



IC-6821 Diseño de Software

Detalles de correcta instalación

Profesora:

Ericka Solano Fernández

Integrantes:

Julian López Mata: 2022207831

Javier Rojas Rojas: 2022199282

Kendell Garbanzo Calvo: 2022123716

II Semestre, 05 de Octubre del 2024

Hosteo de API y MySQL en Google Cloud

Paso 1. Instalar Cloud SDK y crear Instancia de Cloud SQL (MySQL):

Primero tendrás que acceder al siguiente link: <https://cloud.google.com/sdk/?hl=en> Sigue todos los pasos de la instalación.

Luego crearemos la base de datos yendo a <https://console.cloud.google.com/>, crea tu cuenta y todo lo necesario.

Cloud SQL brinda \$300 en créditos gratis para uso de sus servicios, estos se irán gastando conforme al uso de la base de datos.

Google CloudMy First Project

Buscar (/) recursos, documentos, productos y más

Buscar

SQL

← Crear una instancia

Elige tu motor de base de datos

MySQL

Versiones: 8.4, 8.0, 5.7, 5.6

Elegir MySQL

PostgreSQL

Versiones: 17, 16, 15, 14, 13, 12, 11, 10, 9, 6

Elegir PostgreSQL

SQL Server

Versiones: 2022, 2019, 2017

Elegir SQL Server

¿Quieres obtener más información sobre los motores de base de datos de Cloud SQL?

Más información

Presentamos AlloyDB para PostgreSQL

PostgreSQL más lo mejor de la nube: procesamiento y almacenamiento con escalamiento horizontal, almacenamiento en caché inteligente y administración potenciada por IA y AA. Elige AlloyDB para cargas de trabajo transaccionales 4 veces más rápidas, consultas analíticas y vectoriales más eficientes y ANS del 99.99% líder en la industria, con mantenimiento incluido.

ELEGIR ALLOYDB

MÁS INFORMACIÓN

Google CloudMy First Project

Buscar (/) recursos, documentos, productos y más

Buscar

← Crea una instancia de MySQL

COMPARA LOS AJUSTES PREDETERMINADOS DE LAS EDICIONES

Información de la instancia

Versión de la base de datos *MySQL 8.0

MOSTRAR VERSIONES SECUNDARIAS

ID de instancia *prueba

Usa letras minúsculas, números y guiones. Comienza con una letra.

Contraseña *GENERAR

Establece una contraseña para el usuario raíz. Más información

☐ Sin contraseña

POLÍTICA DE CONTRASEÑAS

Elige la región y la disponibilidad zonal

Para obtener un mejor rendimiento, mantén tus datos cerca de los servicios que los necesitan. La región es permanente, mientras que la zona se puede cambiar en cualquier momento.

Regiónus-central1 (Iowa)

Edición de Cloud SQLEnterprise

Regiónus-central1 (Iowa)

Versión de la base de datosMySQL 8.0

CPU virtuales4 CPU virtual(es)

RAM16 GB

Caché de datosInhabilitada

Almacenamiento100 GB

ConexionesIP pública

Copia de seguridadAutomatizada

DisponibilidadZona única

Recuperación de un momento determinadoHabilitada

Capacidad de procesamiento de la red (MB/s)1,000 de 1,000

IOPS

Lectura: 3,000 de 15,000

Escritura: 3,000 de 15,000

Capacidad de procesamiento del disco (MB/s)

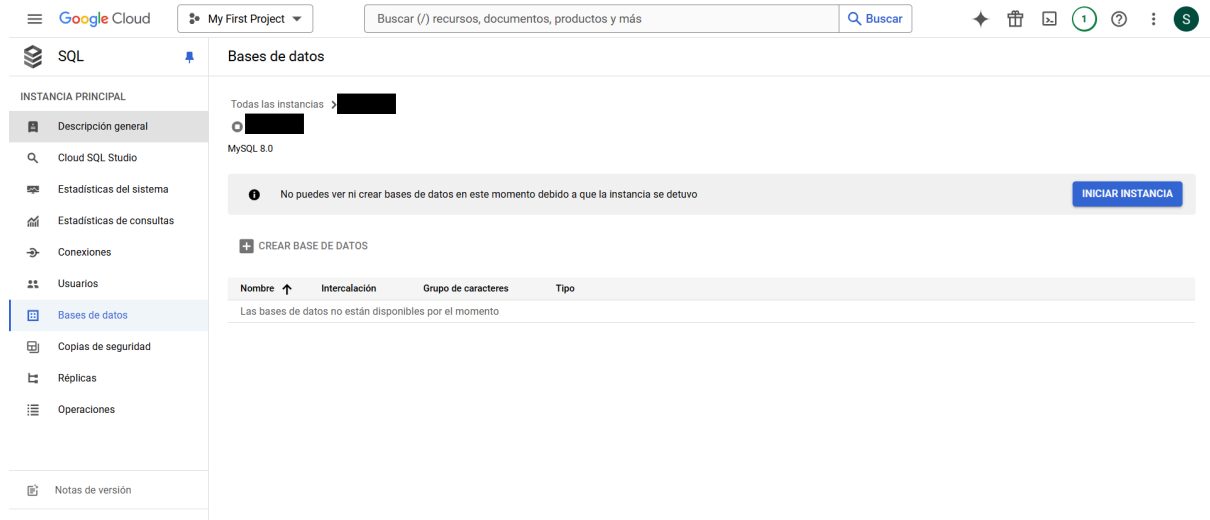
Lectura: 48.0 de 240.0

Escritura: 48.0 de 240.0

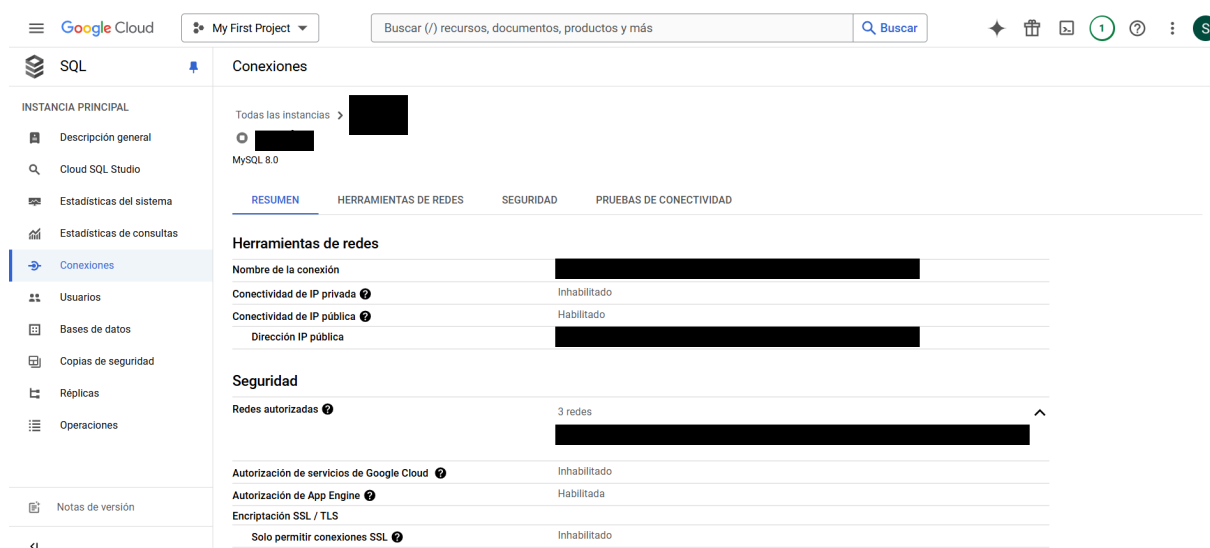
Precio estimado (sin descuentos)

Estos elementos representan únicamente los recursos de procesamiento, memoria y almacenamiento de Cloud SQL y

Luego, en la pestaña de tu instancia Cloud SQL crearás la base de datos.



En la pestaña de **Conexiones** encontrarás información importante de la base de datos.



Es importante que guardes el nombre de la conexión y el nombre de tu proyecto(**texto que viene antes de los dos puntos en el nombre de la conexión**), ya que los usaremos luego.

En la pestaña de usuarios también podrás modificar los datos de los usuarios de tu base de datos, o puedes usar los que definiste como predeterminados.

Paso 2. Correr script de deployment y configurar Cloud Run

Cloud Run es un servicio que nos permitirá crear un REST API basándose en el Dockerfile que tenemos en el repositorio, este Dockerfile generará un servidor de Express con las rutas que están declaradas en `scr/index.ts`.

En la carpeta principal del repositorio, encontrarás un script de bash llamado `deploy.sh`, en este script se guardan las variables de entorno para conectarnos con nuestra base de datos Cloud SQL

Estas variables de entorno las tendrás que configurar de la siguiente manera

```
GOOGLE_PROJECT_ID=tu-nombre-de-proyecto

CLOUD_RUN_SERVICE=nombre-que-le-vas-a-dar-a-tu-api

INSTANCE_CONNECTION_NAME=tu-nombre-de-conexion

DB_USER=tu-usuario

DB_PASSWORD=tu-contraseña

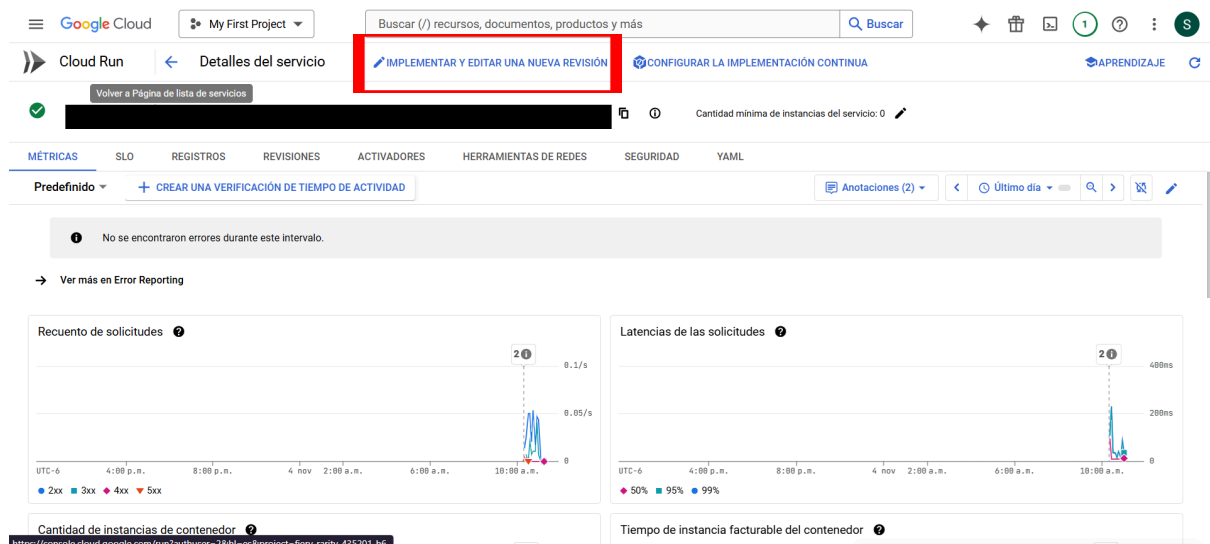
DB_NAME=nombre-de-tu-base-de-datos
```

Luego correrás el siguiente comando: `"gcloud auth login"`, e iniciarás sesión con la cuenta de Google donde creaste la instancia de SQL.

Aquí deberás encender tu instancia de Cloud SQL.

Luego correrás el script de la siguiente manera: `"./deploy.sh"`. Esto generará tu instancia de Cloud Run con el API.

Para que tu API funcione correctamente, deberás configurar tu instancia de Cloud Run, enlazando con tu instancia de Cloud SQL.



Google Cloud | My First Project | Buscar (/) recursos, documentos, productos y más | Buscar

Cloud Run | Implementar la revisión de [Redacted] | MOSTRAR LÍNEA DE COMANDOS | APRENDIZAJE

Ajuste de escala automático de la revisión

Número mínimo de instancias * 0 | Número máximo de instancias * 3

☒ Aumento de CPU de inicio

Conexiones de Cloud SQL

Instancia de Cloud SQL 1

+ AGREGAR CONEXIÓN

☒ Aplicar esta revisión inmediatamente

IMPLEMENTAR | CANCELAR

Ahí seleccionas tu instancia de Cloud SQL. Esto debería hacer que tu API quede configurada correctamente con tu base de datos.

El API usando un ORM se encarga de crear las tablas respectivas en la base de datos, junto con ciertas filas necesarias, como los estados y presentaciones.

Solo necesitarás poblar las tablas de Farmacia, y Medicamentos por tu cuenta, utilizando los endpoints respectivos.

JSON para crear farmacias:

```
{  
  
  "nombre": "Sucre",  
  
  "direccion": "500m del parque de Sucre",  
  
  "telefono": "88888345",  
  
  "email": "sucre@gmail.com"  
}
```

JSON para crear medicamentos:

```
{  
  
  "nombre": "Enantyum Pastilla",  
  
  "descripcion": "Medicamento para el dolor y la fiebre",  
  
  "presentacion": { "id": 1 },  
  
  "precio": 12.5,  
  
  "puntosPorCompra": 10,  
  
  "puntosParaCanje": 100,  
  
  "urlImagen":  
  "https://farmacias.fischelenlinea.com/Images/Upload/1/1/90bb7b946551833_Enantyum_Comprimido  
s_200_400x350.jpg",  
  
  "estadoPromocion": true  
}
```

Donde los ids de las presentaciones son 1: Pastilla, 2: Boli, 3. Botella

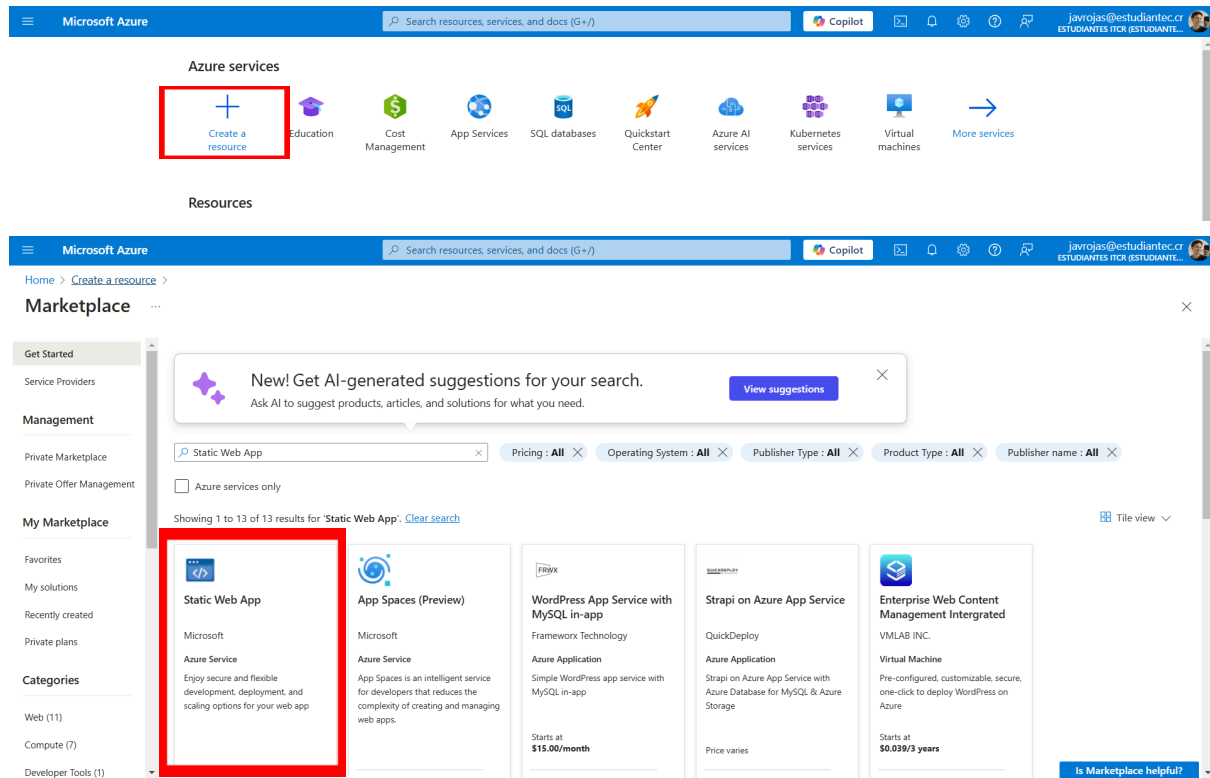
IMPORTANTE

Ya que tenemos nuestro link del API, en `snupe/src` deberás reemplazar el link que viene predeterminado, por el de tu API en todos los request HTTP.

Hosteo de Aplicación de React en Azure

En este caso utilizamos el servicio de Static Web Apps que tiene un plan gratuito y es más que suficiente.

Iremos a <https://portal.azure.com/#home>



Acá es importante que seleccionemos el plan gratis, y que llenemos el resto con los datos de nuestra cuenta.

[Home](#) > [Create a resource](#) > [Marketplace](#) > [Static Web App](#)

Create Static Web App

Basics Deployment configuration Advanced Tags Review + create

App Service Static Web Apps is a streamlined, highly efficient solution to take your static app from source code to global high availability. Pre-rendered content is distributed globally with no web servers required. [Learn more](#)

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure for Students

Resource Group * Snuple

[Create new](#)

Static Web App details

Name * Prueba

Hosting plan

The hosting plan dictates your bandwidth, custom domain, storage, and other available features. [Compare plans](#)

Plan type
☒ Free: For hobby or personal projects
☐ Standard: For general purpose production apps

Acá seleccionamos el repositorio donde tengamos el código fuente, que sea un proyecto de React, y en el output location pondremos “build”.

GitHub account

javialroro [Change account](#) ⓘ

i If you can't find an organization or repository, you might need to enable additional permissions on GitHub. You must have write access to your chosen repository to deploy with GitHub Actions. ×

Organization *	<input type="text" value="javialroro"/>
Repository *	<input type="text" value="SnupieDiseno"/>
Branch *	<input type="text" value="main"/>

Build Details

Enter values to create a GitHub Actions workflow file for build and release. You can modify the workflow file later in your GitHub repository.

Build Presets	<input type="text" value="React"/>
i These fields will reflect the app type's default project structure. Change the values to suit your app. Learn more	
App location * ⓘ	<input style="border: 1px solid #ccc;" type="text" value="/"/>
Api location ⓘ	<input type="text" value="e.g. 'api', 'functions', etc..."/>
Output location ⓘ	<input type="text" value="build"/>

Workflow configuration

Click the button below to preview what the GitHub Actions workflow file will look like before setting up continuous deployment.

[Preview workflow file](#)

[Review + create](#)

[< Previous](#)

[Next : Deployment configuration >](#)

Create Static Web App ...

Basics Deployment configuration Advanced Tags **Review + create**

Summary



Static Web App
by Microsoft

Details

Subscription	650ba6b4-425e-49cc-8e12-39054328a05f
Resource Group	Snupie
Name	Prueba
Region	centralus
SKU	Free
Repository	https://github.com/javialroro/SnupieDiseno
Branch	main
App location	/
API location	
Output location	build
Deployment authorization policy	Deployment token

Create

< Previous

Next >

[Download a template for automation](#)

Luego iremos al dashboard de nuestra Static Web App y tendremos el link de nuestra aplicación de React.