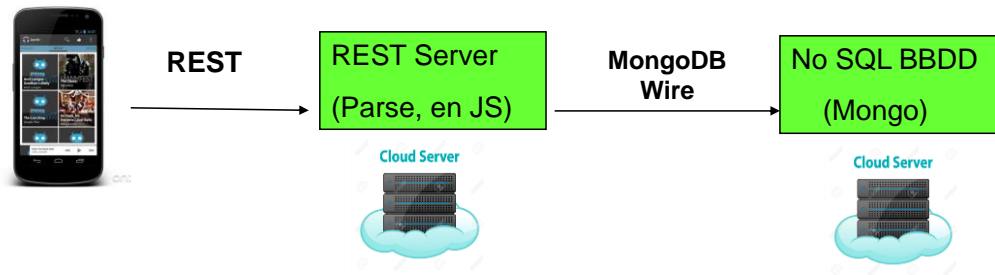


### Práctica 3: Instalación de Servidor REST

Vamos a instalar un servidor de tipo REST, y una base de datos No-SQL. En la práctica siguiente conectaremos Android con el servidor REST. En la siguiente figura se muestra la arquitectura que tendrá en sistema, y que empleareís en el proyecto.



El código del servidor REST lo creo una empresa y después lo ha liberado <http://parseplatform.org/>

Lo instalaremos en la “nube” a través de la empresa Heroku <https://www.heroku.com/>

Como base de datos No-SQL usaremos la empresa MongoLab <https://www.mongodb.com/>

Ambas empresas Heroku y Mongolab utilizan el hardware de la empresa Amazon, en sus servidores de la nube Amazon Web Services. (es importante que usemos servidores que estén físicamente en el mismo lugar o cercanos para que la conexión sea eficiente).

Vamos a tener que abrir una cuenta en los tres sitios, utiliza un username y password que puedas recordar (pej. el mismo en los tres, y que sea sencillo, 1234 nos vale, son solo prácticas).

### Clonar código Parse-server (Servicio de GITHUB)

Crea una cuenta en <https://github.com>

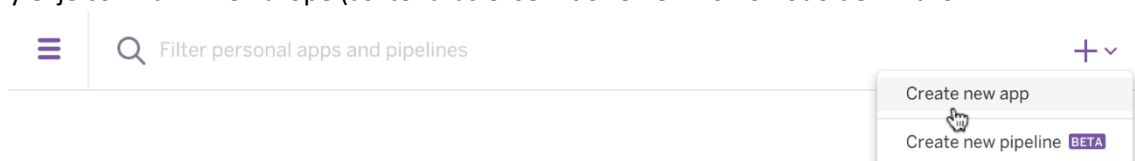
Accede a la página <https://github.com/parse-community/parse-server-example>

Haz un fork del código (botón en la parte superior derecha de la página): de esta manera se hace una copia del código en tu cuenta de github que solo tú puedes modificar y acceder.

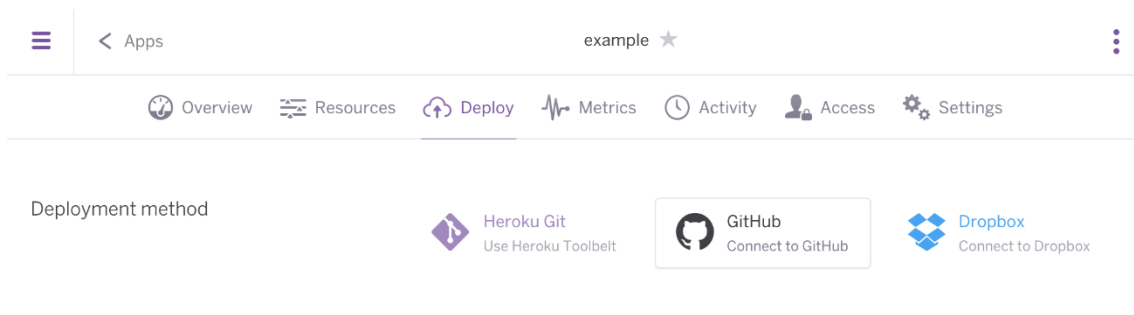
### Desplegar Parse-Server (en Servicio de HEROKU)

Crea una cuenta en <https://heroku.com>

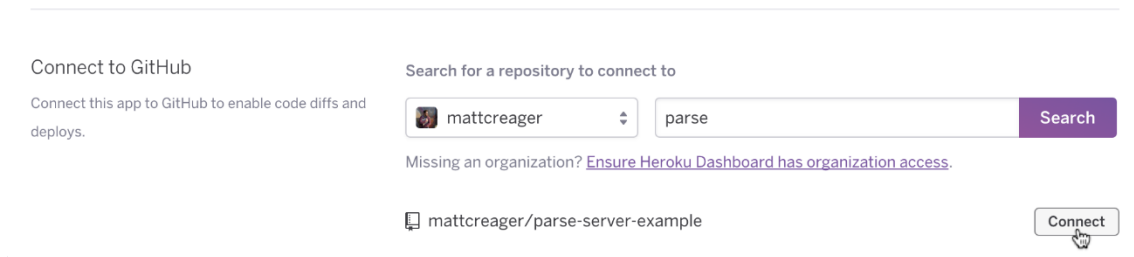
Crea una app, usa el menú del botón superior derecho: Ponle un nombre único, pej. *miServidor* y elige con RunTime Europe (asi tendrás el servidor en el mismo nodo de Amazon).



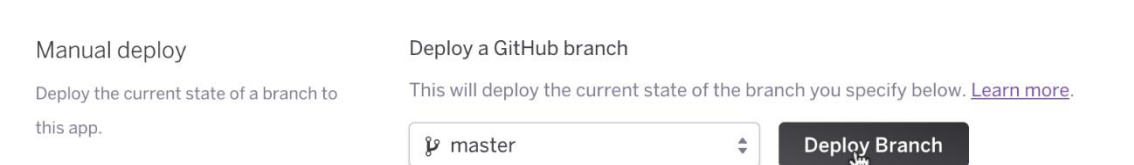
Elije como método de despliegue (deployment) “GitHub”: tendrás que logearte en GitHub y permitir que Heroku pueda acceder a tus ficheros en GitHub.



A continuación conectarás la app al repositorio de GitHub que habíamos clonado antes:



A continuación realizaremos un despliegue manual del código en gitHub a Heroku:



Si todo va bien, puedes entrar en el navegador y acceder desde un navegador a <https://miServidor.herokuapp.com/> sustituyendo *miServidor* con el nombre que elegiste antes: veras una página con el texto: **I dream of being a website**".

## Uso del servidor API Parse Server (I)

Ahora ya podemos acceder al servidor API de varias maneras: por línea de comandos haciendo peticiones HTTP con el programa CURL, desde Javascript, desde Android, y desde otras plataformas haciendo uso de las librerías correspondientes, las cuales internamente hacen las peticiones HTTP que se puede ver con CURL.

Ejecutando este comando, nos conectamos con el servidor `miservidor.herokuapp.com` (cámbialo por el nombre de tu app en Heroku) y le enviamos una petición POST con los datos `{}` en formato JSON. Se ejecutando la función remota *hello* que nos devuelve un JSON.

```
curl -X POST -H "X-Parse-Application-Id: myAppId" -H "Content-Type: application/json" -d '{} 'https://miServidor.herokuapp.com/parse/functions/hello
```

Esta llamada remota tiene que devolver:

```
{"result":"Hi"}
```

En este momento solo puedes acceder a las funciones remotas del servidor sin persistencia de datos, vamos a crear un BBDD en MongoLab y conectar el servidor para poder guardar datos.

## crear BBDD No-SQL (Servicio de MONGODB)

Crea una cuenta en <https://www.mongodb.com/>

Crea un cluster gratuito (se crea un cluster de 3 servidores donde se replicará una base de datos No-SQL de tipo MongoDB).




Elige opciones gratuitas: Región: Frankfurt; Cluster Tier: M; Ponle en nombre que quieras.

### Create New Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our [documentation](#).
















Global Cluster Configuration >

Cloud Provider & Region AWS, Frankfurt (eu-central-1) v

Create a **free tier cluster** by selecting a region with **FREE TIER AVAILABLE** and choosing the **M0** cluster tier below.

★ recommended region ⓘ

| NORTH AMERICA   | EUROPE   | AUSTRALIA  |
|---|--|--|
|  N. Virginia (us-east-1) ★<br><b>FREE TIER AVAILABLE</b> |  Stockholm (eu-north-1) ★                                   |  Sydney (ap-southeast-2) ★                                    |
|  Ohio (us-east-2) ★                                    |  Ireland (eu-west-1) ★                                    | <b>ASIA</b>  |
|  N. California (us-west-1)                             |  London (eu-west-2) ★                                     |  Tokyo (ap-northeast-1) ★                                   |
|  Oregon (us-west-2) ★                                  |  Paris (eu-west-3) ★                                      |  Seoul (ap-northeast-2)                                     |
|  Montreal (ca-central-1)                               |  Frankfurt (eu-central-1) ★<br><b>FREE TIER AVAILABLE</b> |  Singapore (ap-southeast-1) ★<br><b>FREE TIER AVAILABLE</b> |
|   |  |  Mumbai (ap-south-1)  |

Configura el cluster para permitir acceso desde cualquier IP:

En la pestaña superior Security, elije la subpestaña Whitelist IP y Add IP address.

Elige “Allow Access from Anywhere” o añade 0.0.0.0/0.


Clusters Build a New Cluster

Overview **Security**

MongoDB Users MongoDB Roles **IP Whitelist** Peering Enterprise Security

+ ADD IP ADDRESS

| IP Address | Comment | Status | Actions |
|------------|---------|--------|---------|
|------------|---------|--------|---------|

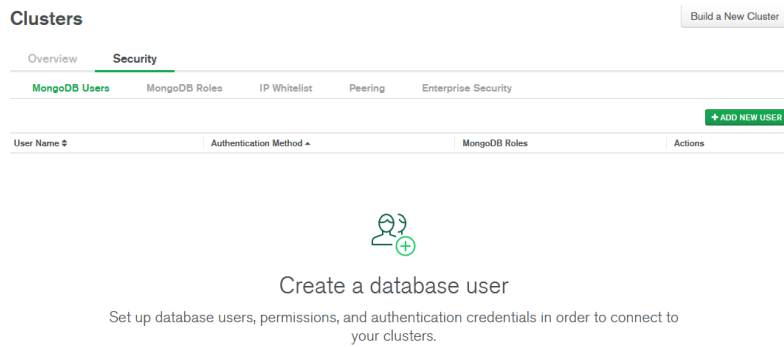


Whitelist an IP address

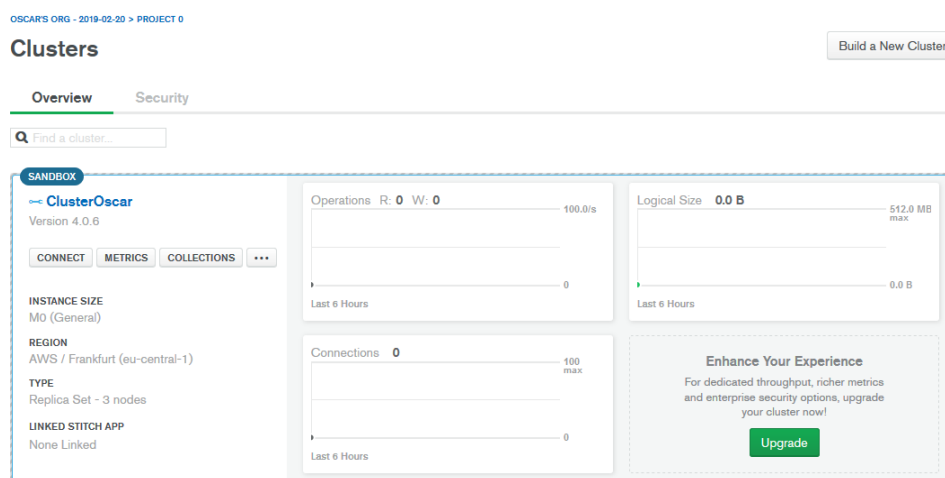
Configure which IP addresses can access your cluster.

[Learn more](#)

Crea un usuario de la BBDD con lo suficientes privilegios.



Entra en la opción CONNECT para obtener el URI que usar en las aplicaciones para conectarse:



Elige el método “Connect your Application”

Copia el “Short SRV Connection String” y sustituye <password> por el password de ese usuario. hay que quitar los < y >. Pej:

*mongodb+srv://oscarardaiz:mipass@clusterosca-al7xr.mongodb.net/test?retryWrites=true*

test será el nombre de la BBDD que se creará automáticamente. Si hemos creado una BBDD desde la pestaña collections, entonces tendremos que sustituir test por el nombre de la BBDD.

✓ Setup connection security > ✓ Choose a connection method > Connect

1 Copy the connection string compatible with your driver version:  
 Check which MongoDB versions your driver version is compatible with  
[See documentation on how to check the version of your driver](#)

Short SRV connection string (For drivers compatible with MongoDB 3.6+)

Standard connection string (For drivers compatible with MongoDB 3.4+)

Copy the SRV address:

mongodb+srv://oscarardaiz:<PASSWORD>@clusteroscar-al7xr.mongodb.net  
 /test?retryWrites=true

Copied!

COPY



Note: If using the node.js driver make sure you specify the name of your database after making your connection (example), otherwise your collections will all appear in a database called "test".  
 Alternatively you can replace "test" in the connection string with a different default database name.

En el siguiente paso el “Short SRV Connection String” lo copiaremos en heroku.com en la configuración de nuestro App de Heroku (el servidor Parse).

## Conectar Servidor Parse con MongoDB (en servicio de HEROKU)

-abrimos el menú de Settings de nuestra app en Heroku, donde nos permite configurar la app, crearemos una variable DATABASE\_URI y podremos el URI que anotamos anteriormente. (otras variables de configuración que podemos poner son APP\_ID, MASTER\_KEY, javascriptKey, restAPIKey, si no las pones sus valores por defecto son “myAppId”, “”, “”, “”)

### Config Vars

|              |                                  |   |
|--------------|----------------------------------|---|
| DATABASE_URI | mongodb+srv://oscarardaiz:mipass |   |
| KEY          | VALUE                            | Add   |

## Uso del servidor API Parse Server (II)

Ahora ya podemos crear objetos que se guardaran en la BBBDD.

Pje. Haciendo la siguiente petición POST a

<https://miservidorREST.herokuapp.com/parse/classes/GameScore>

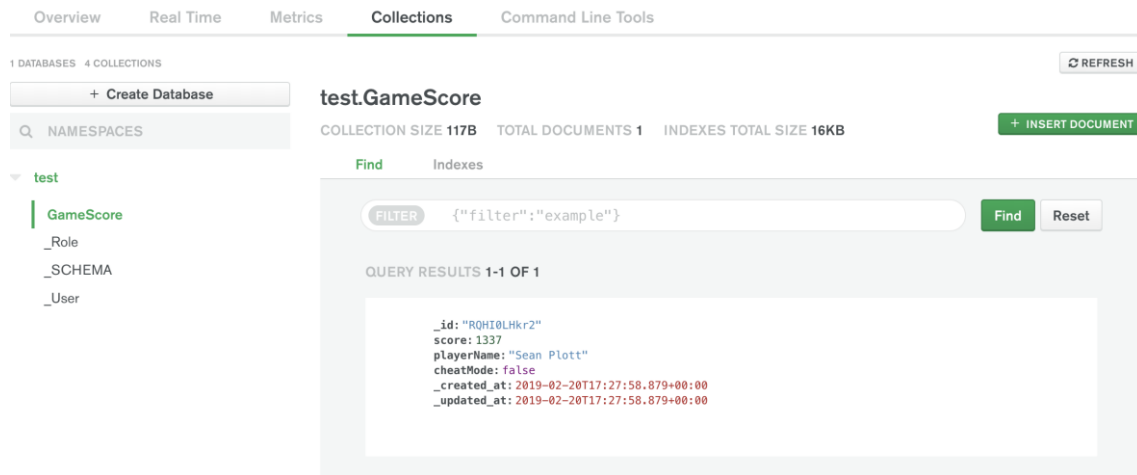
enviamos datos en formato JSON. En esta petición pedimos que se cree un objeto de la clase **GameScore** que tendrá tres campos con los valores que se incluyen. (Si el servidor no conoce la clase, se encarga de crearla, esta es una particularidad del servidor PARSE, otros servidores REST no lo tienen).

```
curl -X POST -H "X-Parse-Application-Id: myAppId" -H "Content-Type: application/json" -d '{"score":1337,"playerName":"Sean Plott","cheatMode":false}' https://miServidor.herokuapp.com/parse/classes/GameScore
```

La respuesta es un JSON con un objectId único y la fecha de creación:

```
{"objectId":"RQH1OLHkr2","createdAt":"2019-02-07T11:21:21.789Z"}
```

Si accedemos a nuestra BBDD de MongoLab, veremos que se ha creado una Collection *GameScore* (Collection es similar a una tabla en No-SQL, y cada fila de una tabla se llama Document) en una base de datos llamada *test*, la cual contiene los datos de los objetos que hemos creado. Antes se ha explicado que podemos llamar a la Database como queramos, pero ese nombre tiene que estar en el "Connection String".



## Ejercicio

En esta página están todas las peticiones REST que se pueden realizarse a un servidor API de Parse:

<http://docs.parseplatform.org/rest/guide/>

Si no diste un valor a la variable de configuración restAPIKey no es necesario usar el campo -H "X-Parse-REST-API-Key: \${REST\_API\_KEY}"

Prueba a hacer peticiones para:

- obtener un objeto,
- modificar un objeto,
- borrar un objeto,
- realizar Queries,
- subir un fichero,
- asociar un fichero con un objeto.