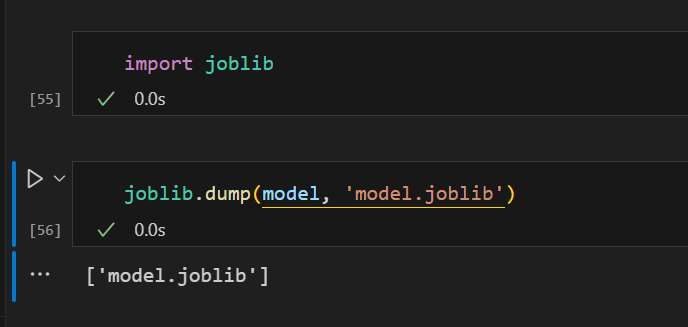
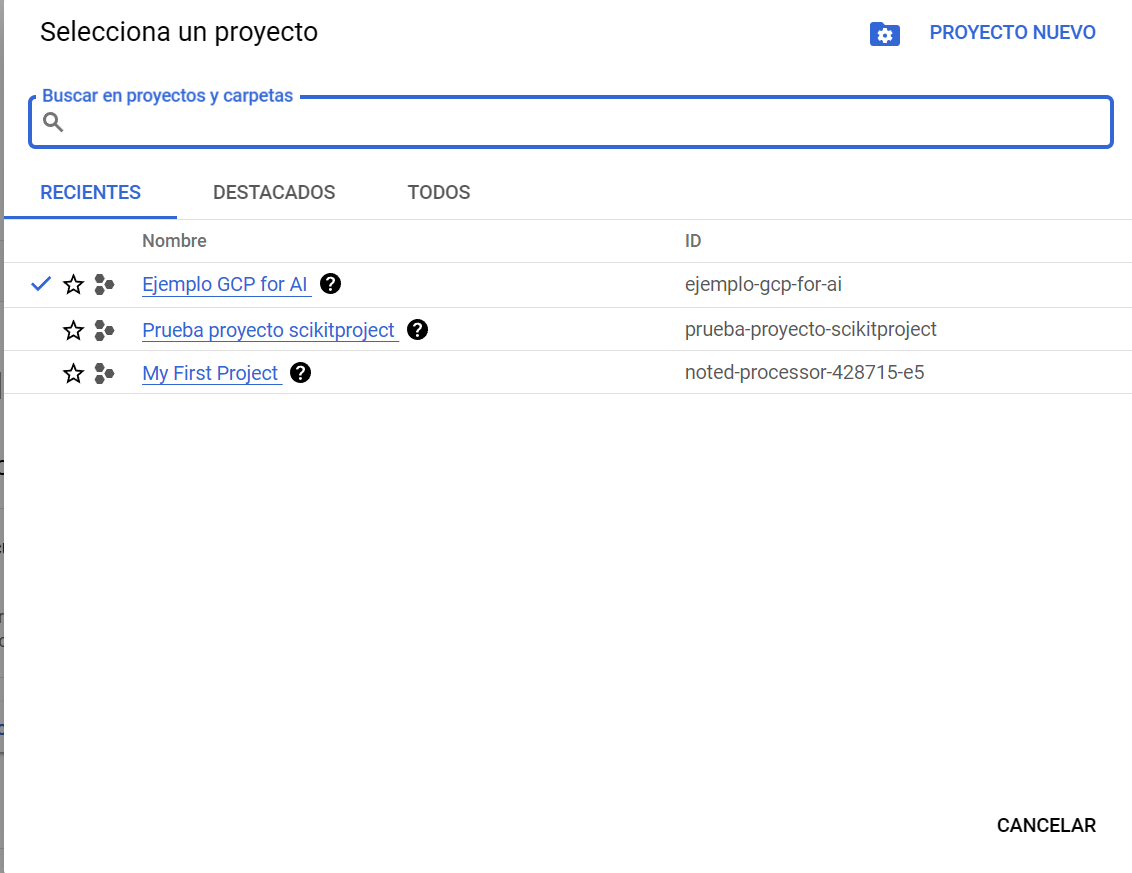
# Export del modelo



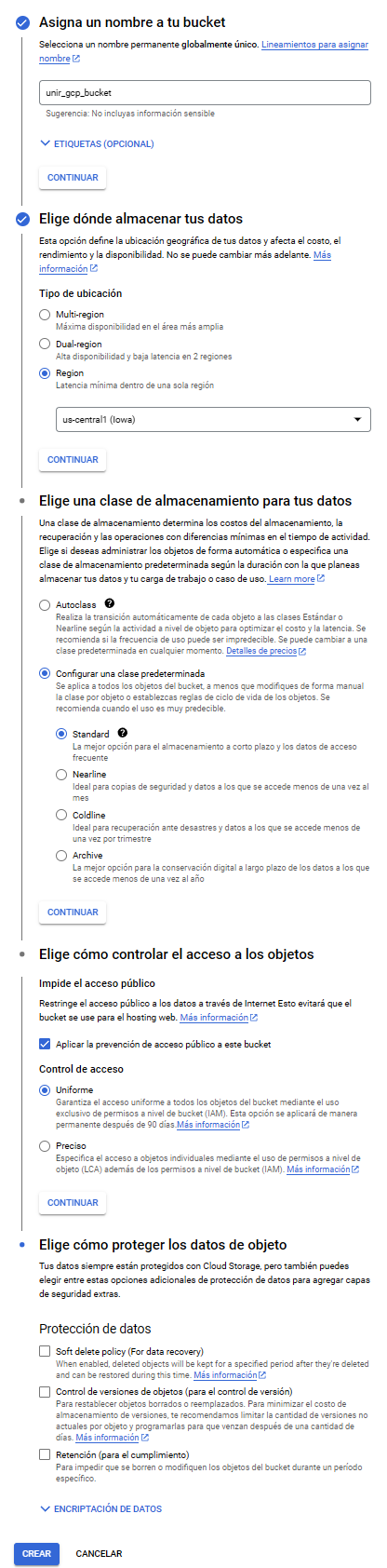
Exportamos el modelo para subirlo a la nube

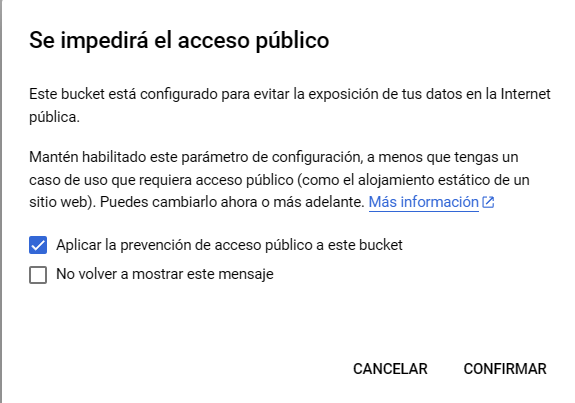
# Creación proyecto GCP

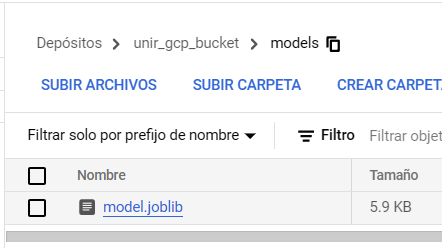


Creamos el proyecto de google cloud platform para realizar todo el procedimiento desde el mismo.

# Creación del bucket y subir modelo



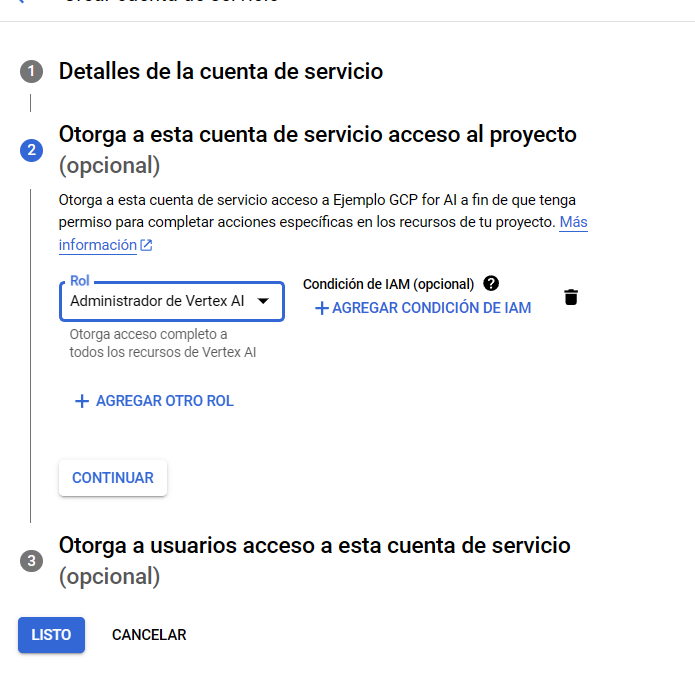




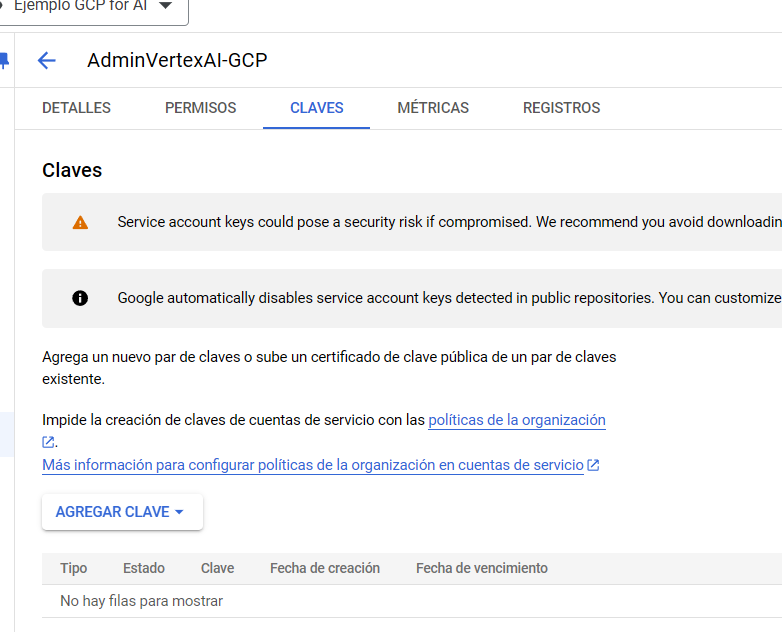
Creamos el bucket y subimos el modelo al bucket, en nuestro caso crearemos el bucket en una sola región.

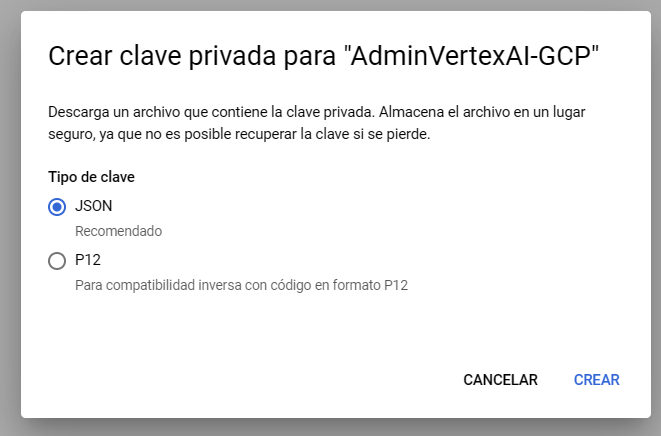
Es importante crear dentro del bucket una subcarpeta llamada models ya que es la que usaremos más adelante para vertex AI

# Creación de Service Account

Ahora procedemos a crear la cuenta de servicio, esta es la cuenta que usaremos para darle accesos y poder tener unos identificadores para acceder a la API.

Una vez creada la service account le tenemos que generar las claves, las cuales generaremos en formato JSON

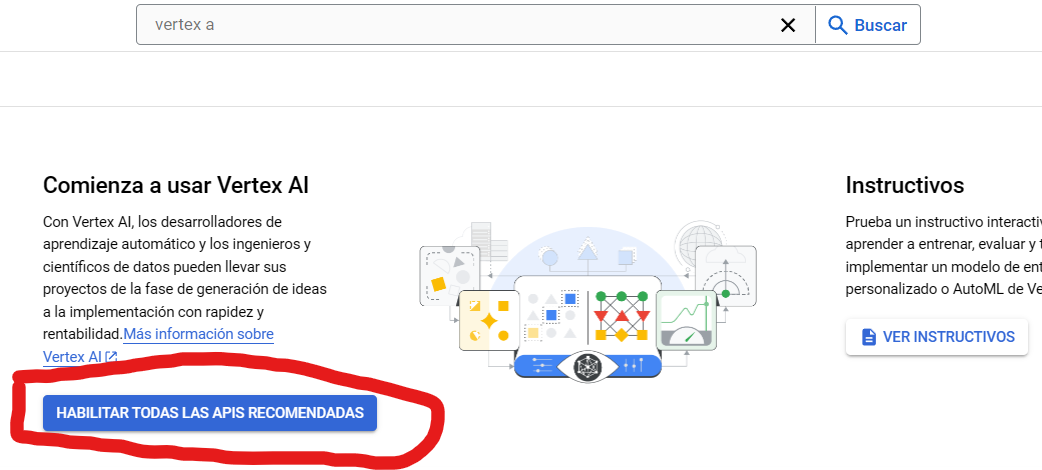


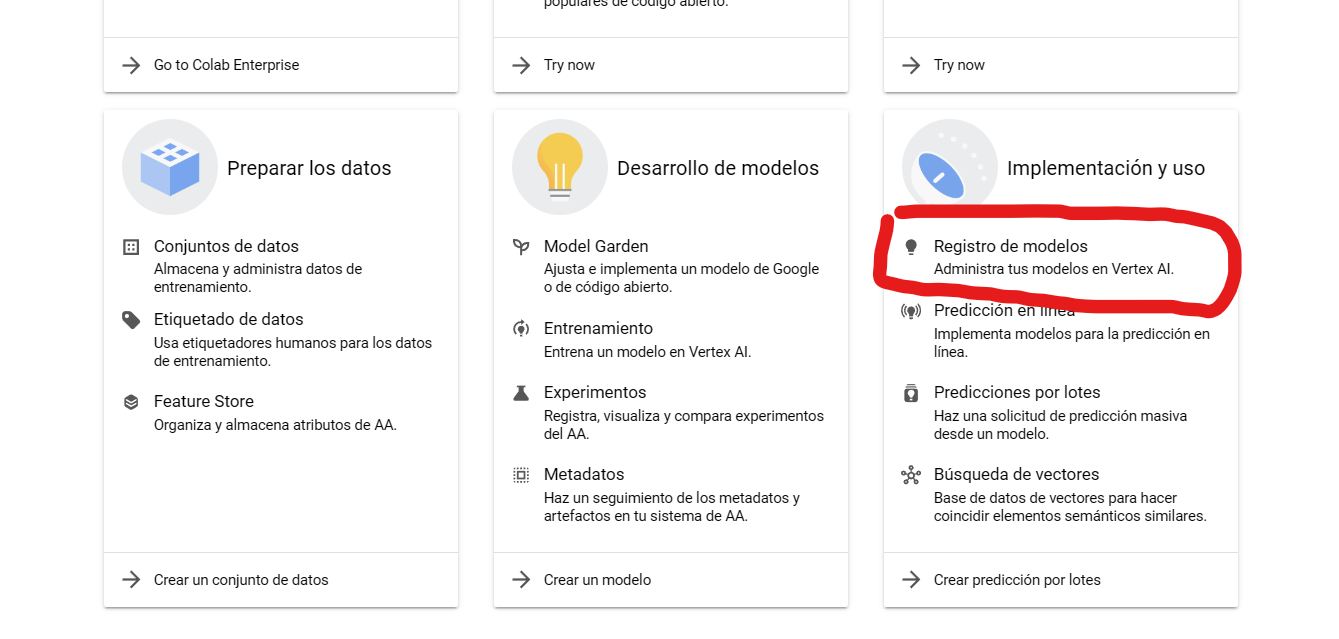


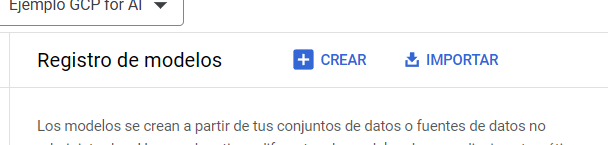
# Despliegue del modelo

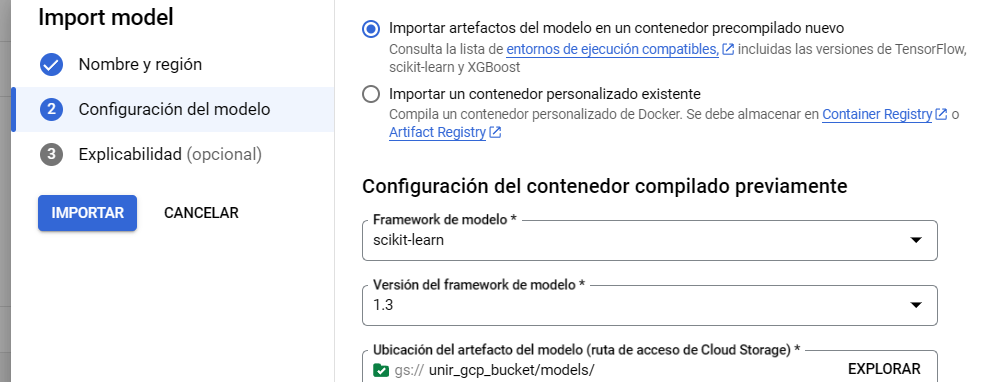
Para el despliegue del modelo utilizaremos Vertex AI

Primero habilitaremos las APIs recomendadas, seguido de eso iremos al apartado de registro de modelos, importaremos el modelo como modelo de sckit-learn 1.3 con ruta de acceso la carpeta models (*\*importante escoger la carpeta no el modelo en sí)*

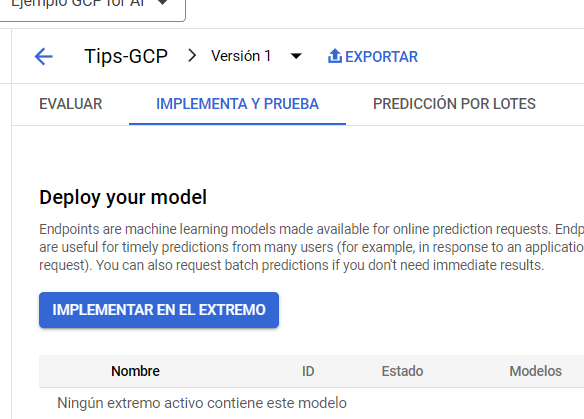
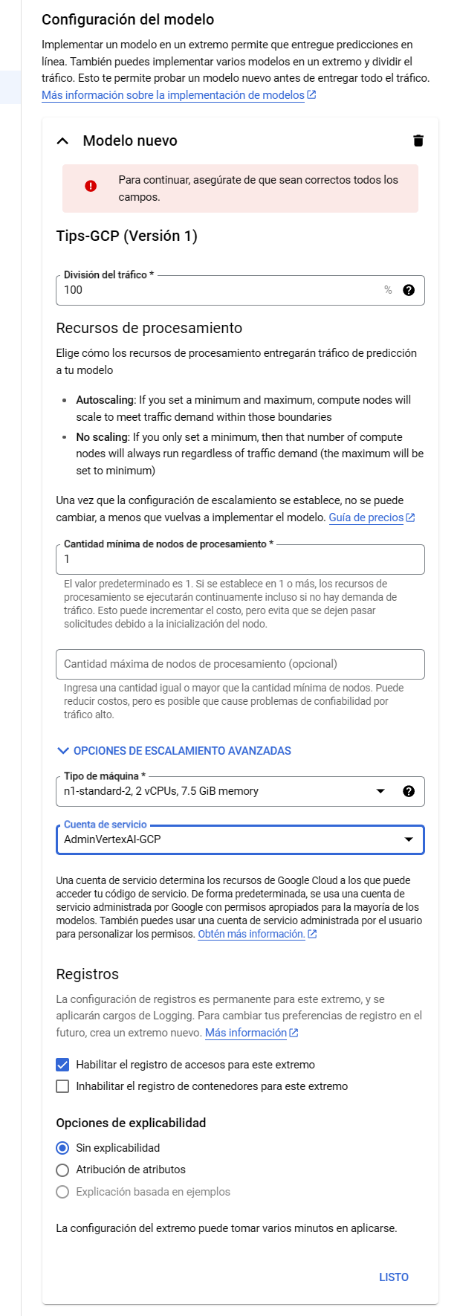


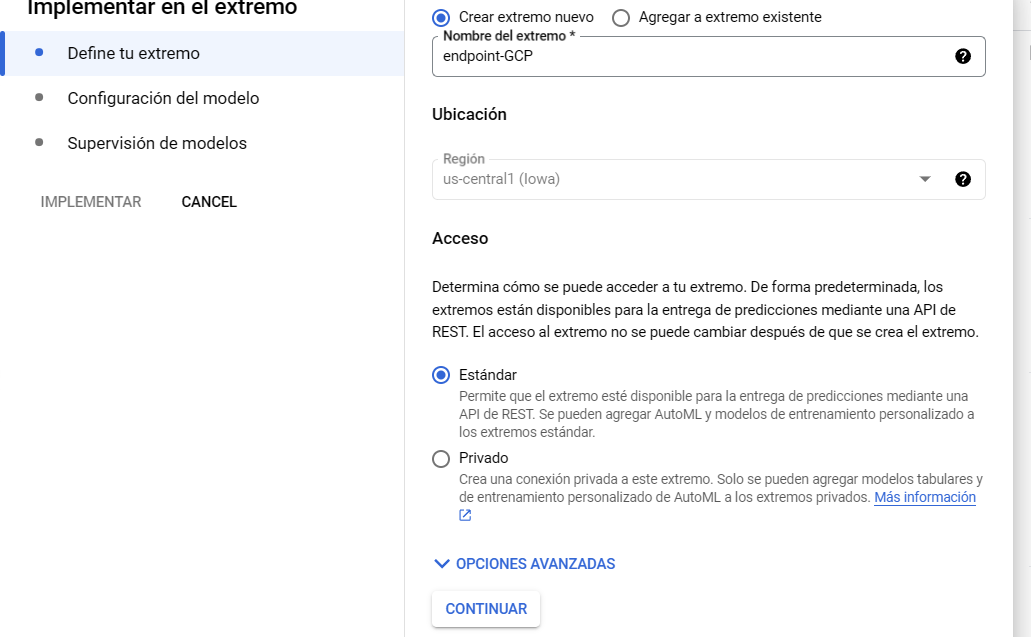




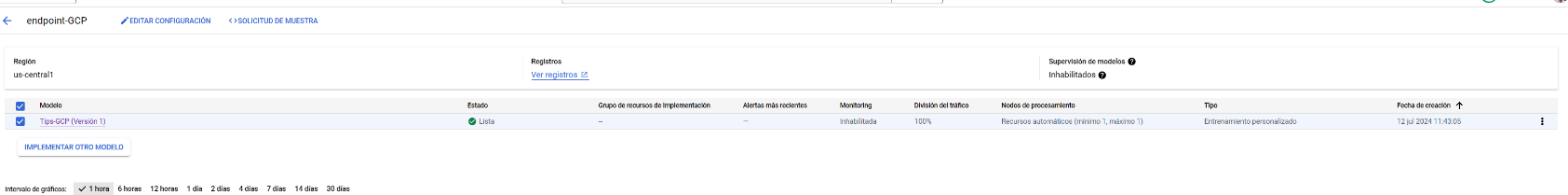
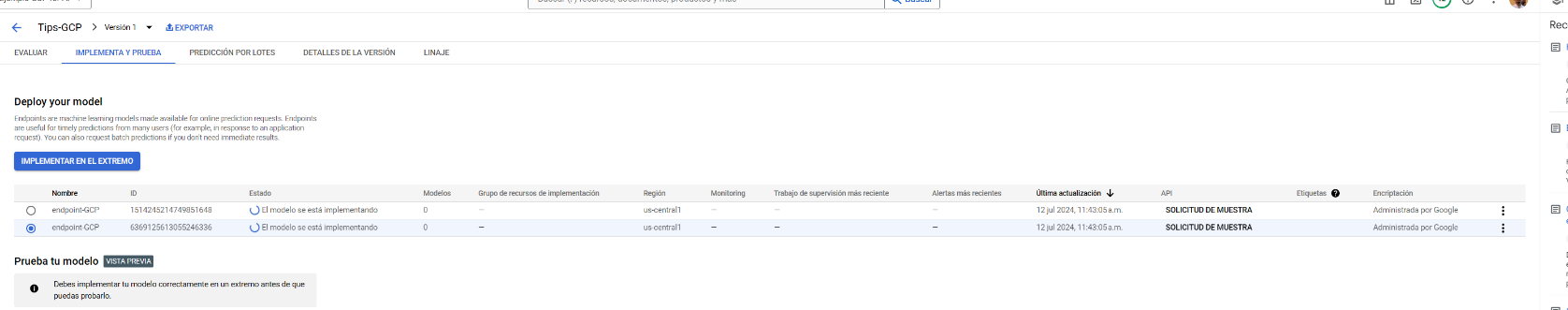


Tras esto iremos al modelo creado e iremos al apartado de implementa y prueba para crear asi el endpoint necesario para su acceso, este endpoint lo crearemos en acceso standard.





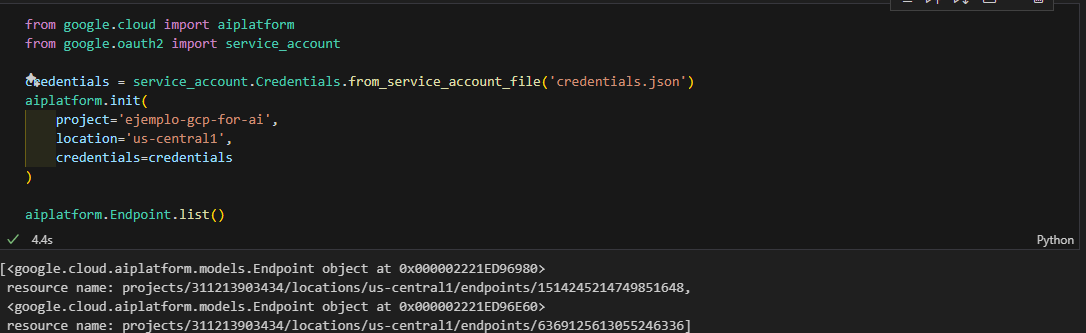
A la hora de la configuración del modelo es importante asignar una máquina que se adapte a las necesidades y potencia de nuestro modelo y se le asigne la cuenta de servicio que creamos con anterioridad para que tenga lo permisos correctamente asignados.



# Llamada del modelo desde Python

Utilizaremos la librería de google para acceder a nuestro modelo.

Primero configuraremos las credenciales en un archivo en nuestro caso “credentials.json”

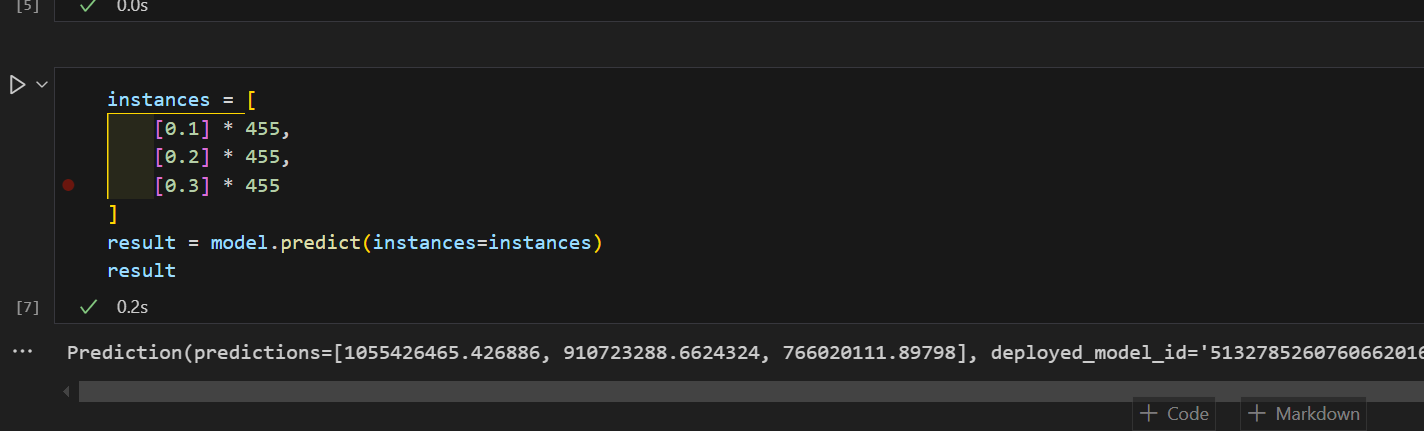


Nos conectamos al modelo con la ruta que vemos en el output del código anterior



Generamos el array que le enviaremos al modelo para predecir los datos.

Por como está entrenado este modelo requiere de 455 features, asi que hacemos una prueba rápida



Con eso acabaríamos la prueba de GCP y la carga de modelos.

Es importante que dependiendo de cómo hayamos entrenado al modelo, si es de regresión, clasificación, cuantas columnas diferentes necesite para predecir…

Necesitaremos una diferente cantidad de valores para generar una predicción