****

**Manual técnico**

**2020**

**Manual Técnico de AvantiParking**

Tabla de contenido

[Base de datos: 3](#_Toc41029957)

[Spring Boot 6](#_Toc41029958)

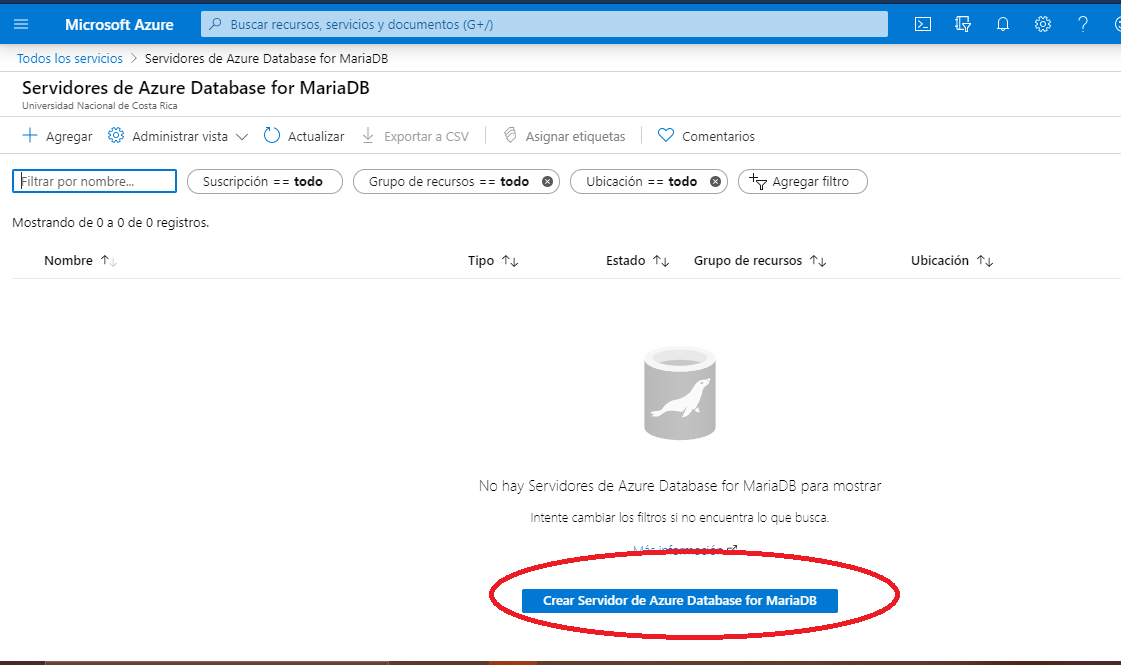
[Angular 8](#_Toc41029959)

[Deploy de Spring Boot en Azure 9](#_Toc41029960)

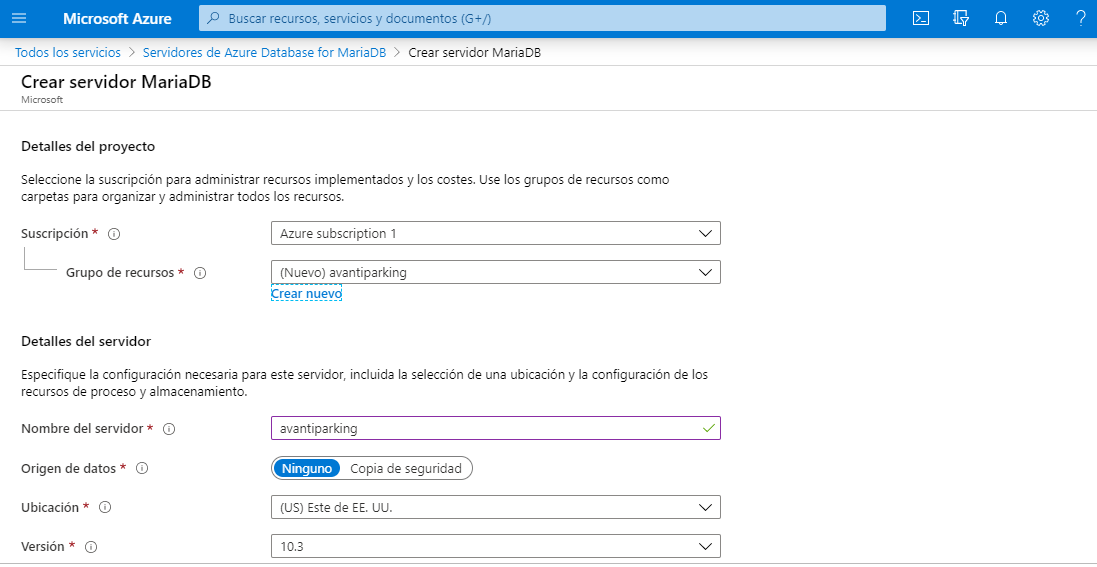
[Angular deploy 11](#_Toc41029961)

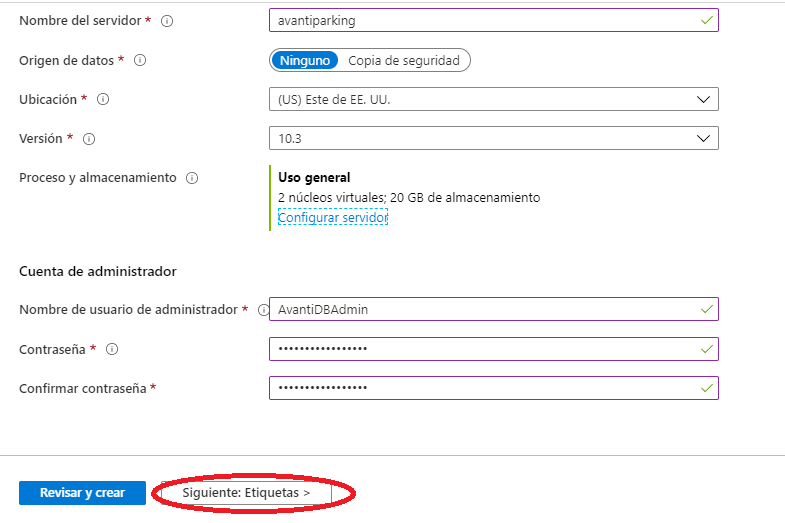
[Referencias: 13](#_Toc41029962)

**Base de datos:** Para la creación y montaje de la base de datos en un servidor se implementaron los servidores de Azure Database para MariaDB.  
1-Primeramente, se procedió a la creación de un nuevo servidor.



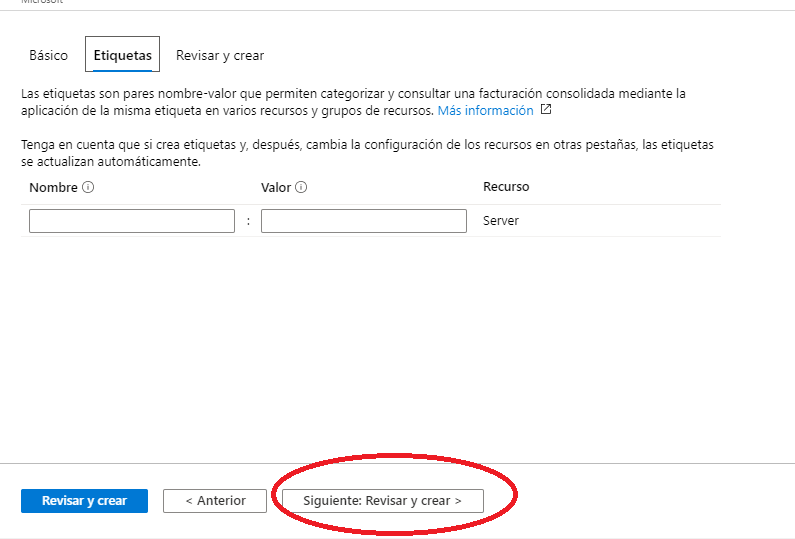
2-Rellenamos las credenciales, de la siguiente manera:

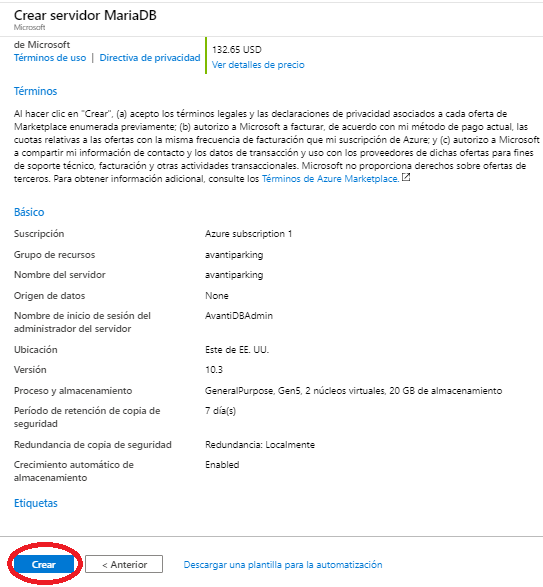




20 gigas de almacenamiento, con 2 nucleos, el nombre del usuario administrador será AvantiDBAdmin y la contraseña Avantiparking2020

3-En la página siguiente del formulario dejamos las opciones por defecto:



4-Por último, se mostrará un resumen de como nos quedaría la BD si todo esta correcto procedemos a realizar su creación.

**Spring Boot**  
  
Base de datos:

Dentro del proyecto de Spring Boot, necesitamos irnos a src/main/resources/application.yml  
  
datasource:

url: jdbc:mysql://avantiparking-db.mariadb.database.azure.com:3306/avantiparking?useUnicode=true&useJDBCCompliantTimezoneShift=true&useSSL=false&useLegacyDatetimeCode=false&serverTimezone=UTC

username: master@avantiparking-db

password: Aekm2020

Como vemos en el ejemplo de arriba, la base de datos se compone de una url, username y un password, así que lo unico que debemos hacer es poner nuestras credenciales en ese apartado.

Email:

Dentro del proyecto de Spring Boot, necesitamos irnos a src/main/resources/application.yml

mail:

host: in-v3.mailjet.com

port: 587

username: a9183edea445a2346abdb068d8eed198

password: d9f033b49ac3cfe3711a8edd1881ef3e

properties:

mail:

smtp:

auth: true

connectiontimeout: 5000

timeout: 5000

writetimeout: 5000

starttls:

enable: true

Como vemos en el ejemplo de arriba, el apartdo email, solo debemos de colocar nuestras credenciales, host, port, username y password, asdemás de poder modificar otras aspectos un poco más opcionales. Además, en el apartado src/main/java/com.avantiparking.controller/Email\_Controller.java en la linea 44, debemos de colocar nuestro dominio, de apartado Frond End, ejemplo:

String emailHtml = i.TempleteEmailAvantiParking(email.getSubject(),email.getText(),email.getHtml(), "https://test-avantiparking.web.app/", **true**, "Go to AvantiParking");

Autentificación con Google:

Dentro del proyecto de Spring Boot, necesitamos irnos a src/main/resources/application.yml

security:

oauth2:

client:

registration:

google:

clientId: 458329009342-pmqi5sqaj68oih2sdoemrfkh379fd99o.apps.googleusercontent.com

clientSecret: bt-qeGINzt005bI4xWM1oIyv

redirectUriTemplate: "{baseUrl}/oauth2/callback/{registrationId}"

scope:

- email

- profile

Solo debemos colocar nuestras credenciales de la API de autentificación con google, en clientId y clientSecretUna captura de pantalla de una computadora

Descripción generada automáticamente

Las cuales obtenemos en la consola de google, después de crear un proyecto y habilidar la API de gmail

Además de esto, es importate que agreguemos el, token y su expiración, como tambíen agregar dominio/oauth2/redirect del Frond End en la sección de redireccionamiento

app:

auth:

tokenSecret: 926D96C90030DD58429D2751AC1BDBBC

tokenExpirationMsec: 864000000

oauth2:

authorizedRedirectUris:

- http://localhost:4200/oauth2/redirect

- https://test-avantiparking.web.app/c

# Angular

API

Dentro del proyecto angular, necesitamos irnos a src/app/services/api.modules.ts debemos de agregar el dominio de nuestra aplicación BackEnd  
  
export const domain: string="https://avantiparking-1590019036260.azurewebsites.net";

Nota: No coloque “/” al final del dominio.

Redireccionamiento

Dentro del proyecto angular, necesitamos irnos a src/app/services/user.service.ts debemos de agregar el dominio de nuestra aplicación FrondEnd para el redireccionamiento al autentificarnos con Google, de la siguiente manera:  
//Frond End  
public oauth2UrlRedirect = 'https://test-avantiparking.web.app/oauth2/redirect';

# Deploy de Spring Boot en Azure

1. Captura de pantalla de un celular

   Descripción generada automáticamenteLo primero que debemos hacer, es irnos a la raíz del proyecto, pero con la terminal.
2. Previamente debemos de tener maven instalado en nuestra MAC, una vez, tengamos este requisito cumplido, solo debemos de colocar el siguiente comando y esperar a que termine:  
   mvn clean install
3. Después de este comando, solo debemos de comenzar a configurar el proyecto, para su debido deploy, con el siguiente comando:

mvn azure-webapp:config

Captura de pantalla de un celular

Descripción generada automáticamenteSeleccionamos 1. Application

Captura de pantalla de un celular

Descripción generada automáticamente

Que nos quede algo similar a la terminar de ejemplo.  
  
Y se nos desplegara la siguiente información:  
  
Please confirm webapp properties

AppName : AvantiParking-1590019036260

ResourceGroup : AvantiParking-1590019036260-rg

Region : westeurope

PricingTier : PremiumV2\_P1v2

OS : Linux

RuntimeStack : JAVA 11-java11

Deploy to slot : false

Confirm (Y/N)? : Y

1. Al cual le daremos que Y y ya estariamos listos para hacer el deploy, pero antes de hacerlo, debemos loguearnos con Azure, con el siguiente comando:  
     
   az login
2. Una vez logueados, solo debemos de colocar el siguiente comando para hacer el deploy:

mvn azure-webapp:deploy;

Y listo, ya tenemos en el aire nuestro proyecto Spring Boot

# Angular deploy

Lo primero que debemos hacer, es colocar el comando en nuestra terminar, en la raíz del proyecto Angular

ng build  
  
Esto nos creara todo lo necesario para poder lanzar nuestro deploy, el cual se aloja en la carpeta dist/avantiparking.

Ahora creamos una carpeta, en el cual alojaremos nuestro deploy, y ingresaremos los siguientes comandos:

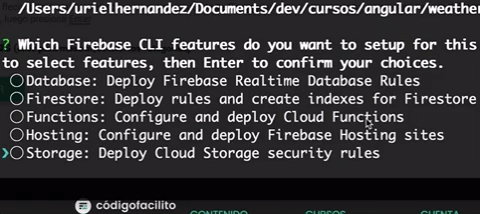
npm install -g firebase-tools

firebase login

firebase init

Este comando iniciará un asistente que te guiará por la configuración del proyecto, las opciones que debes seleccionar son:

1. Usando las flechas arriba/abajo selecciona qué característica de Firebase usarás, una vez que hayas seleccionado la del Hosting, presiona la barra espaciadora para activar el uso de esta característica, luego presiona Enter



Una vez seleccionado, debes seleccionar un nuevo hosting y lo creas en base al proyecto a necesitar

Un vez creado nuestro proyecto en Firebase, solo debemos mover los archivos de la carpeta dist/avantiparking a la carpeta public de unestro deploy y escribir el siguiente comando:

firebase deploy

# Referencias:

Deploy en Angular con FireBase:

<https://codigofacilito.com/articulos/deploy-angular-firebase>

Deploy en Spring Boot:

<https://www.youtube.com/watch?v=qFy9v454GyY&t=965s>

# 