Heroku

Web: https://www.heroku.com/

Heroku es una plataforma de computación en la nube que soporta diferentes tipos de lenguajes de programación y permite desplegar, alojar, supervisar y escalar las aplicaciones.

Características:

Permite el uso de diferentes lenguajes de programación

Integra diferentes servicios

Despliegue de una aplicación React

Creamos nuestra aplicación React:

Nos dirigimos al directorio donde queremos crear nuestra aplicación.

Creamos nuestra app React con: create-react-app componente-simple

Vamos a la ruta de la aplicación para ver que nos la ha creado correctamente.
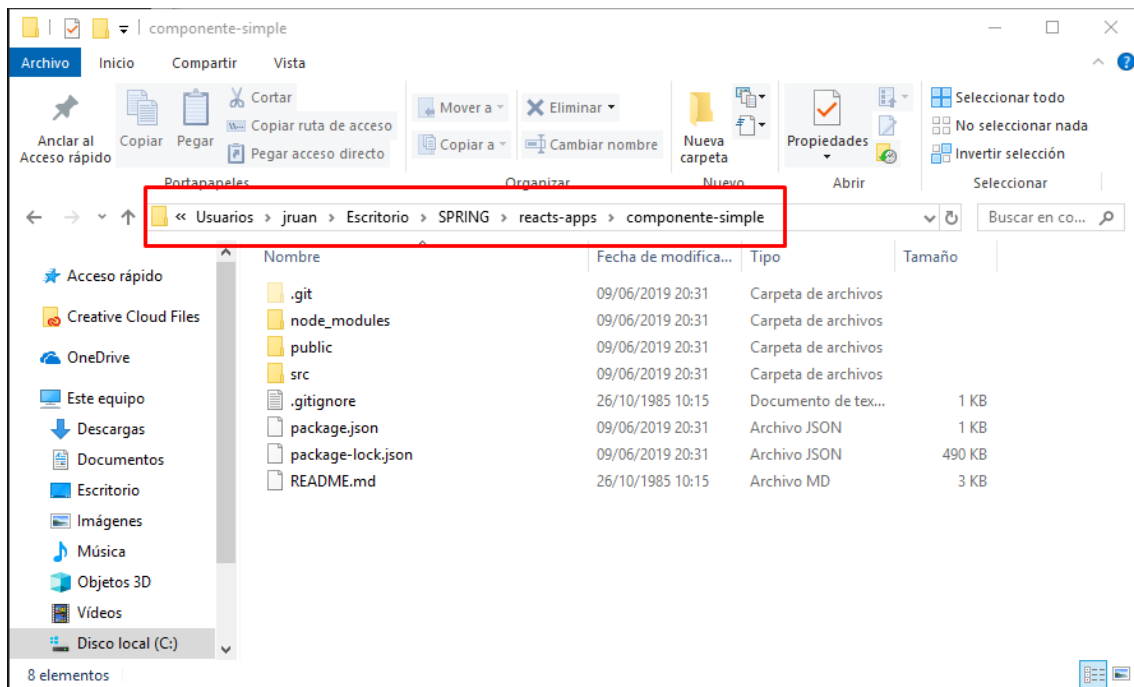
Modificamos el fichero index.js

```javascript
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';

class HolaMundo extends React.Component {
  render() {
    return (
      <div>
        Hola {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HolaMundo name="Mundo" />,
  document.querySelector('#root')
);
```

Descargar e instalar git:

https://git-scm.com/download/win

Downloading Git

Your download is starting...

You are downloading the latest (**2.22.0**) **64-bit** version of **Git for Windows**. This is the most recent maintained build. It was released **1 day ago**, on 2019-06-08.

If your download hasn't started, **click here to download manually**.

Other Git for Windows downloads

Git for Windows Setup
32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Git for Windows Portable ("thumbdrive edition")
32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

The current source code release is version **2.22.0**. If you want the newer version, you can build it from the source code.

Establecemos nuestro nombre de usuario y email de Git

git config --global user.name "jruano****"

git config --global user.email jruano***@****.com

Descargar e instalar Heroku CLI

https://devcenter.heroku.com/articles/heroku-cli

Download and install

⚠ The Heroku CLI requires **Git**, the popular version control system. If you don't already have Git installed, complete the following before installing the CLI:

- Git installation
- First-time Git setup

macOS

Download the installer

Also available via Homebrew:

```
$ 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.

Verificar que está instalado heroku --version

```
C:\>heroku --version
heroku/7.25.0 win32-x64 node-v11.14.0
```

Iniciar sesión en heroku heroku login

```
C:\>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/browser/92c193db-e2c8-4c44-ab6a-023f64bb57e1
heroku: Waiting for login... /
```

Nos abre una página para que inicies sesión.

Nos solicita que cierres la página que se ha abierto anteriormente.

E iniciamos nuestra sesión.

Nos dirigimos al directorio donde esta nuestra aplicación para crear un repositorio git:

git init

Añadimos los ficheros del directorio con:

git add .

Comiteamos la aplicación:

Git commit -m "initial commit"

Creamos la aplicación de heroku, este nombre debe de ser único y como máximo tiene que tener 30 caracteres.

heroku create hello-world-app-react-jruano

Como podemos ver en nuestro panel de administración de heroku la ha creado

Desplegamos nuestra aplicación con:

git push heroku master

Nos dice que ha ido bien.

Si vamos a nuestra dirección para ver que tenemos desplegada la aplicación.

https://hello-world-app-react-jruano.herokuapp.com/
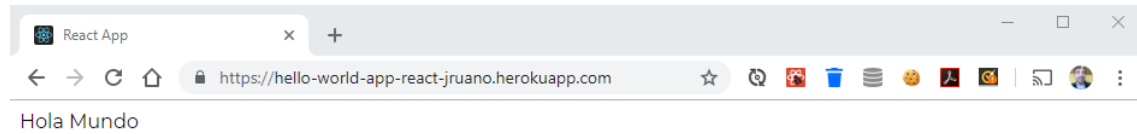
En nuestro panel de administración si vamos a Settings.
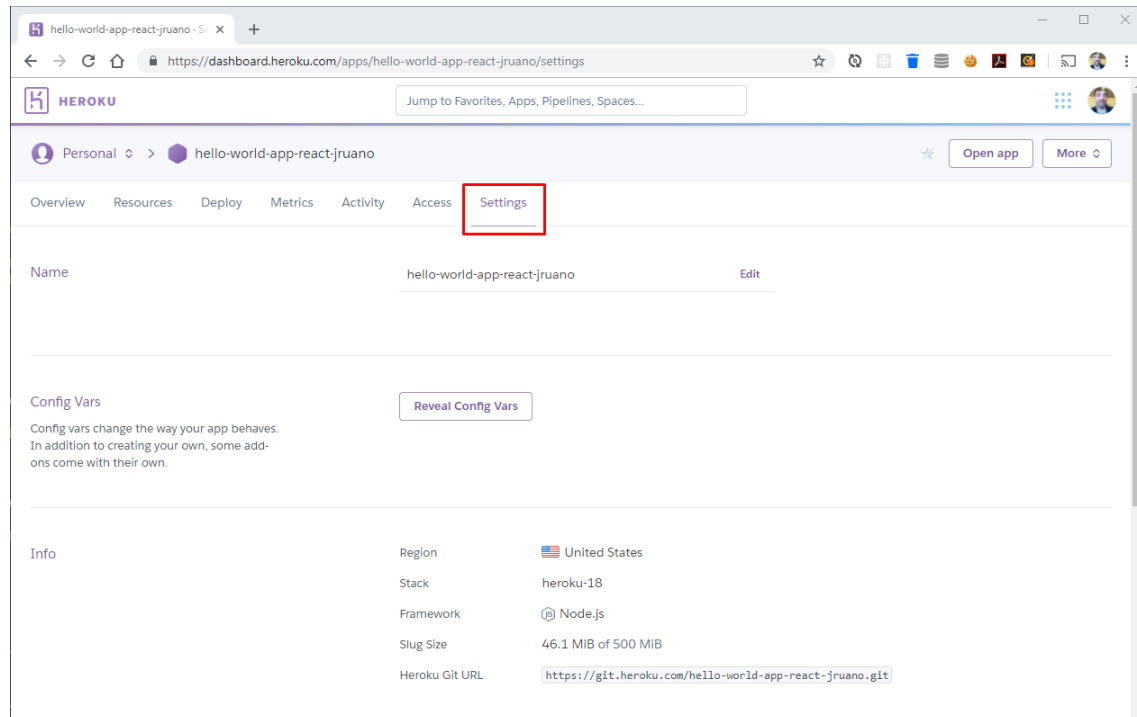
Podemos ver el dominio de nuestra aplicación.
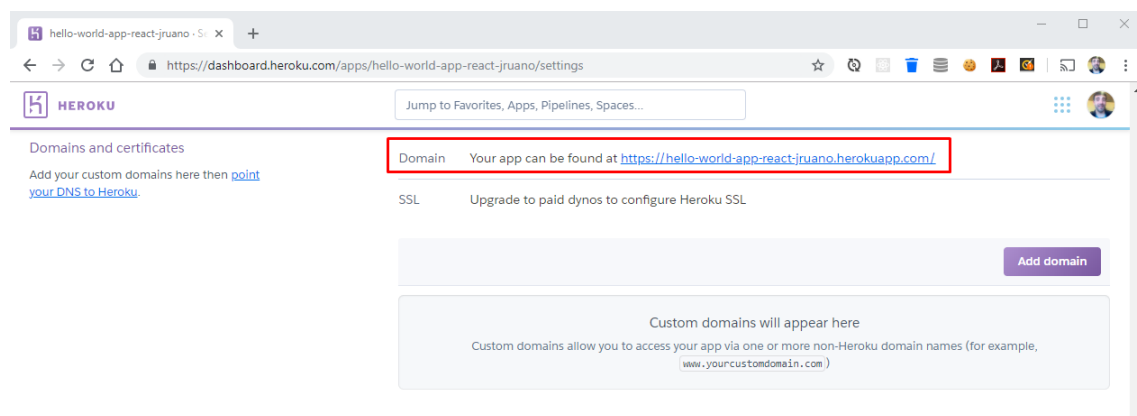
Despliegue de una aplicación Rest en Java con Spring boot y mongodb
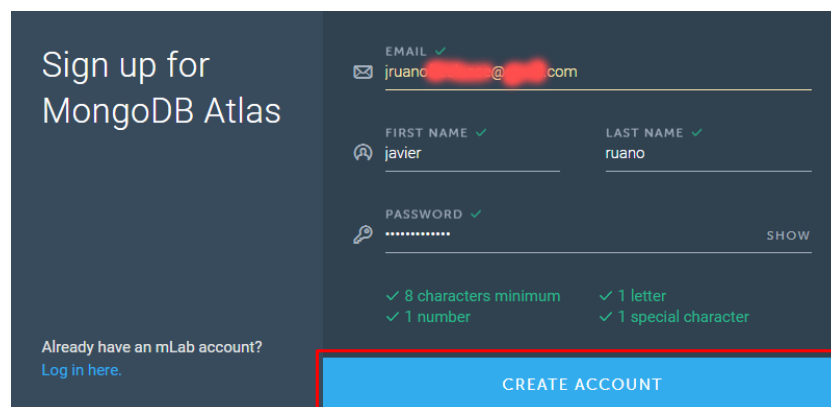
Creamos nuestra base de datos en mlab:

https://mlab.com/

Accedemos a la web y hacemos click para registrarnos.
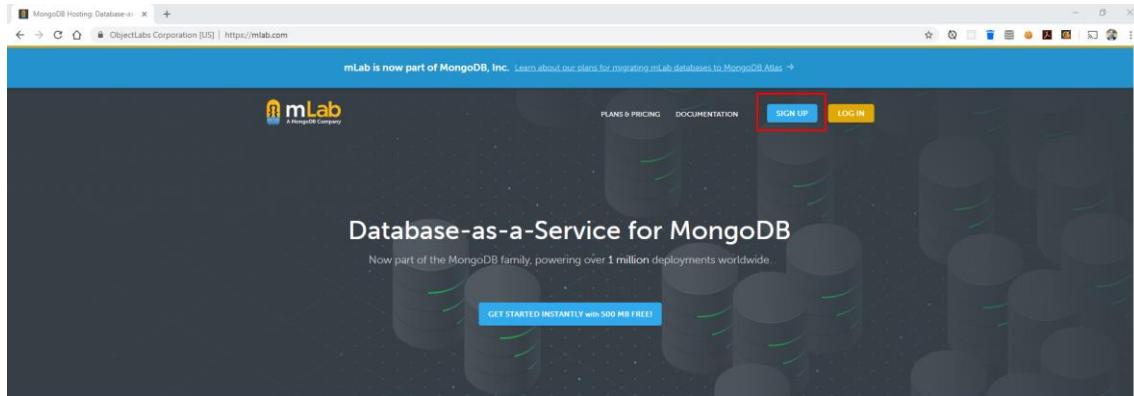
Completamos el formulario y hacemos click en create account.
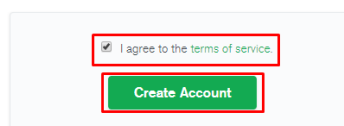
Aceptamos los términos del servicio.

Podemos cerrar el siguiente mensaje para que nos ofrezcan ayuda o proseguir con una pequeña ayuda, hacemos click en Build my first cluster.
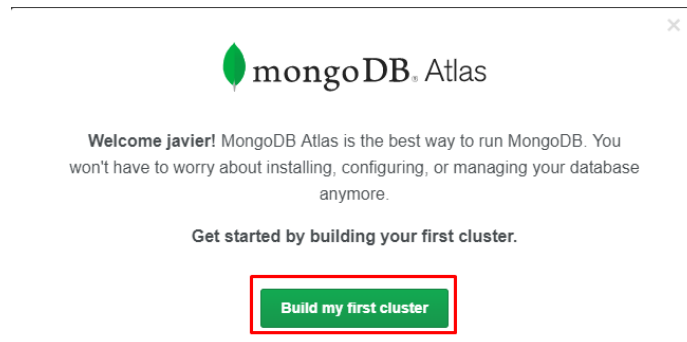
Elegimos la configuración global, en nuestro caso gratuita y desde Europa y hacemos click en Create Cluster.

Nos dirigirá a nuestra pantalla de administración donde nos informa de que debemos esperar unos minutos a que se cree el Cluster.
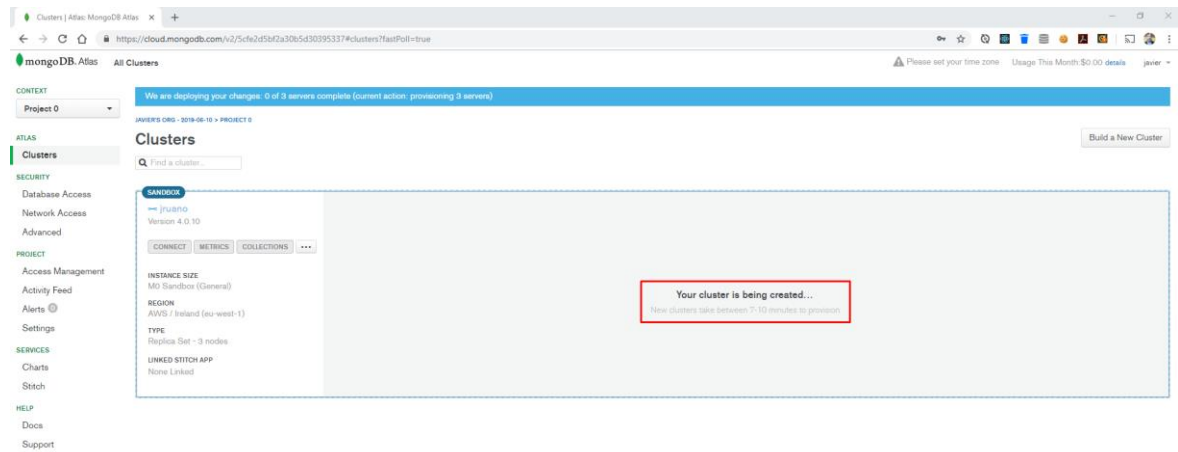
Una vez ya se haya creado podemos empezar por introducir nuestra zona horaria.
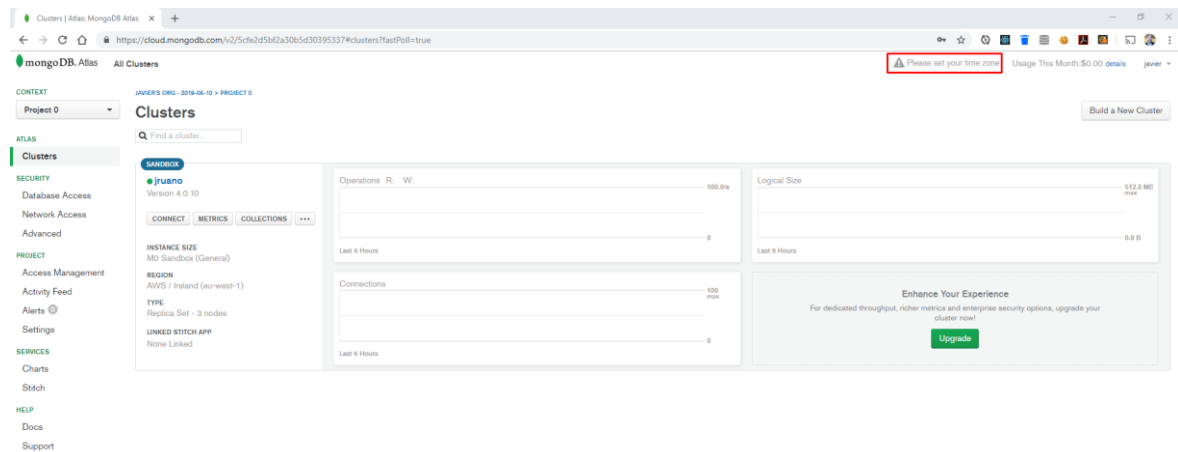
Seleccionamos parís y hacemos click en Set.

Hacemos click en Database Access para configurar nuestra base de datos en mongo.
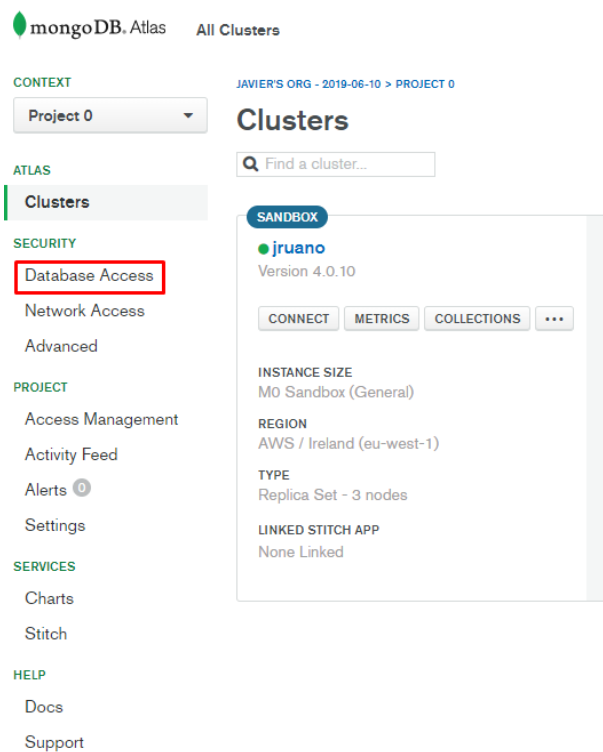
Primeramente, añadimos nuestro usuario para ello hacemos click en Add new user.
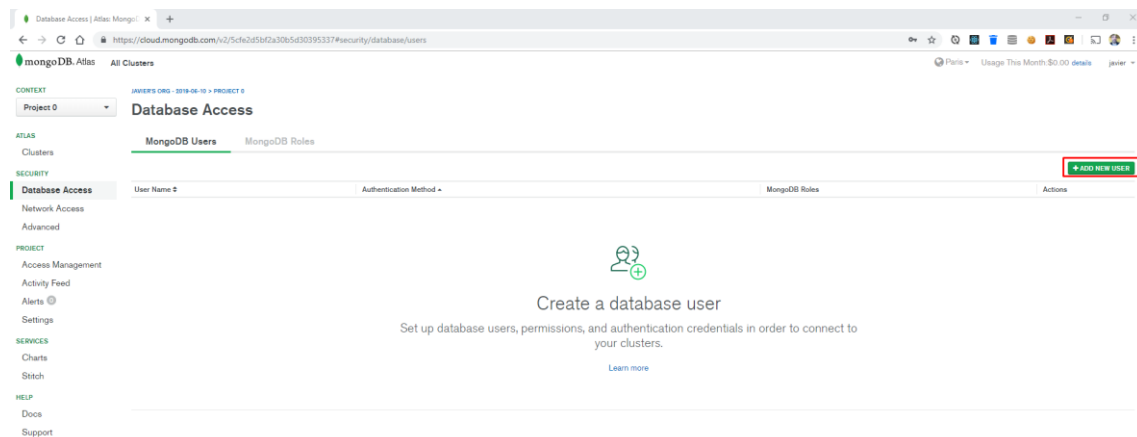
Completamos el formulario de creación del usuario y hacemos click en Add User.

Y como vemos se nos ha creado nuestro usuario.
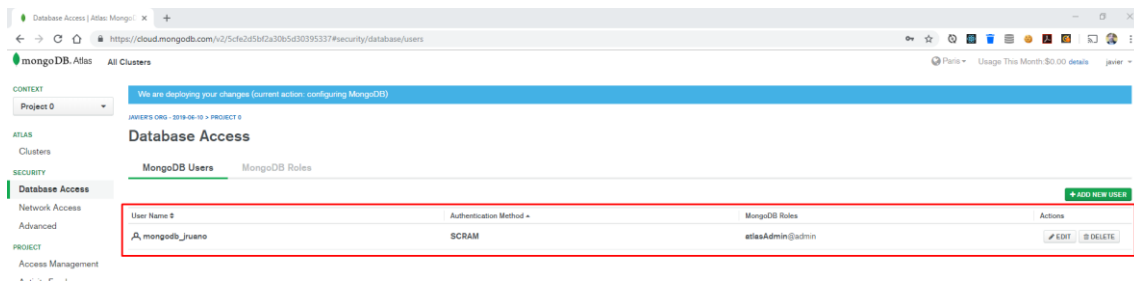
Ahora debemos de configurar las direcciones IPs que tendrán acceso a nuestra base de datos haciendo click en Network Access y después en Add ip address.
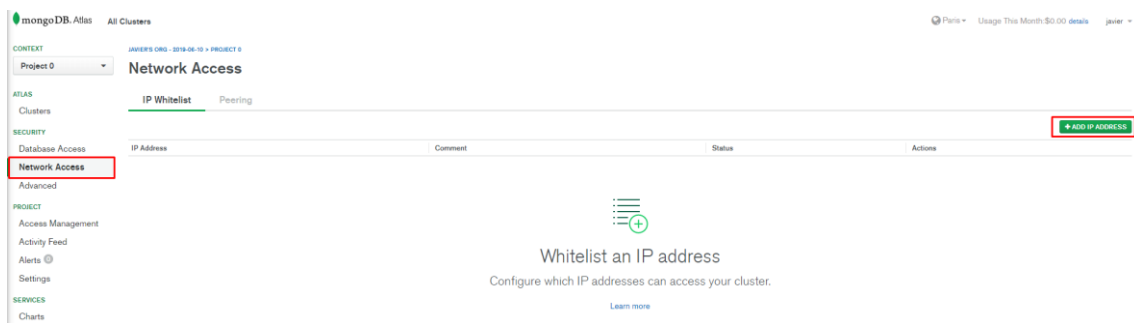
Al ser una prueba haremos click en que cualquier dirección se pueda conectar, si fuera un entorno real añadiríamos la dirección o direcciones que quisiéramos que se conectaran a la bd por lo que hacemos click en Allow Access from anywhere y click en Confirm.

Y como vemos se añade a nuestro panel.

Ahora volveríamos a Clusters y haríamos click en connect.
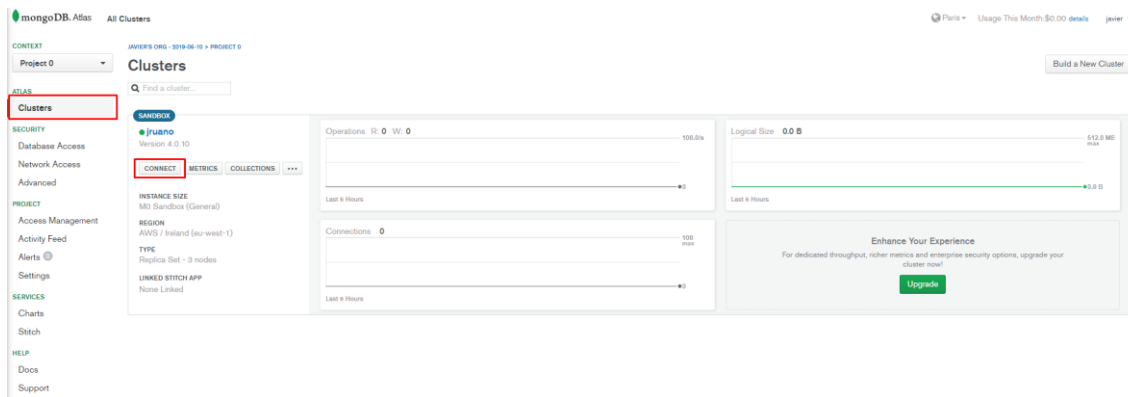
Y nos aparecen tres opciones, usar la consola, conectar la aplicación o usar un cliente gráfico (MongoDB Compass), seleccionamos conectar nuestra aplicación haciendo click sobre Connect Your Application.
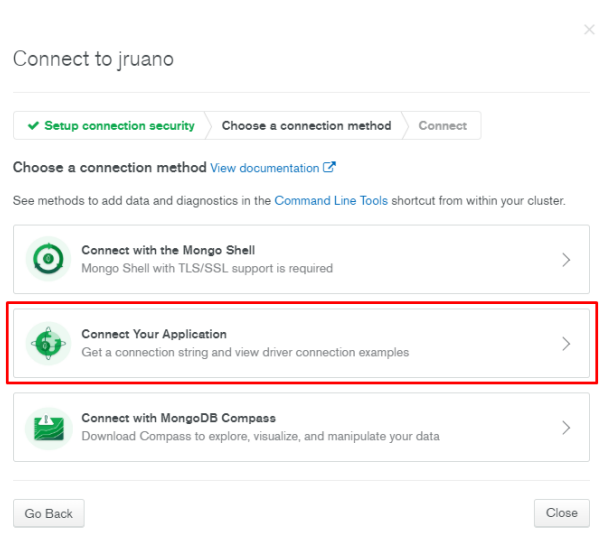
Copiamos nuestra conexión haciendo click sobre Copy y cerramos en Close.
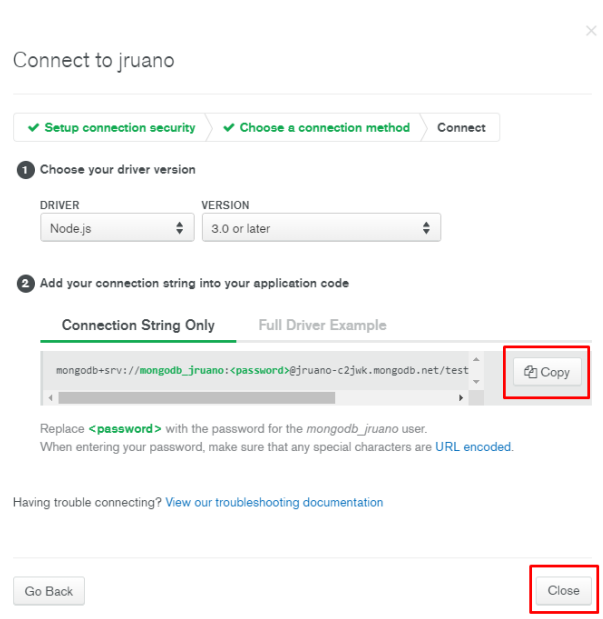
Creamos nuestro Servicio REST con Spring Boot:

Nos dirigimos a la web de spring initializr

https://start.spring.io/

Nombramos la paquetería, añadimos las dependencias Spring Web Starter, Spring Data MongoDB y hacemos click en Generate the project.
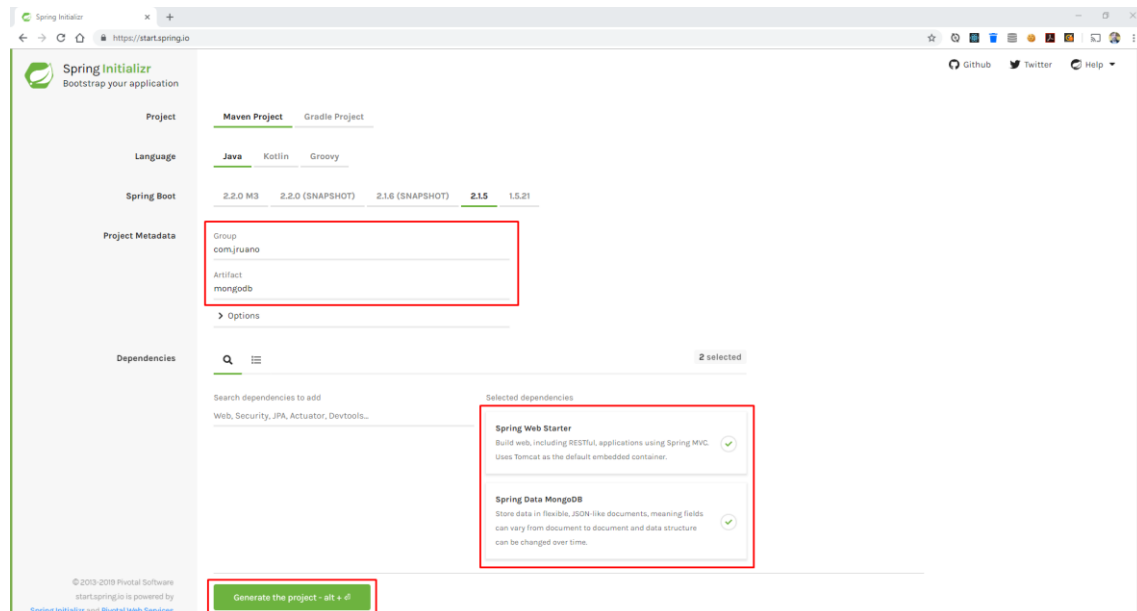
Desarrollamos nuestra aplicación, una vez la tengamos acabada procedemos a añadir la conexión a nuestra bd de mongo, lo que ya podemos hacer es ir a nuestra aplicación en Spring boot y añadir la conexión a nuestro fichero de properties o yaml.

mongodb+srv://mongodb_jruano:<password>@jruano-c2jwk.mongodb.net/test?retryWrites=true&w=majority
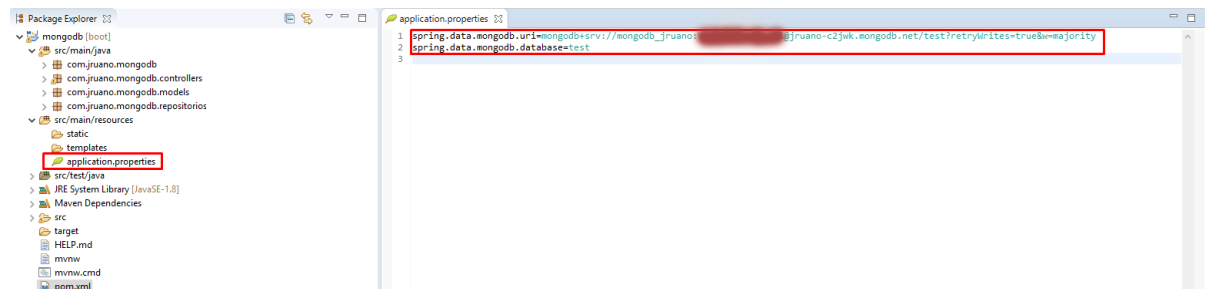
Hacemos maven clean mediante click botón derecho sobre nuestro proyecto, vamos a Run as y click en Maven clean.
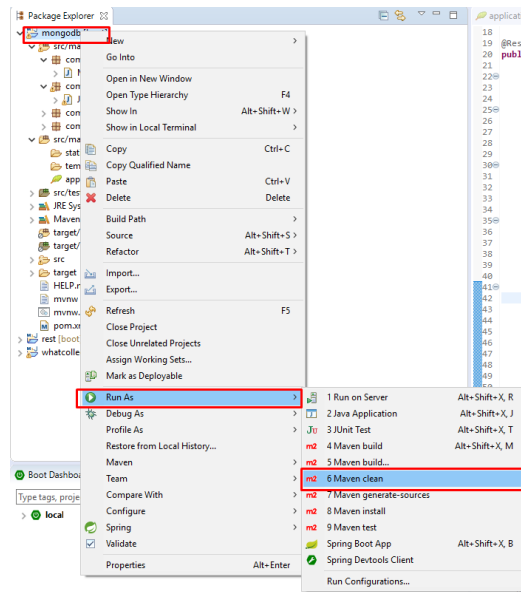
Vemos que ha acabado correctamente.
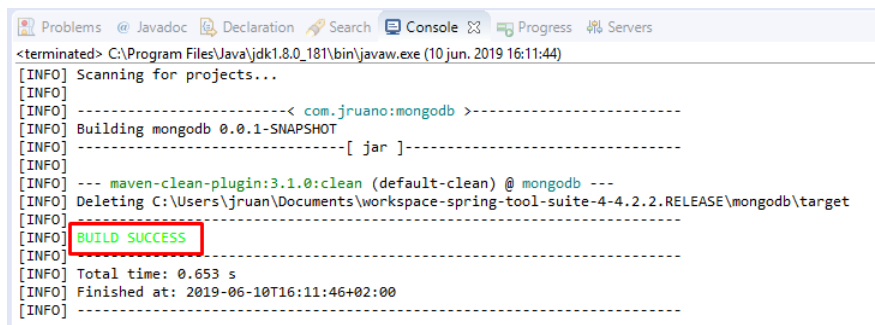
Hacemos Update Project, haciendo click botón derecho sobre nuestro proyecto, vamos a Maven y click en Update Project.
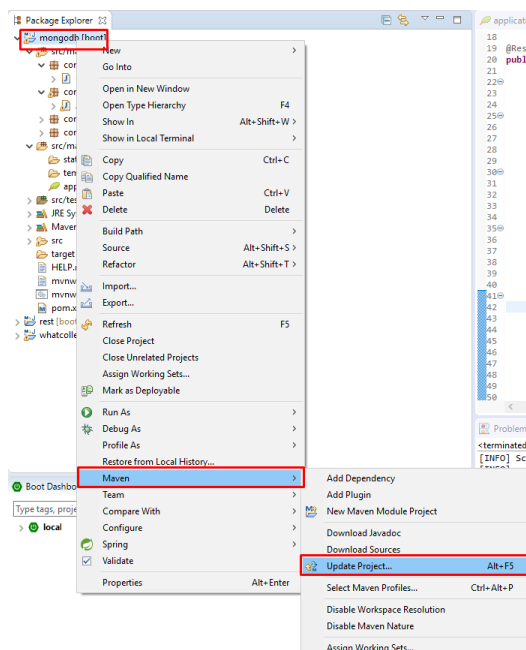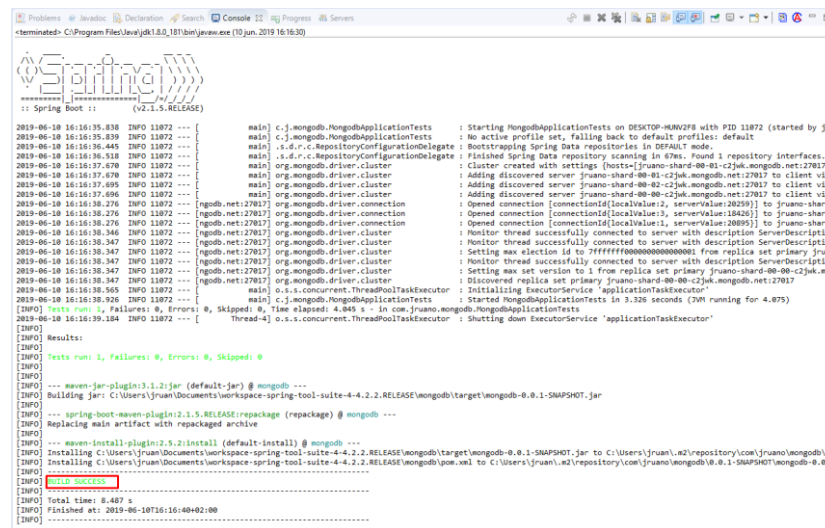
Seleccionamos nuestro proyecto y marcamos Force Update of Snapshots/Releases y click en
OK
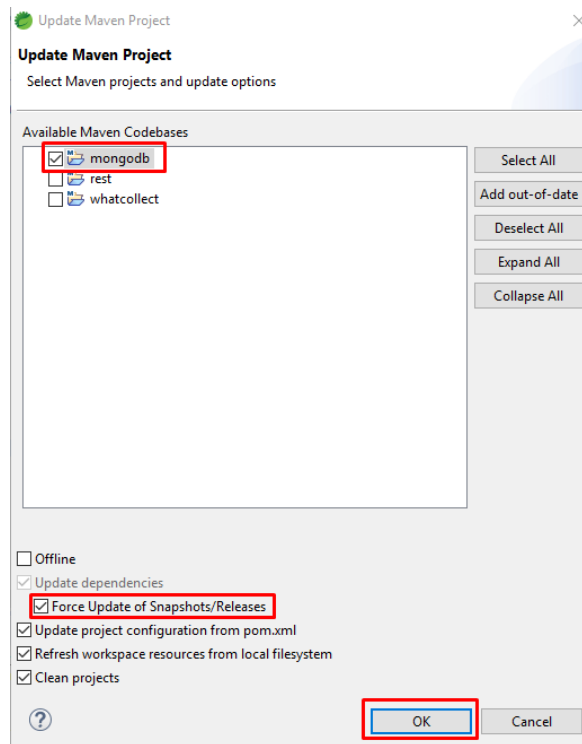
Realizamos el maven install para generar nuestro .jar para ello hacemos click derecho sobre
nuestro proyecto, vamos a Ruan As y click en Maven install.
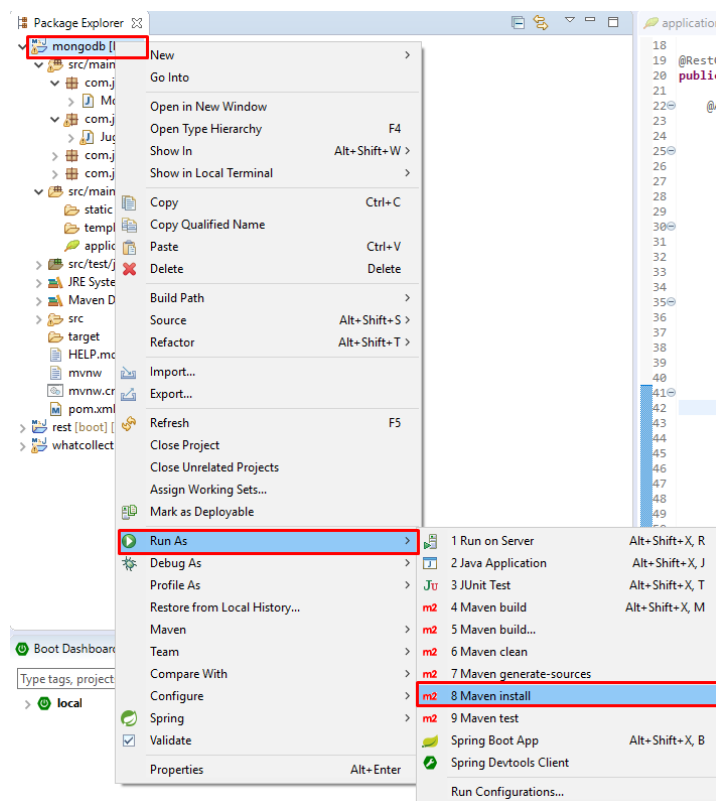
Vemos que todo ha ido bien.

Seguidamente arrancamos nuestra aplicación spring boot en local para probar que la conexión funciona correctamente, haciendo click derecho sobre el proyecto, y seleccionando Run As, click en Spring Boot App.

Vemos que arranca y conecta correctamente.

Iniciamos Soap UI para probar nuestra aplicación REST está conectada a la bd de mongo en remoto, hacemos click sobre REST.

Añadimos nuestra dirección http://localhost:8080 y hacemos click en OK.

Realizamos una inserción en bd a través de POST sobre nuestra api REST, y como vemos en el recuadro verde se ha insertado correctamente.

Ahora igualmente llamamos a la petición que nos devuelve todos los jugadores a través de GET y como vemos nos devuelve el jugador insertado.
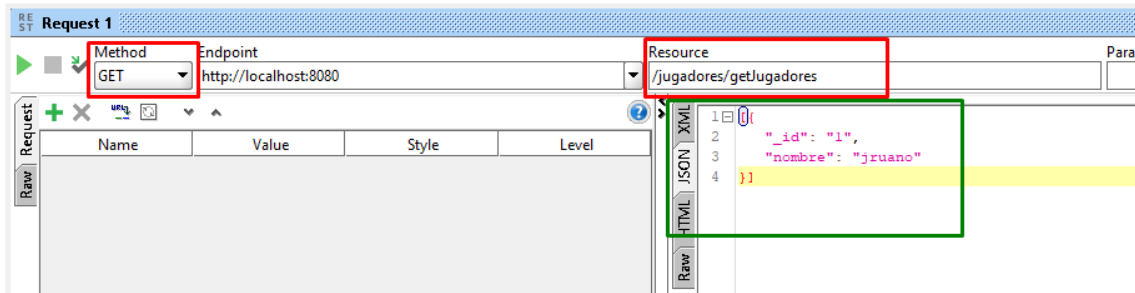
Igualmente, desde el navegador podemos ver el resultado
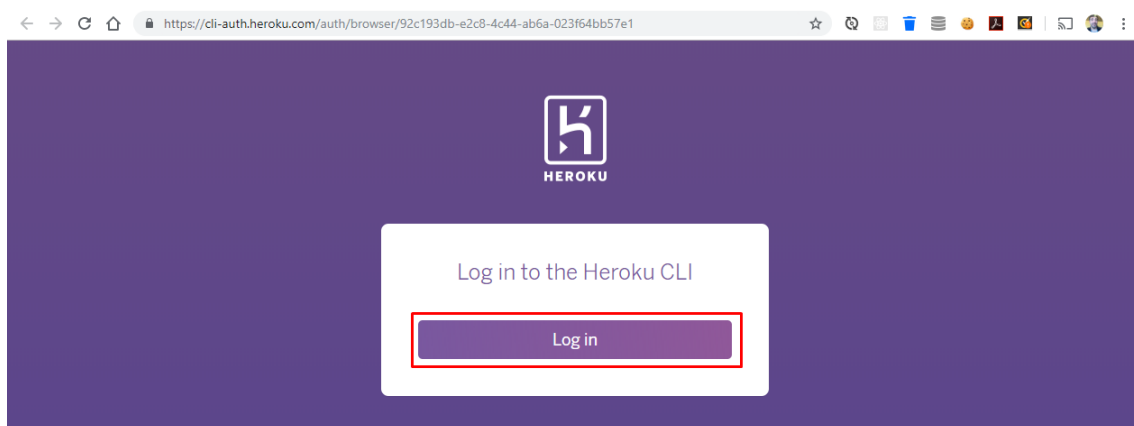

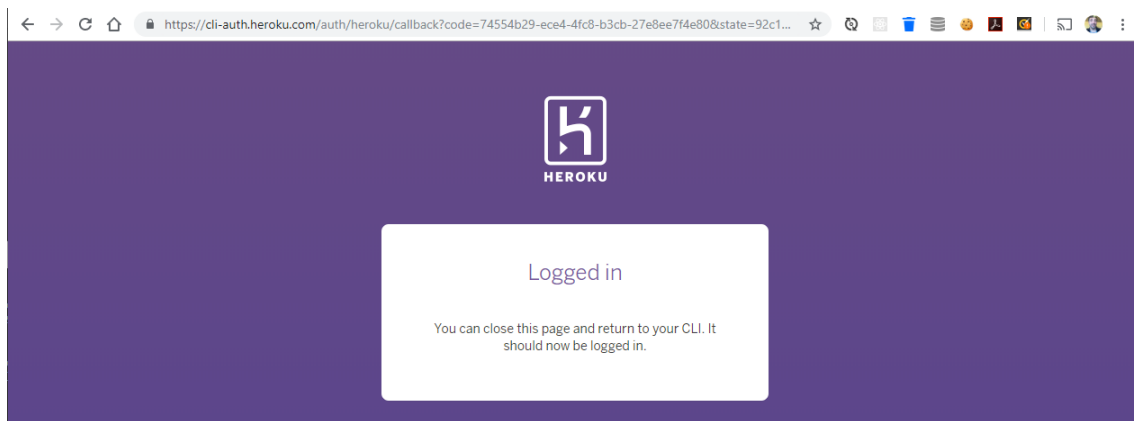Ahora procedemos al despliegue para ello iniciamos sesión en heroku con:

heroku login

Nos abre una página para que inicies sesión.

Nos solicita que cierres la página que se ha abierto anteriormente.

E iniciamos nuestra sesión.

Nos dirigimos al directorio donde esta nuestra aplicación para crear un repositorio git:

git init

Añadimos los ficheros del directorio con:

git add .

Comiteamos la aplicación:

Git commit -m "initial commit"

Creamos la aplicación de heroku, este nombre debe de ser único y como máximo tiene que tener 30 caracteres.

heroku create api-rest-prueba-jruano

Como podemos ver en nuestro panel de administración de heroku la ha creado.

Desplegamos nuestra aplicación con:

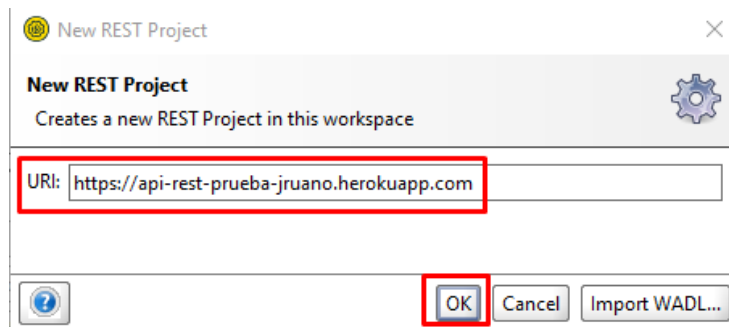git push heroku master

Nos dice que ha ido bien.

Abrimos Soap UI, y creamos una nueva conexión REST.

Realizamos una inserción en bd a través de POST sobre nuestra api REST, y como vemos en el recuadro verde se ha insertado correctamente.
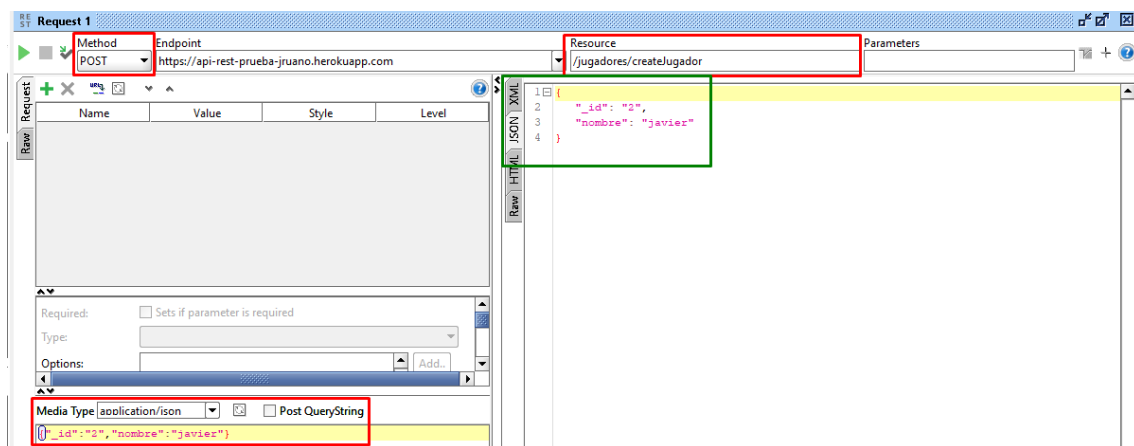
Lanzamos una petición GET en la web para ver que funciona correctamente.

https://api-rest-prueba-jruano.herokuapp.com/jugadores/getJugadores