



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

Ciclo III

## Desarrollo de Software



### Capa de Presentación: Despliegue

15

**Jeisson Andrés Vergara Vargas**

Departamento de Ingeniería de Sistemas e Industrial

<http://colswu.unal.edu.co/~javergarav/>

[javergarav@unal.edu.co](mailto:javergarav@unal.edu.co)

2020

©

# Objetivo de Aprendizaje

**Desplegar** un componente de presentación en una *PaaS*.

# Despliegue

## Repositorios

1. Actualizar el repositorio **cajero-api**, con la última versión del proyecto.
2. En **GitHub**, crear un repositorio llamado **cajero-app**.
3. **Clonar** el repositorio de manera local.
4. Ingresar a la raíz del directorio del repositorio clonado (**client-app/**)
5. Añadir los archivos correspondientes al proyecto de Vue desarrollado en las sesiones anteriores.

## Recursos de Despliegue

En la raíz del repositorio **cajero-app/**, crear un archivo llamado **server.js**. Añadir el siguiente contenido:

```
const express = require('express')
const serveStatic = require('serve-static')
const path = require('path')

const app = express()

//here we are configuring dist to serve app files
app.use('/', serveStatic(path.join(__dirname, '/dist')))

// this * route is to serve project on different page routes except root `/'
app.get(/.*/, function (req, res) {
  res.sendFile(path.join(__dirname, '/dist/index.html'))
})

const port = process.env.PORT || 8080
app.listen(port)
console.log(`app is listening on port: ${port}`)
```

## Recursos de Despliegue

- Ejecutar el siguiente comando (instalación de servidor *Express*):

```
npm install express serve-static --save
```

- Modificar el archivo **package.json**:

```
"scripts": {  
  "dev": "webpack-dev-server --inline --progress --config build/webpack.dev.conf.js",  
  "start": "node server.js",  
  "build": "node build/build.js"  
}
```

## Recursos de Despliegue

- En la raíz del proyecto, crear un directorio llamado **/dist**.
- Ejecutar el siguiente comando:

```
npm run build
```

- Como verificación previa, desplegar el componente localmente:

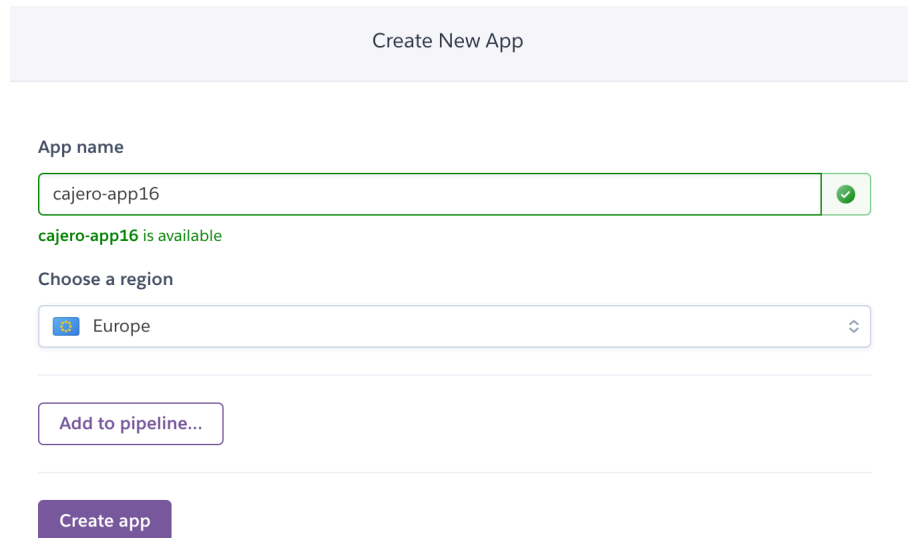
```
node server.js
```

- Subir los cambios al repositorio de GitHub.

# Heroku

1. Iniciar sesión en Heroku
2. Crear una nueva aplicación, cuyo nombre incluya «**cajero-app**»:

**Ejemplo:** cajero-app16

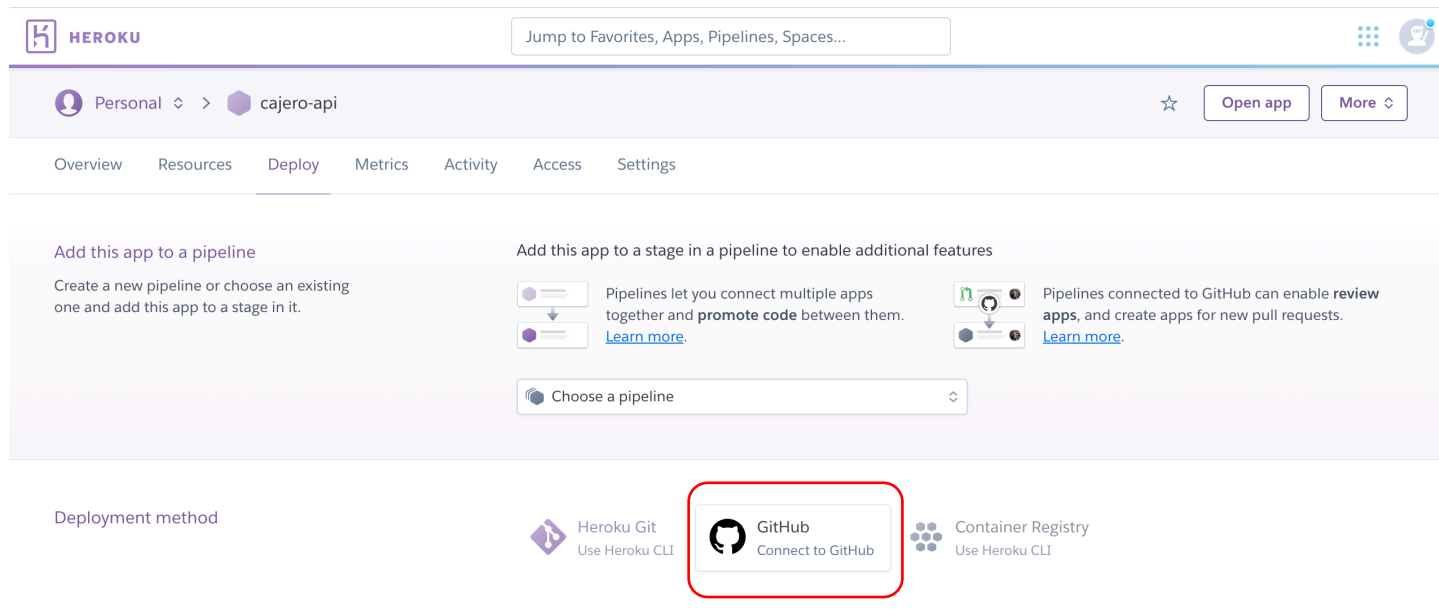


The screenshot shows the 'Create New App' interface on Heroku. At the top is a light purple button labeled 'Create New App'. Below it, the 'App name' field contains 'cajero-app16' and is marked with a green checkmark. A green message below the field states 'cajero-app16 is available'. The 'Choose a region' dropdown menu is set to 'Europe'. Below these fields is a purple button labeled 'Add to pipeline...'. At the bottom is a dark purple button labeled 'Create app'.



# Heroku

## Enlazar cuenta de GitHub:



The screenshot shows the Heroku dashboard for the 'cajero-api' app. The 'Deploy' tab is selected, and the 'Add this app to a pipeline' section is visible. Below this, the 'Deployment method' section shows three options: Heroku Git, GitHub, and Container Registry. The GitHub option is highlighted with a red box.

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

Personal > cajero-api Open app More

Overview Resources **Deploy** Metrics Activity Access Settings

**Add this app to a pipeline**  
Create a new pipeline or choose an existing one and add this app to a stage in it.

**Add this app to a stage in a pipeline to enable additional features**

Pipelines let you connect multiple apps together and **promote code** between them. [Learn more.](#)

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more.](#)

Choose a pipeline

**Deployment method**

- Heroku Git  
Use Heroku CLI
- GitHub**  
Connect to GitHub
- Container Registry  
Use Heroku CLI

# Heroku

Seleccionar el **repositorio** correspondiente:

## Deployment method



Heroku Git  
Use Heroku CLI



GitHub  
Connect to GitHub



Container Registry  
Use Heroku CLI

## Connect to GitHub

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

Search for a repository to connect to

 javergarav

repo-name

Search

Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)

 javergarav/cajero-app

Connect

# Heroku

## Desplegar manualmente:



### 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](#).

Choose a branch to deploy

 main 

Deploy Branch

# Heroku

## Desplegar manualmente:

### 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](#).

Choose a branch to deploy

 main

Deploy Branch

Receive code from GitHub



Build main 892c2cec



Release phase



Deploy to Heroku



Your app was successfully deployed.

 View

## Ajustes de Conexión entre Componentes

En el **componente lógico**: (**cajero-api/**)

Añadir la URI del componente de presentación desplegado en Heroku, como política CORS: archivo **main.py** (**origins = []**):

```
origins = [  
    "http://localhost.tiangolo.com",  
    "https://localhost.tiangolo.com",  
    "http://localhost",  
    "http://localhost:8080",  
    "https://cajero-app16.herokuapp.com"  
]
```

**Nota:** verificar el nombre del *host* proporcionado por Heroku (<https://cajero-app16.herokuapp.com/>). Este nombre dependerá del nombre creado para la aplicación.

## Ajustes de Conexión entre Componentes

En el **componente de presentación**: (**cajero-app/**)

Modificar las URIs asociadas al componente lógico (**UserBalance.vue**): ahora deben apuntar a la dirección del componente lógico desplegado en Heroku:

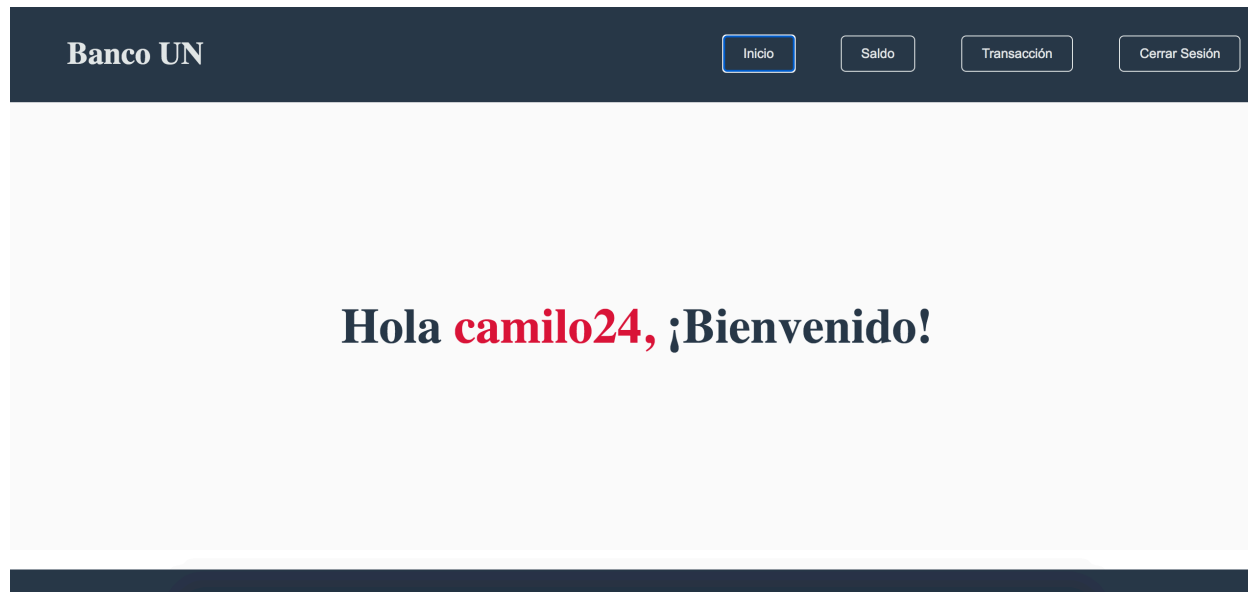
```
axios.get("https://cajero-api.herokuapp.com/user/balance/" + this.username)
```

**Tras estas acciones:** 1) actualizar los repositorios, 2) desplegar de nuevo los dos componentes en Heroku.

# Prueba del Componente

## Probando el Componente

Ingresa a la aplicación web mediante un navegador y probar la funcionalidad:



**Nota:** verificar el nombre del *host* proporcionado por Heroku (<https://cajero-app16.herokuapp.com/>). Este nombre dependerá del nombre creado para la aplicación.