua (CI).



Heroku - Características

- **Data Services and Ecosystem:**

- Heroku Elements permite a los desarrolladores ampliar sus aplicaciones con Add-ons, personalizar su pila de aplicaciones con Buildpacks y poner en marcha sus proyectos con Buttons.
- Los complementos son servicios en la nube de terceros que los desarrolladores pueden utilizar para ampliar inmediatamente sus aplicaciones con una serie de funcionalidades como almacenes de datos, registros, monitorización y mucho más.



Heroku - Características

- **Heroku Operational Experience (OpEx):**
 - Ayuda a los desarrolladores a través de la solución de problemas y la remediación de problemas comunes y la personalización de su experiencia operativa para identificar y abordar rápidamente las tendencias negativas en la salud de su aplicación.
 - Heroku proporciona un conjunto de herramientas para alertarle si algo va mal, o para escalar automáticamente su red dynos si el tiempo de respuesta de las peticiones web supera un umbral que usted especifica.



El futuro digital
es de todos





MinTIC



Vigilada Mineducación

Heroku - Pricing

Heroku ofrece distintos planes:

 <p>Free and Hobby \$0 and up per month</p> <p>Non-commercial apps, such as proof of concepts, MVPs, and personal projects.</p> <p>Estimate</p>	 <p>Production \$25 and up per month</p> <p>Business-focused apps, such as customer-facing or internal web apps and APIs.</p> <p>Selected</p>	 <p>Advanced \$250 and up per month</p> <p>Mission-critical apps with complex functionality that require high availability, very low latency, and handling a high volume of concurrent requests.</p> <p>Estimate</p>	 <p>Enterprise Contact Sales for custom pricing</p> <p>Apps that meet the control, compliance, and collaboration needs of large scale organizations.</p> <p>View Options</p>
---	---	--	--

Para más información visitar: <https://www.heroku.com/pricing>





Heroku - Registro

Free account

Create apps, connect databases and add-on services, and collaborate on your apps, for free.

Your app platform

A platform for apps, with app management & instant scaling, for development and production.

Deploy now

Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.

First name *

Last name *

Email address *

Company name

Role *

Country *

Primary development language *

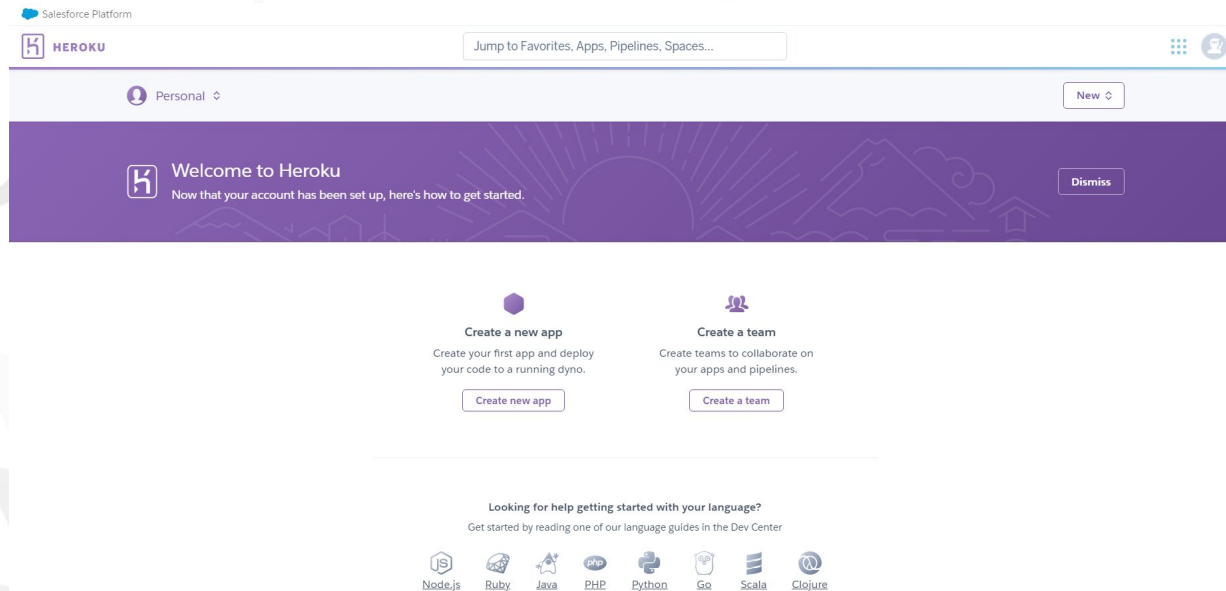
- Registro en Heroku (capa gratuita)
<https://signup.heroku.com/>



El futuro digital
es de todos

MinTIC

Heroku - Dashboard



- En el dashboard de Heroku se pueden crear aplicaciones y equipos.



Heroku - Creación de un proyecto

Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Create New App

App name

app-prueba-mintic

app-prueba-mintic is available

Choose a region

United States

Add to pipeline...

Create app

- Ir a la opción 'Create new App' del dashboard.
- Dar un nombre al proyecto.
- Seleccionar United States como región.
- Hacer click en Create App.



Despliegue de Aplicación Flask en Heroku

- Instalar dependencias necesarias a nuestro proyecto:

```
pip install gunicorn
```

```
pip install python-decouple
```

```
pip freeze > requirements.txt
```



Despliegue de Aplicación Flask en Heroku

- Crear un archivo Procfile. En este fichero se define el proceso que va a ejecutar el dyno de Heroku. Procfile es siempre un archivo de texto simple que se nombra Procfile sin una extensión de archivo. Una vez exista Procfile, se debe añadir la siguiente línea:

```
web: gunicorn main:app
```



Despliegue en Heroku con Git

The screenshot shows the Heroku dashboard. At the top, there's a search bar and a 'Choose a pipeline' dropdown. Below, under 'Deployment method', three options are listed: 'Heroku Git' (Use Heroku CLI), 'GitHub' (Connect to GitHub), and 'Container Registry' (Use Heroku CLI). The 'Connect to GitHub' section is expanded, showing instructions on how to connect the app to a GitHub repository, including viewing code diffs, deploying changes, and enabling automatic deployments. A 'Connect to GitHub' button is visible at the bottom of this section.

- Conectamos la aplicación de Heroku creada a nuestro repositorio de Github.

Desde la pestaña Deploy vamos a escoger la opción: Connect to GitHub y desde GitHub le tenemos que dar permiso a la aplicación Heroku, para que accede a nuestros repositorios.

The screenshot shows a browser window with the GitHub 'Confirm access' dialog. It prompts the user to confirm access for the application 'github.com/login/oauth/authorize'. There is a password field with masked characters, a 'Forgot password?' link, and a green 'Confirm password' button. A tip at the bottom states: 'Tip: You are entering sudo mode. We won't ask for your password again for a few hours.'



Despliegue en Heroku con Git

- Conectar el repositorio donde tenemos nuestra aplicación.

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

josedom24

tienda_videojuegos

Search

Missing an organization? [Ensure Heroku Dashboard has organization access.](#)

josedom24/tienda_videojuegos

Connect

- Escoger la manera de hacer los despliegues: Automáticos o Manual

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. Deploys happen automatically; be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#)

heroku

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more](#)

heroku

Deploy Branch



Despliegue en Heroku con Git

- Acceder a la página desplegada
- Por último indicar que podemos ejecutar un terminal en el dyno para ejecutar cualquier instrucción. Para ello seleccionamos la opción Run console de la opción More y ejecutamos heroku run bash:

Console

```
~ $ python3 manage.py create_admin  
Usuario:█
```



Despliegue en Heroku con CLI

- Descargar e instalar **Heroku CLI**.

🍏 macOS

```
$ brew tap heroku/brew && brew install heroku
```

🪟 Windows

Download the appropriate
installer for your Windows
installation:

64-bit installer

32-bit installer

🐧 Ubuntu 16+

Run the following from your terminal:

```
$ sudo snap install --classic heroku
```

Snap is available on other Linux OS's as well.



Despliegue en Heroku con CLI

- Desde la terminal, iniciar sesión en tu cuenta de Heroku y seguir las indicaciones para crear una nueva clave pública SSH.

```
$ heroku login
```

- Inicializar el repositorio de Git

```
$ cd my-project/  
$ git init  
$ heroku git:remote -a app-prueba-mintic
```



Despliegue en Heroku con CLI

Despliega tu aplicación

- Confirma tu código en el repositorio y desplégalo en Heroku usando Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```



El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

Ejercicios de práctica

Mision
TIC2022



Seguimiento Habilidades Digitales en Programación

* De modo general, ¿Cuál es grado de satisfacción con los siguientes aspectos?

	Nada Satisfecho	Un poco satisfecho	Neutra	Muy satisfecho	Totalmente satisfecho
Sesiones técnicas sincrónicas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sesiones técnicas asincrónicas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sesiones de inglés	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apoyo recibido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Material de apoyo: diapositivas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Material de apoyo: ejercicios prácticos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Completa la siguiente encuesta para darnos retroalimentación sobre esta semana ▼▼▼

<https://www.questionpro.com/t/ALw8TZIxOJ>



El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

¡GRACIAS
POR SER PARTE DE
ESTA EXPERIENCIA
DE APRENDIZAJE!



Misión
TIC 2022