Juego 21 blackjack

### Juan Polo; Harold Peña

UNIVERSIDAD DE CARTAGENA

Este proyecto basado en el juego 21(blackjack), se realizó a través de diferentes tecnologías las cuales a través de estás se logró trasladar nuestra aplicación a un funcionamiento de mejor nivel, que fueron implementar notificaciones, usar un servidor para nuestra aplicación, y servicios de Google.

**Tecnologías:**

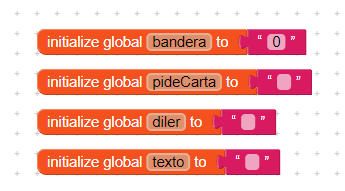
* Azure Microsoft
* Firebase Google
* Spark(microservicios)
* Kodular

**VISTA (kodular)**

En el video que se encuentra entre los demás archivos se explica el como crear e ingresar a una cuenta de kodular y como crear una app para Android

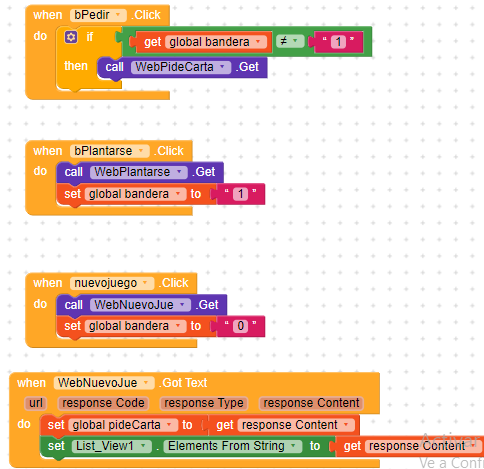
Bloques

Lo primero es crear las variables globales que se irán alterando al transcurrir el uso de la aplicación.

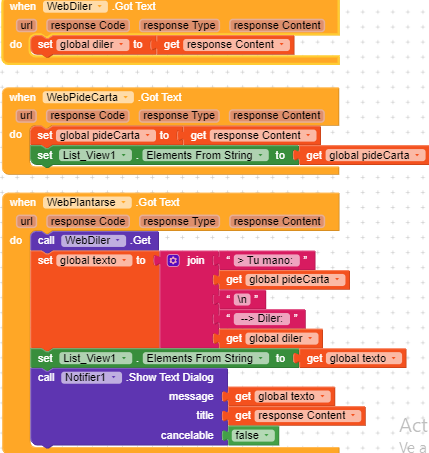


Se implementan en los bloques los llamados a los servicios rest que se encuentran en una maquina virtual de Azure, mas concretamente, ubicado en el sur de Brasil.

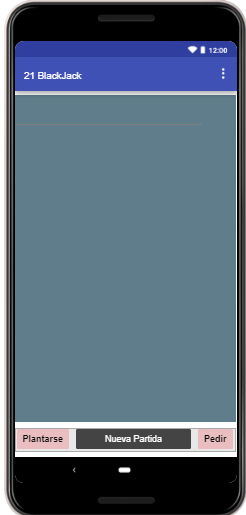
Se crean los botones bPedir, bPlantarse, y nuevojuego, que tendrán como lógica la implementación del llamado de los servicios, y al llamarse estos le regresan la información necesario para el juego, a través de un json le envían las letras de las cartas que se generaron aleatoriamente, como se refleja en el último bloque(WebnuevoJue) ahí se valida que al recibir la información se modifique el campo del listview por el que se recibió del servicio rest.



Aquí se realiza el mismo procedimiento. Los bloques siguientes quieren decir que los servicios al enviar la información y al validarse que, si se recibieron en la app, se modifican las variables globales anteriormente mencionadas, en WebDiler se recibe el puntaje del diler(maquina) con la que se está jugando, en webPideCarta se recibe las cartas de más que pidió el jugador (en caso de que lo haga). En WebPlantarse, se llama el webDiler para obtener su puntaje y se compara la mano del jugador y del diler, se declara un ganador y se notifica.

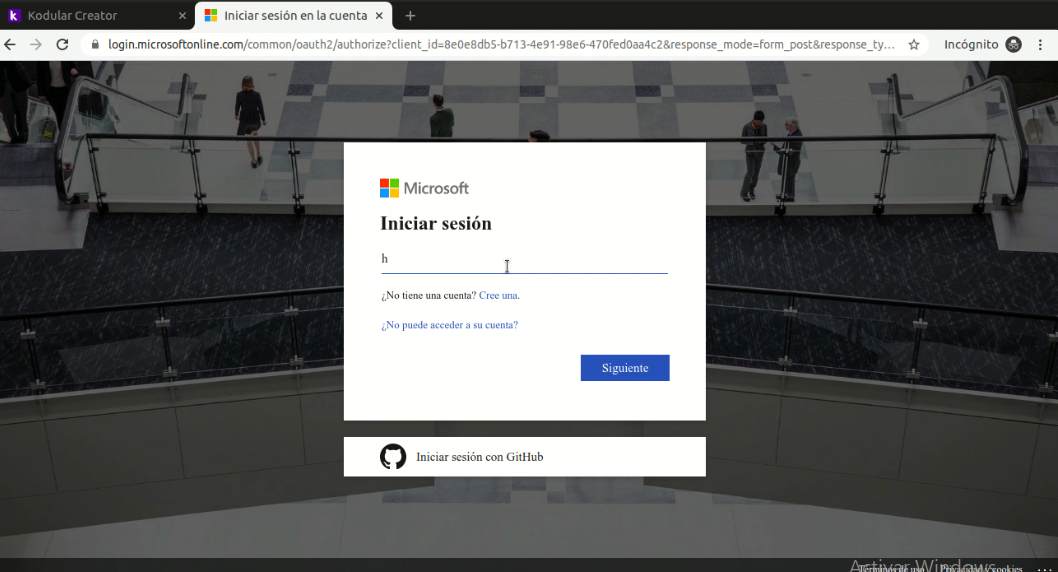


**Vista previa de la aplicación**

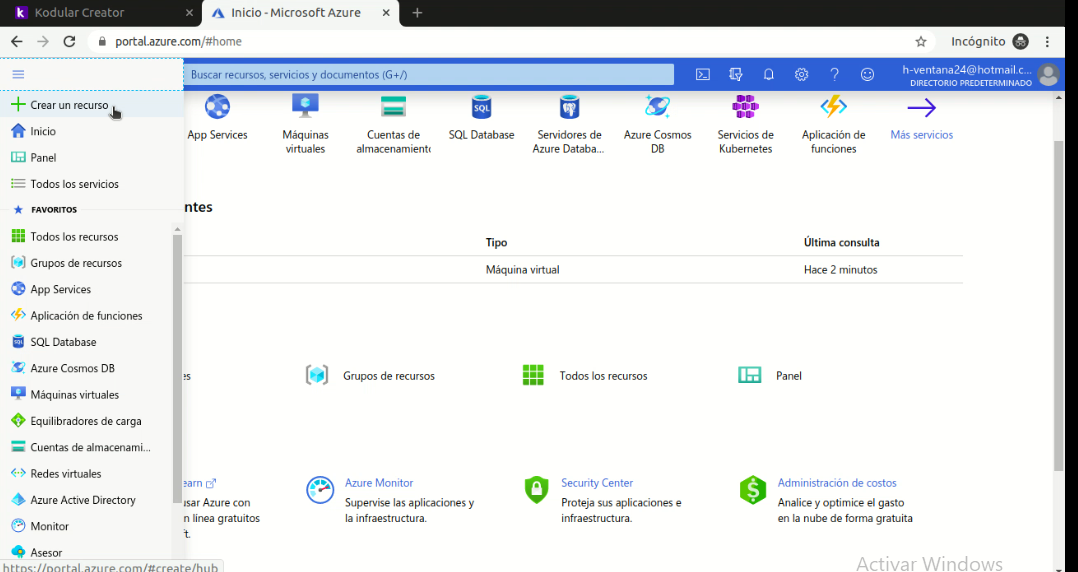


**Backend(azure)**

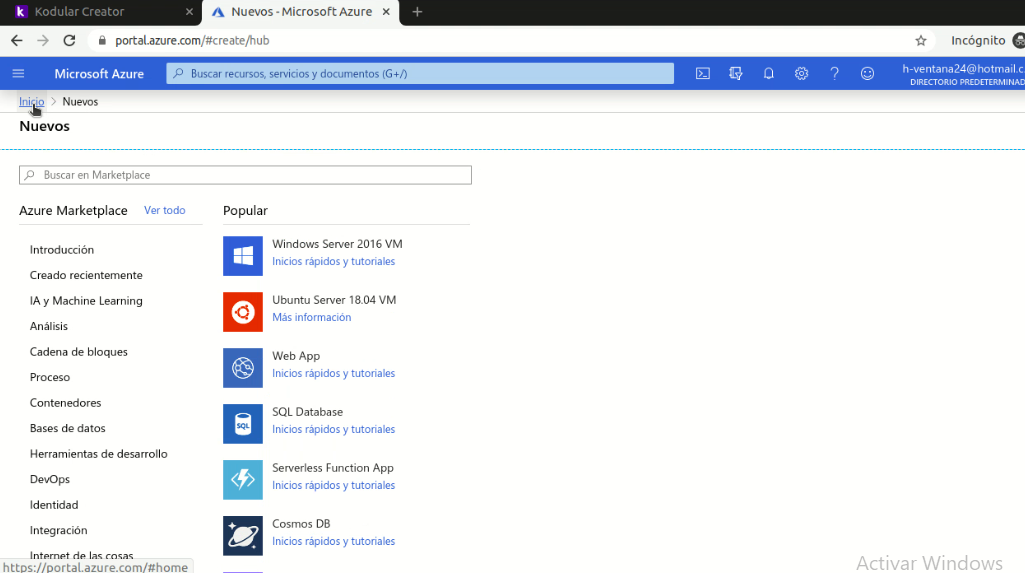
Lo primero sería crearse una cuenta de azure e iniciar sesión



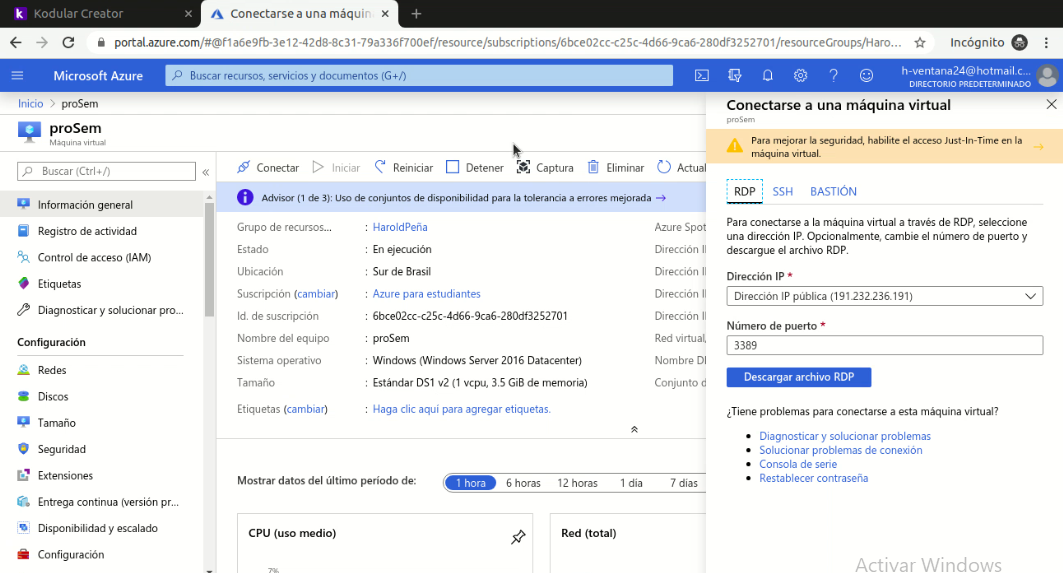
Nos vamos a la barra de menú, y vamos a crear un nuevo recurso en el que crearemos la maquina virtual



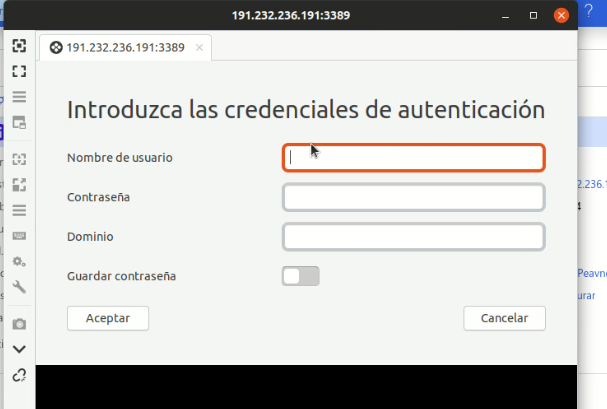
Se elige el tipo de SO que se usara en la maquina virtual, en nuestro caso, usamos Windows server



Se crea la maquina virtual con la dirección publica que se ve en la imagen, al intentar conectarse se pedirá de que tipo si RDP,SSh o BASTION, nosotros elegimos RDP, y se descarga un archivo que al abrirse saldrá nuestra maquina con el SO que se eligió.

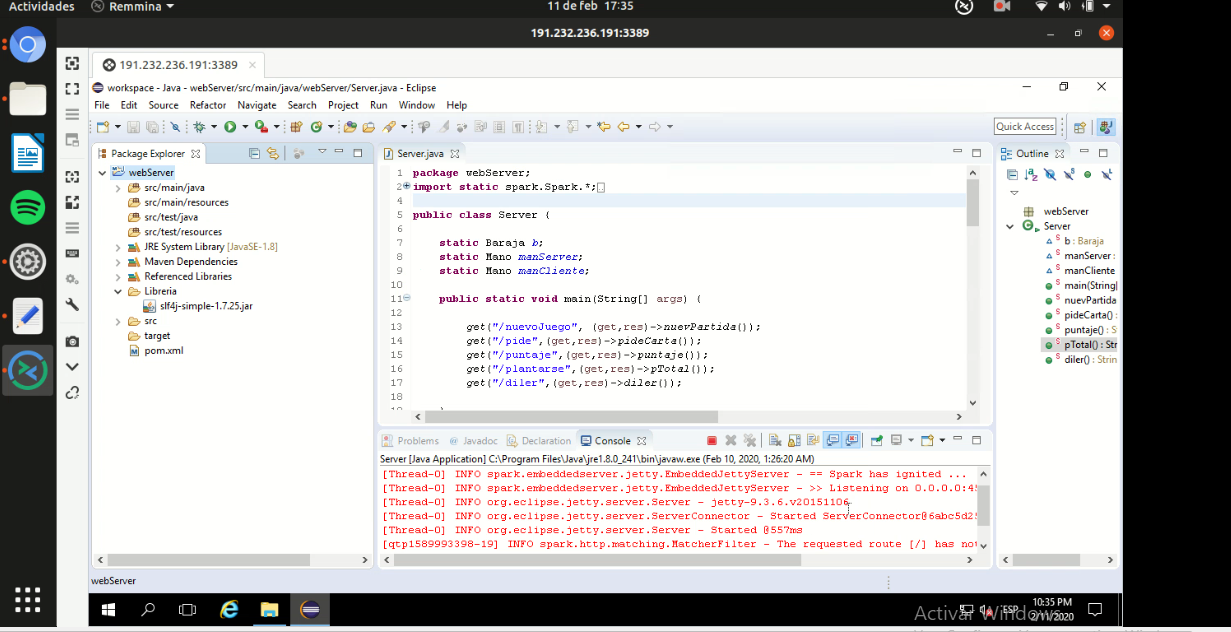


Pero, antes se nos pedirá unas credenciales de autenticación para ingresar, esta se nos dice en la imagen anterior, por lo que no es complicado hallarlas



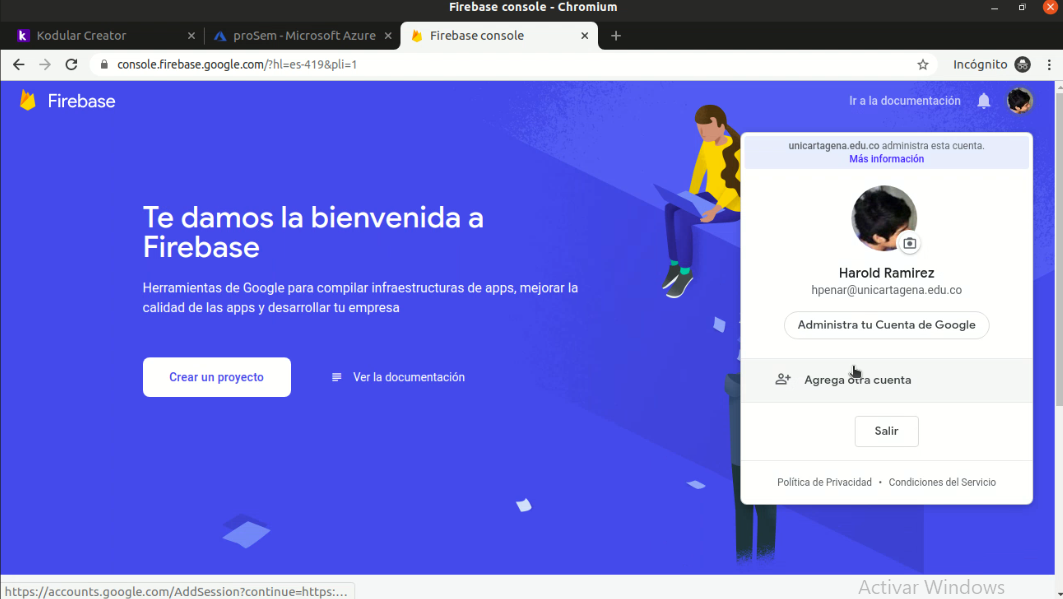
En este apartado lo que se hizo fue, en la maquina virtual, descargar todos los archivos necesarios para colocar a correr nuestro servidor, que son, Google Chrome, Eclipse, JDK, y la librería SPARk(En los documentos anteriores se explica su implementación).

Luego, en el entorno de eclipse, se implemente el proyecto WebServer que es donde con la librería SPARK, se crearon los servicios rest que es donde se hace todo el proceso de envío de las cartas al jugador, como se puede ver en la imagen.

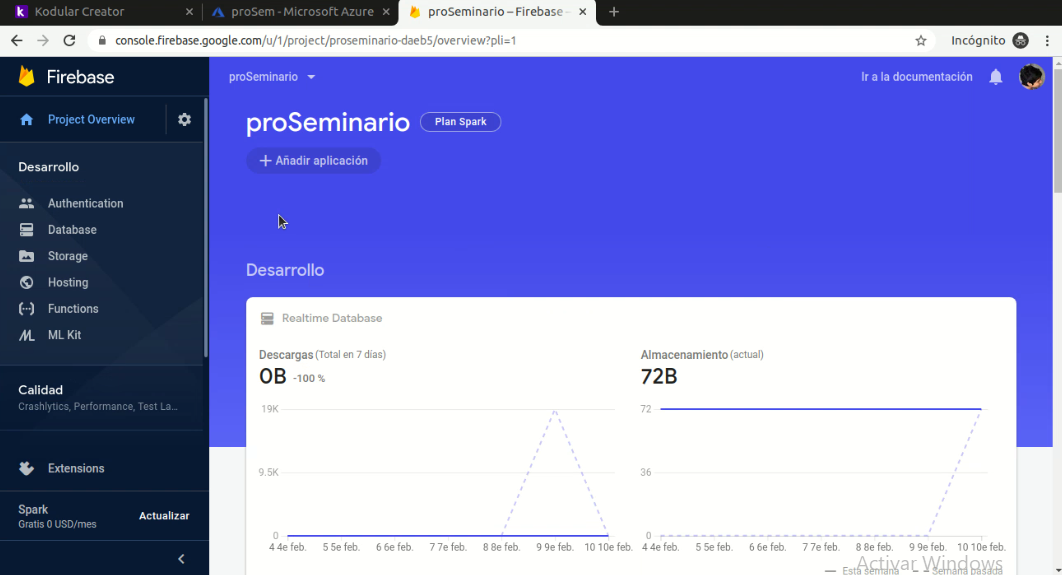


**Notificaciones**

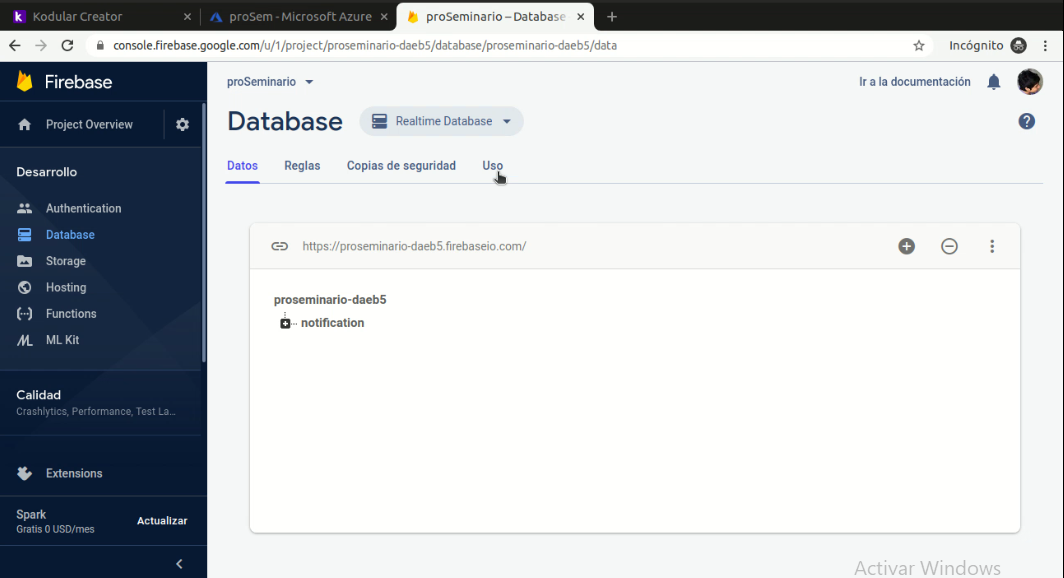
Se crea una cuenta en firebase de Google



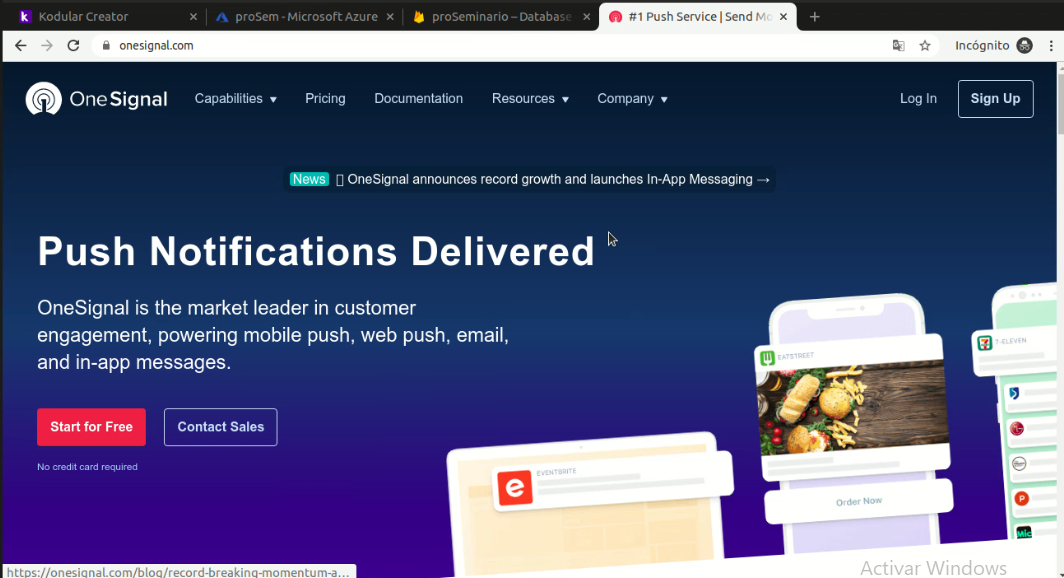
Este sería nuestro proyecto inicial en el portal.



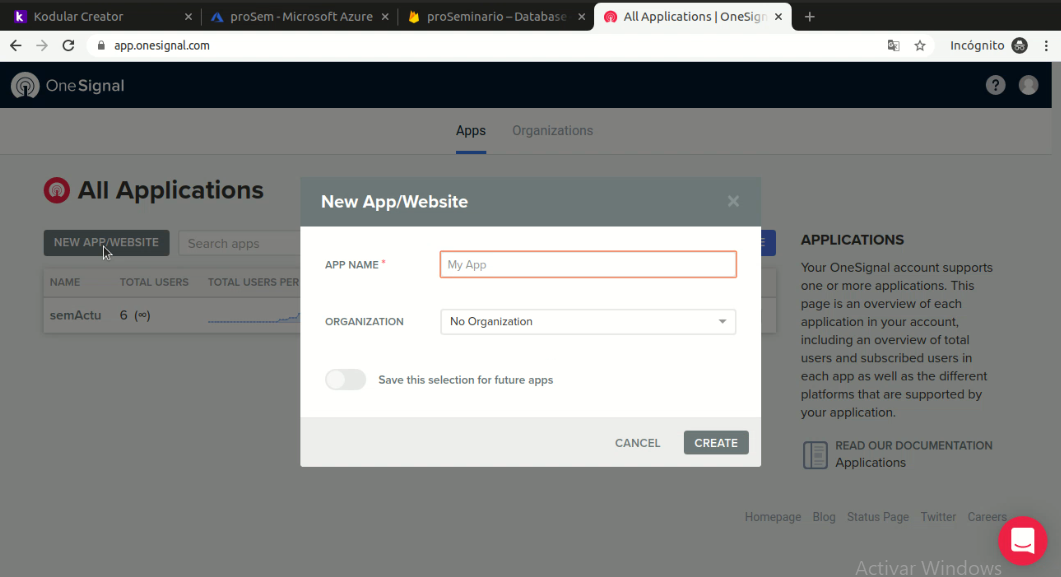
Nos vamos al apartado de database y creamos un mensaje tipo json



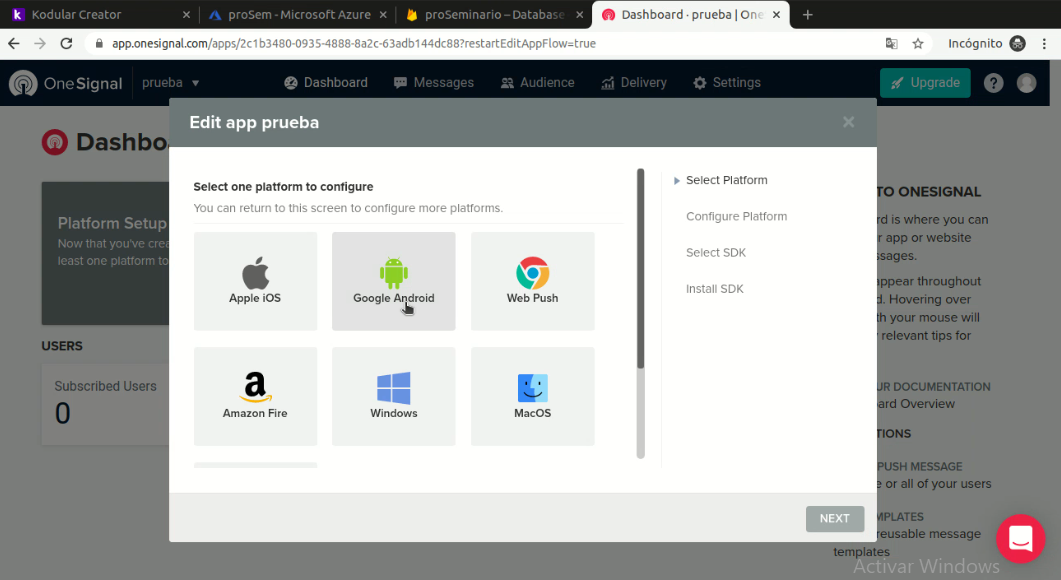
Nos dirigimos a one signal, nos creamos una cuenta y entramos



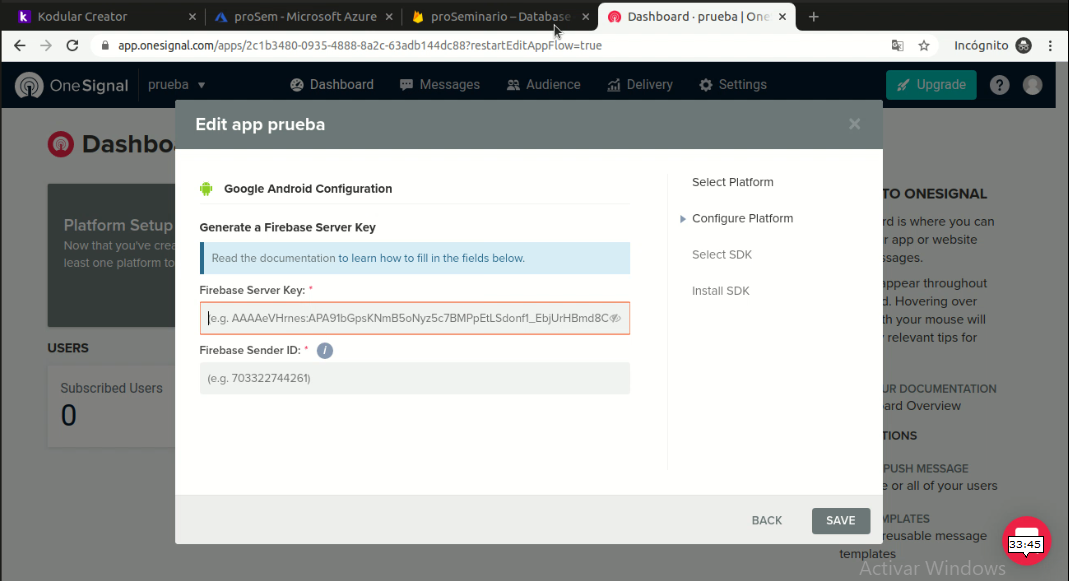
Damos clic en NewApp/WebSITE, le damos nombre a la app, y la organización a la que pertenece(opcional)



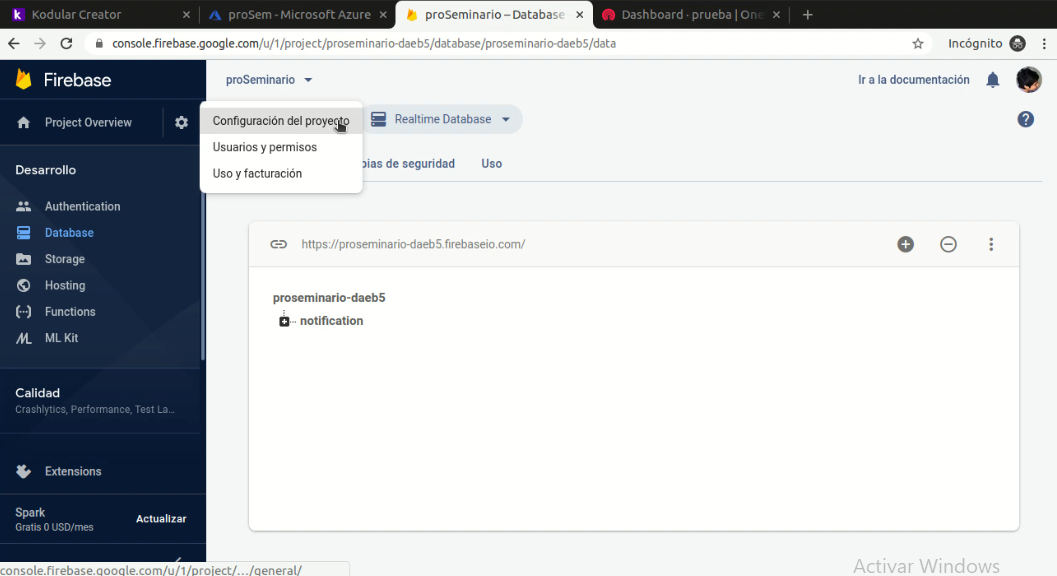
Seleccionamos la plataforma Android



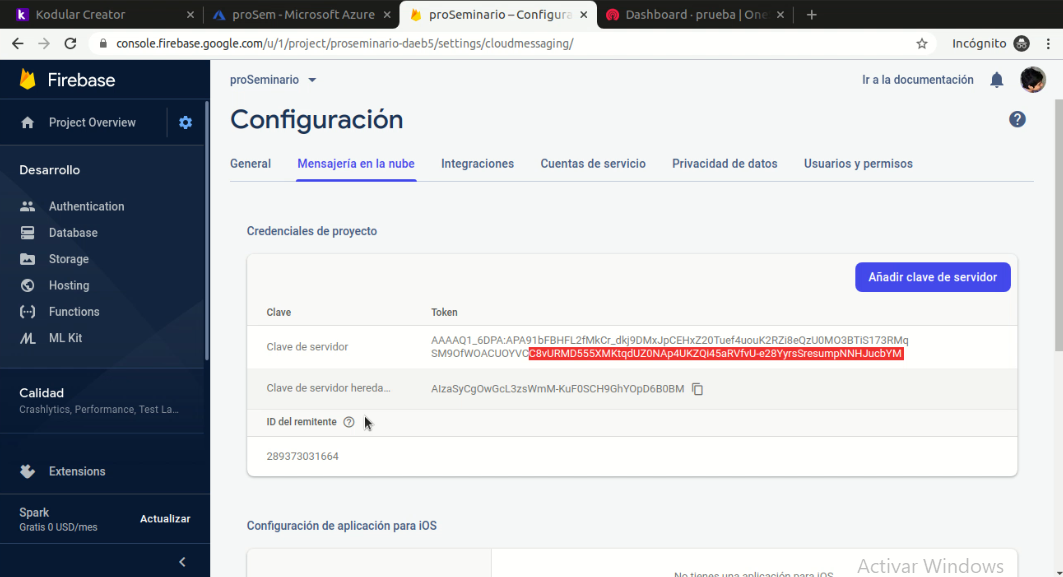
Aquí tenemos que ir a la plataforma de Firebase y buscar la llave del servidos y su ID



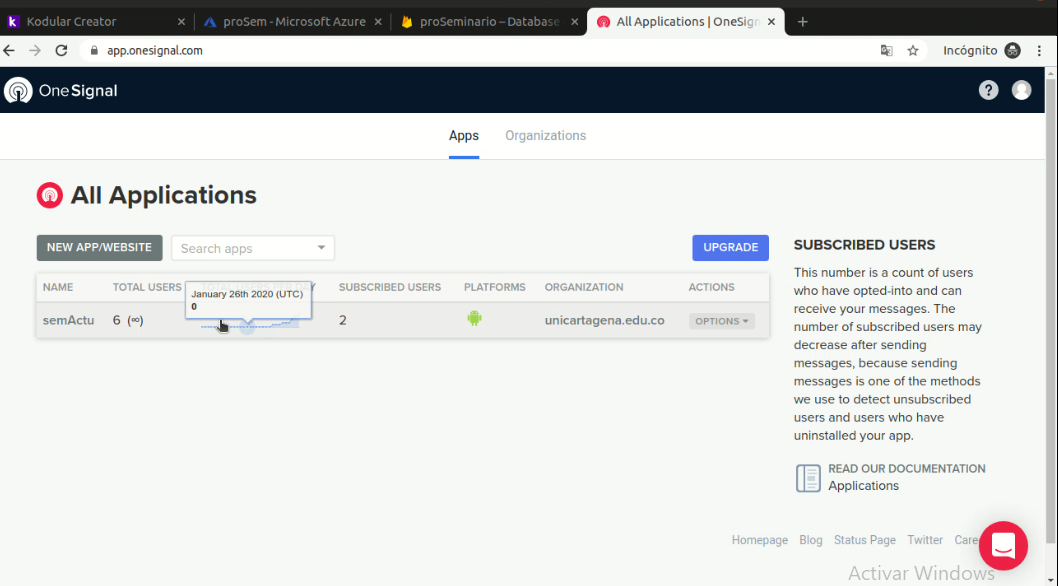
Nos vamos al portal de firebase y nos vamos a configuración del proyecto



Ahí encontremos la clave y el id remitente, lo copiamos en las campos solicitados en la pagina de oneSignal



Al finalizar, tendríamos nuestra aplicación en el portal de oneSignal, ahora tenemos que crear un mensaje automatizado para a través de un notificador de kodular recibir el mensaje desde oneSignal y firebase.



Creamos el componente de notificaciones en kodular y colocamos el id de one signal de nuestro proyecto

