28/02/2021

08

**Fall**

**SISTEMAS INFORMÁTICOS**

*JulenJunguitu*

*Javier Ruiz*

*Mikel Goicoechea*

Documentación

**ITC Cuatrovientos - Pamplona**

TABLA DE CONTENIDOS

[1.- EL PROYECTO 3](#_Toc65338152)

[2.- SESIONES DEL PROYECTO 3](#_Toc65338153)

[1. Sesión 1 – 3 3](#_Toc65338154)

[2. Sesión 4 – 5 7](#_Toc65338155)

[3. Sesión 6 - 16 8](#_Toc65338156)

# 1.- EL PROYECTO

Adaptación del proyecto individual realizado en Programación Multimedia y Dispositivos Móviles “Memory” en el cual estan involucrados 3 módulos de segundo curso del ciclo Desarrollo de Aplicaciones Multiplataforma.

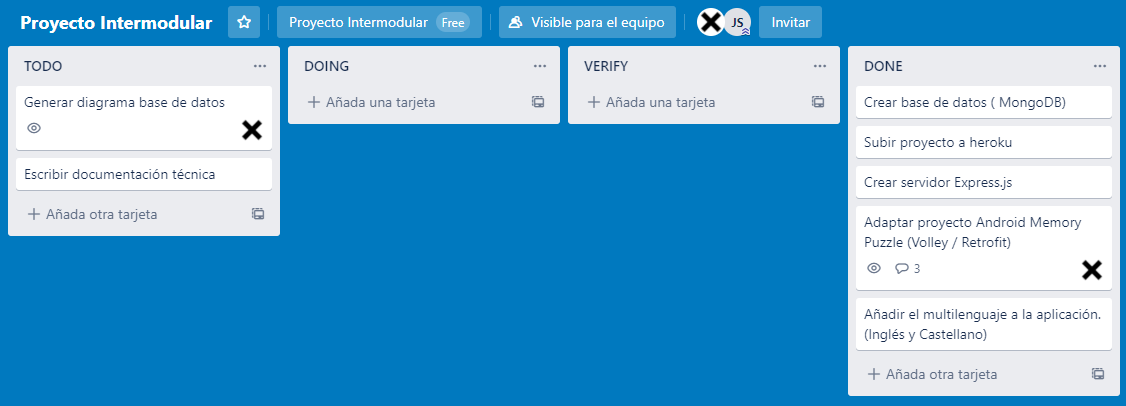
El proyecto consiste en adaptar el juego para que realice el intercambio de información entre diferentes dispositivos móviles, como por ejemplo para compartir un ranking de jugadores con puntos.

Será necesario que la aplicación móvil se comunique con otra aplicación desarrollada en JavaScript que actuará de servidor. El servidor se encargará de guardar un ranking global del juego y enviarlo a los clientes para que puedan consultar el ranking global desde la aplicación. El servidor utilizara una base de datos relacional para almacenar los datos necesarios para el funcionamiento del juego.

# 2.- SESIONES DEL PROYECTO

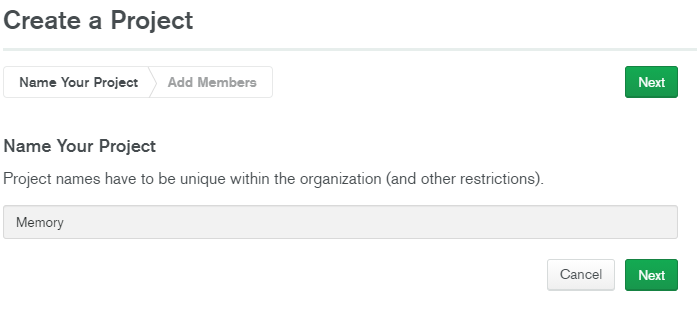
## Sesión 1 – 3

Organizamos el trabajo que íbamos a hacer en un tablero Trello.

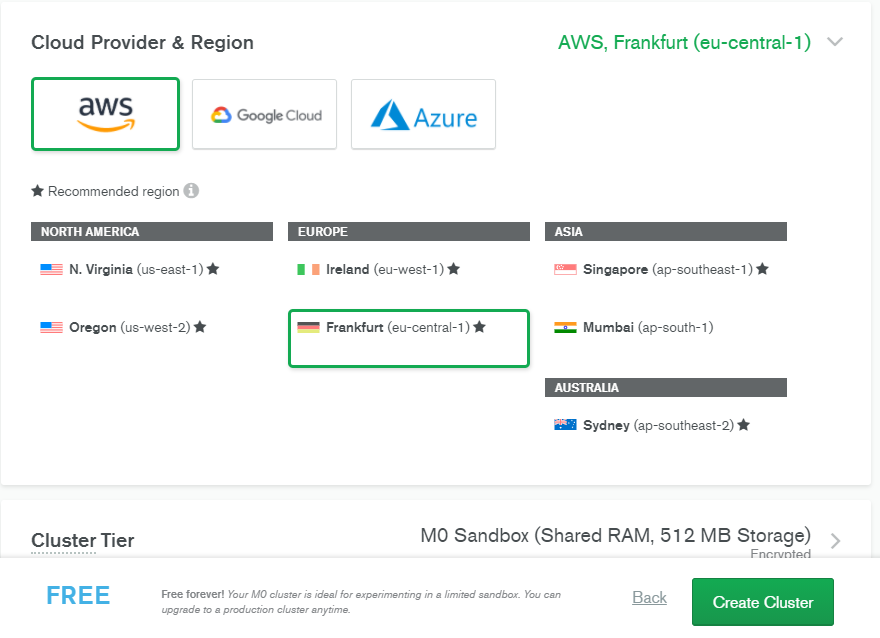


El primer día lo dedicamos para crear la base de datos en MongoDB, algo que hicimos sin dificultad y en 10 minutos.

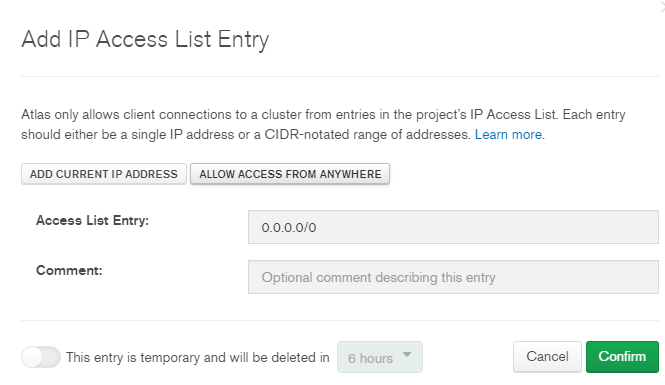
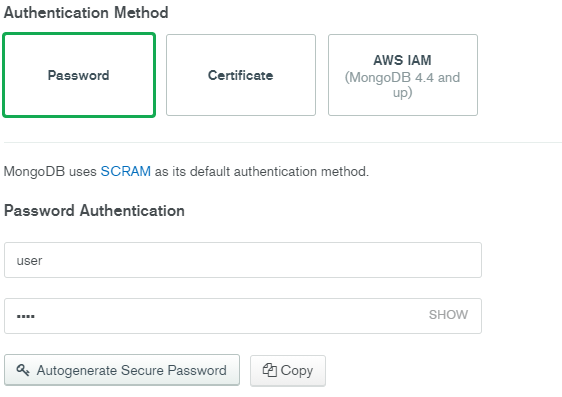
Para crear la base de datos utilizamos Mongo Atlas. Creamos un proyecto.



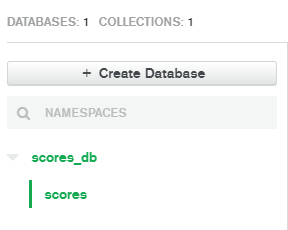
Creamos el Cluster



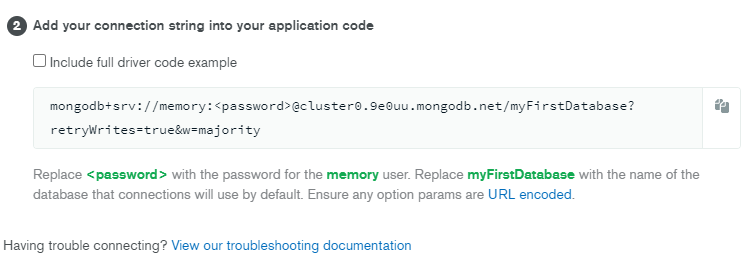
Creamos el usuario que va a entrar en la base de datos y desde que IPs se puede acceder en la cual pusimos que se puedan conectar desde cualquier IP.



Vamos al apartado Collections en cluster y creamos la base de datos y la colección de “scores”.

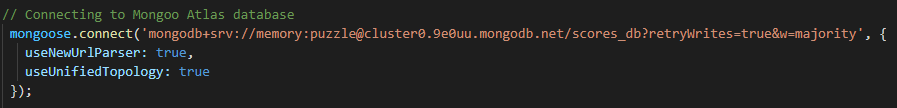


Y Ya solo queda conectar a nuestra aplicación con la siguiente línea. Esta línea la meteremos dentro

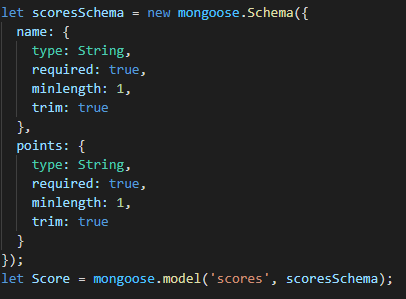


Después, creamos el servidor Express que serviría para hacer todas las funciones de almacenamiento y listado de elemento de la base datos de MongoDB en Android (POST y GET). Esto nos costó algo más ya que tuvimos que crear el schema de un objeto score con sus campos, y definir como funcionaban las ya mencionadas funciones POST y GET.

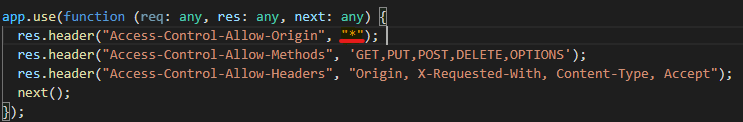
En el servidor hay que añadir la línea de conexión.



Junto con el Schema

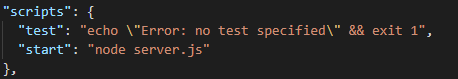


Y hay que poner desde que dominio se va a conectar al servidor. Ponemos \* para que se pueda conectar desde cualquier lado.



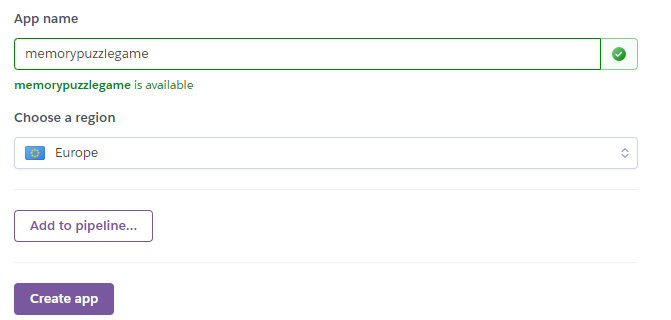
Ya para terminar hay que poner estas líneas para que el Express se pueda utilizar desde Heroku.





## Sesión 4 – 5

En Heroku creamos un nuevo proyecto



Nos pide instalar el CLI de Heroku y seguir unos comandos para subir el el servidor.

$ heroku login

$ heroku git:clone -a memorypuzzle

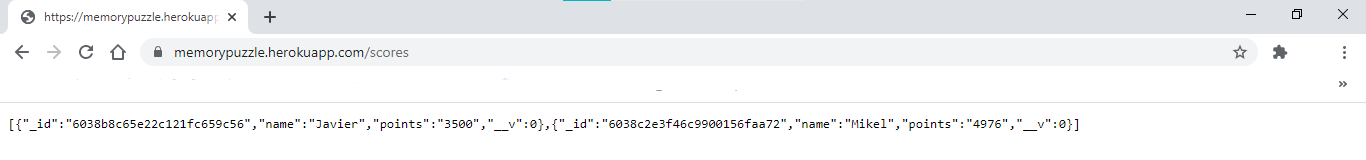
$ cd memorypuzzle

$ git add .

$ git commit -am "make it better"

$ git push heroku master

Una vez ya subido podemos acceder a, en este caso, *https://memorypuzzle.herokuapp.com/scores*



Ya podemos ver que funciona bien. Ahora tenemos que implementar los métodos para que funcione con nuestra aplicación Memory.

## Sesión 6 - 16

A continuación, vino algo de lo más complicado, añadir las funciones GET y POST a nuestro proyecto Android Studio el cual funcionaba para listar todas las puntuaciones de la base de datos y cuando jugamos una partida que se añada a la misma. Elegimos Volley y no Retrofit(Añadimos las dependencias necesarias al Manifest), y tuvimos bastantes problemas ya que muchos ejemplos y tutoriales que seguíamos de internet no nos funcionaban. Finalmente, encontramos una función de GET en una página web que al debuggear no daba errores y parecía que iba bien, nos dimos cuenta que la función estaba bien pero que había un error que no permitía la conexión HTTP, lo cual lo resolvimos añadiendo un fichero al XML que permitía conexiones HTTP y luego lo llamábamos desde el Manifest, y en vez de poner Localhost en la URL que hacía el GET, pusimos la IP del equipo ya que al parecer localhost no funciona en Android Studio. Para la función POST no usamos nada de internet, adaptamos la función GET que ya teníamos y nos funcionó perfectamente.

Más tarde, arreglamos un problemilla con el margen del fragment que hacía que la lista de puntuaciones no se viera bien en un dispositivo real e hicimos el multilenguaje traduciendo todos los strings. Creamos un archivo para español de España y lo copiamos del strings, y otro para inglés de Inglaterra y traducimos todos los strings.

Por último, utilizamos Code Iris para generar el diagrama de datos, un plugin que descargamos de internet e integramos en Android studio, el cual genera el diagrama automáticamente.

Diagrama de clases:

