

# Entrega 4 – Aplicación Elástica En Paas Migración De Una Aplicación Web A Un Plataforma Como Servicio En La Nube Pública

Gina Eveling Posada , Martin Daniel Rincón, Juan Camilo Muñoz, Felipe Serrano

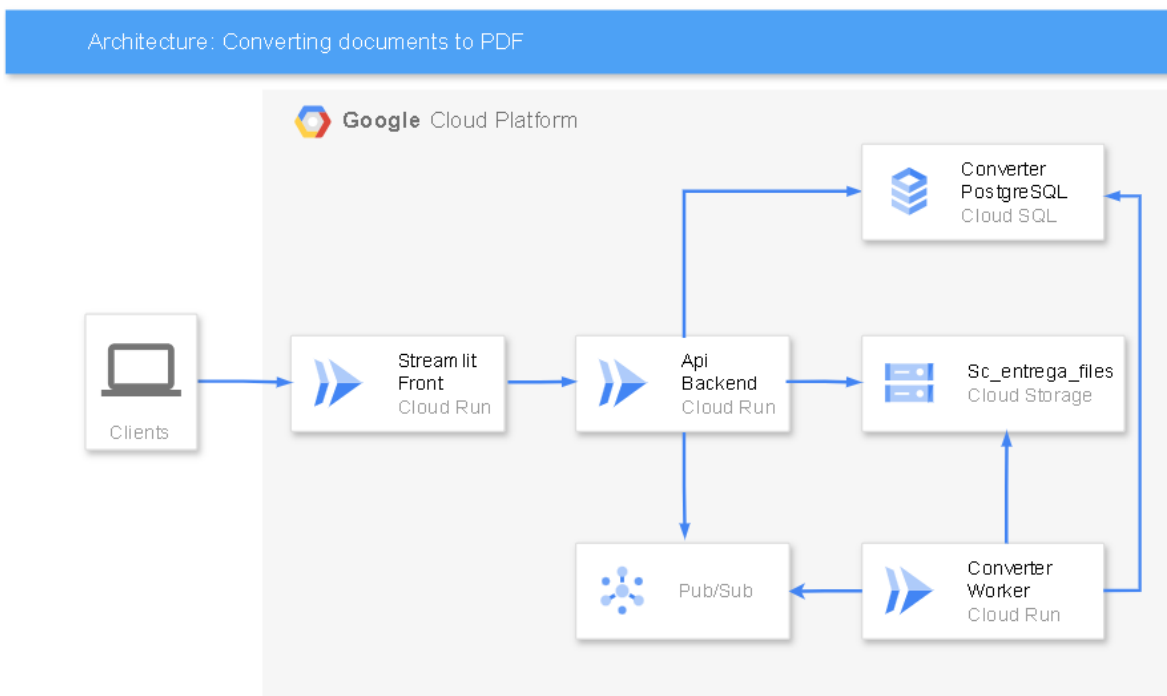
MINE semestre 202410

Universidad de los Andes, Bogotá, Colombia

{g.posadas, md.rincon, jc.munozc12, ff.serrano42}@uniandes.edu.co

Fecha de presentación: abril 09 de 2024

## Arquitectura propuesta en GCP



La aplicación ha sido desarrollada implementando 3 componentes de Cloud Run, un servicio Cloud SQL, un Cloud Storage para el almacenamiento de archivos y un servicio de Pub/Sub para gestionar el procesamiento desacoplado de la conversión de los archivos.

Estos componentes interactúan en la nube de GCP, dentro de la misma región/zona y red para poder desplegar y cumplir con la funcionalidad requerida para convertir archivos de diferentes (docx, pptx, xlsx, odt) formatos a PDF.

### 1. Frontend:

- Imagen de Docker desplegada en Cloud Run
- Desarrollado utilizando Streamlit Web API para garantizar una presentación ágil y eficiente para el usuario.
- Facilita la interacción con los servicios del API Rest del Backend a través del protocolo HTTP.

### 2. Api-taller:

- Imagen de Docker desplegada en Cloud Run
- Implementado en el framework FastApi, proporcionando todas las funcionalidades de negocio.
- Incluye características como registro y login de usuarios mediante JSON Web Tokens (JWT).
- Gestiona la carga y descarga de archivos, así como la gestión de mensajes en la cola de conversión de archivos.

### 3. Converter (Base de Datos):

- Para esta entrega se está utilizando el servicio SQL ofrecido por la plataforma GCP (Cloud SQL), como motor de db se utiliza PostgreSQL para alojar las tablas relacionadas con usuarios y las tareas de conversión de documentos.

### 4. Pub/Sub:

- Cada solicitud de conversión genera un mensaje en la cola del PUB/SUB la cual se encuentra sobre su propio contenedor dentro de la maquina denominada Worker

### 5. Worker (Consumer):

- Imagen de Docker desplegada en Cloud Run
- Programa en python, ejecutado en su propio contenedor, el cual Procesa los mensajes puestos en el PUB/SUB para la conversión de archivos de manera asíncrona. Recupera la ruta desde la que debe leer el archivo desde la base de datos y hace llamado al comando en Linux para el libreoffice, y así convertir utilizando este componente gratuito un documento a formato pdf

```
subprocess.call(['soffice', '--headless', '--convert-to', 'pdf', '--outdir', upload_folder,
source_file_without_first_slash])
```

- Por último guarda la ruta en la que se almaceno el archivo en formato pdf
- Este contenedor tiene como base una imagen sobre la cual se instaló **Libreoffice lo que lo hace más escalable, ya que libreoffice hace parte de la imagen y no de la máquina.**

### 6. Cloud Storage:

- Bucket de Cloud Storage para alojar los archivos cargados y los PDFs procesados.

## Despliegue en GCP

### 1. Base de Datos (Cloud SQL) - Configuración de Cloud SQL:

#### 1. Crear una Instancia de Cloud SQL:

- Ve a SQL en la consola de GCP.
- Haz clic en "Crear instancia".
- Selecciona "PostgreSQL".
- Configura el ID de la instancia, la contraseña del usuario root, la región y zona.
- Configura la máquina y el almacenamiento según tus necesidades.

- Haz clic en "Crear".

## **2. Pub/Sub - Configuración de Pub/Sub:**

### **1. Crear un Tópico:**

- Ve a Pub/Sub en la consola de GCP.
- Haz clic en "Crear tópico".
- Asigna un nombre al tópico.
- Haz clic en "Crear".

### **2. Crear una Suscripción:**

- Ve a la pestaña "Suscripciones".
- Haz clic en "Crear suscripción".
- Asigna un nombre a la suscripción y selecciona el tópico creado.
- Configura las opciones de entrega (pull) y el formato de entrega (JSON, etc.).
- Haz clic en "Crear".

## **3. Cloud Storage - Configuración de Cloud Storage:**

### **1. Crear un Bucket:**

- Ve a Cloud Storage en la consola de GCP.
- Haz clic en "Crear bucket".
- Asigna un nombre único al bucket.
- Selecciona la ubicación (regional).
- Selecciona el tipo de almacenamiento (Standard).
- Configura los controles de acceso (Uniforme).
- Haz clic en "Crear".

## **4. Worker (Consumer en Cloud Run) - Configuración del Worker en Cloud Run:**

### **1. Crear la Imagen de Docker**

### **2. Construir y Subir la Imagen**

### **3. Desplegar en Cloud Run**

- Ve a Cloud Run en la consola de GCP.
- Haz clic en "Crear servicio".
- Selecciona "Desplegar una revisión desde una imagen existente".
- Introduce la URL de la imagen ([gcr.io/\[PROJECT-ID\]/worker-consumer](https://gcr.io/[PROJECT-ID]/worker-consumer)).
- Configura la memoria y el número de instancias.
- Configura las variables para que se conecte a la BD y al Pub/Sub

- Haz clic en "Crear".

#### **5. Backend (API-Taller en Cloud Run) - Configuración del Backend en Cloud Run:**

1. **Crear la Imagen de Docker**
2. **Construir y Subir la Imagen**
3. **Desplegar en Cloud Run:**
  - Ve a Cloud Run en la consola de GCP.
  - Haz clic en "Crear servicio".
  - Selecciona "Desplegar una revisión desde una imagen existente".
  - Introduce la URL de la imagen (**`gcr.io/[PROJECT-ID]/api-taller-backend`**).
  - Configura las variables para que se conecte a la BD, al Bucket y al Pub/Sub
  - Configura la memoria y el número de instancias.
  - Configura el servicio para que sea público.
  - Haz clic en "Crear".

#### **6. Frontend (Streamlit en Cloud Run) - Configuración del Frontend en Cloud Run:**

1. **Crear la Imagen de Docker**
2. **Construir y Subir la Imagen en Docker**
3. **Desplegar en Cloud Run:**
  - Ve a Cloud Run en la consola de GCP.
  - Haz clic en "Crear servicio".
  - Selecciona "Desplegar una revisión desde una imagen existente".
  - Introduce la URL de la imagen (**`gcr.io/[PROJECT-ID]/streamlit-frontend`**).
  - Configura la memoria y el número de instancias.
  - Configura el servicio para que sea público.
  - Configura las variables para que se conecte al endpoint del API
  - Haz clic en "Crear".

#### **7. Reglas de Firewall**

1. **Acceder a la Consola de GCP:**
  - Ve a la consola de Google Cloud Platform (<https://console.cloud.google.com/>).
2. **Navegar a VPC Network:**
  - En el menú de navegación, ve a "VPC network" y luego selecciona "Firewall".
3. **Crear una Nueva Regla de Firewall:**
  - Haz clic en "Crear regla de firewall".

## 2. Configurar la Regla de Firewall

### Paso a Paso para Configurar la Regla de Firewall:

1. **Nombre y Descripción:**
  - Asigna un nombre a la regla de firewall (por ejemplo, **allow-internal-traffic**).
  - Opcionalmente, proporciona una descripción para la regla.
2. **Red:**
  - Selecciona la red a la que se aplicará la regla (por ejemplo, **default**).
3. **Prioridad:**
  - Establece la prioridad de la regla. Las reglas con un número más bajo tienen mayor prioridad (por ejemplo, **1000**).
4. **Dirección del Tráfico:**
  - Selecciona "Entrada" para permitir el tráfico entrante hacia tus instancias.
5. **Acción en Coincidencia:**
  - Selecciona "Permitir" para permitir el tráfico que coincida con esta regla.
6. **Objetivo de la Regla:**
  - Selecciona las instancias a las que se aplicará la regla (por ejemplo, "Todas las instancias en la red").
7. **Filtros de Fuente:**
  - Establece los rangos de IP de origen que permitirás (por ejemplo, **0.0.0.0/0** para permitir tráfico desde cualquier dirección IP).
  - Alternativamente, puedes especificar rangos IP específicos o etiquetas de red.
8. **Protocolos y Puertos:**
  - Permitir el tráfico a la base de datos en el puerto 5432 (PostgreSQL):
9. **Crear la Regla:**
  - Revisa la configuración y haz clic en "Crear" para crear la regla de firewall.

### Configuración de Firewall para Componentes Específicos

1. **Para Google Cloud SQL:**
  - **Puertos:** tcp:5432 (PostgreSQL)
  - **Fuente:** Direcciones IP de las instancias que necesiten acceder a la base de datos.
2. **Para Google Cloud Run:**
  - **Puertos:** tcp:5001 y 8501
  - **Fuente:** 0.0.0.0/0 para permitir tráfico público o direcciones IP específicas si se quiere restringir el acceso.

# Despliegue de los componentes por tipo

## Cloud Run

Google Cloud

devcloud

Search (/) for resources, docs, products, and more

Search

Cloud Run

Services

CREATE SERVICE

CREATE JOB

MANAGE CUSTOM DOMAINS

RELEASE NOTES

SERVICES

JOB

Filter services

Name	Req/Sec	Region	Authentication	Ingress	Recommendation	Last deployed	Deployed by
hse	0.01	us-west1	Allow unauthenticated	All		16 hours ago	jucamucos@gmail.com
conector	0	us-west1	Allow unauthenticated	All		45 minutes ago	jucamucos@gmail.com
front	0.02	us-west1	Allow unauthenticated	All		16 hours ago	jucamucos@gmail.com

## Cloud SQL

Google Cloud

devcloud

Search (/) for resources, docs, products, and more

Search

All instances

conector

Explorer

conector

public (default)

Tables

Documents

Views

Functions

Procedures

Sequences

Types

Extensions

information\_schema

pg\_catalog

SQL

SELECT \* FROM public "Documents" LIMIT 1000;

RESULTS

id_documento	id_usuario	seccion_documento	seccion_usuario	seccion_usuario	pdf_documento
2	3	Actualizaci...	https://storage.cloud.google.com/sc_entrega3_files/Actualizaci...	any	https://storage.cloud.google.com/sc_entrega3_files/Actualizaci...
3	3	pe.xlsx	https://storage.cloud.google.com/sc_entrega3_files/pe.xlsx	any	https://storage.cloud.google.com/sc_entrega3_files/pe.pdf

## Cloud Storage

Google Cloud

devcloud

bucket

Search

Cloud Storage

Bucket details

GO TO PATH

REFRESH

LEARN

Buckets

Monitoring

Settings

sc\_entrega3\_files

Location: us-west1 (Oregon)

Storage class: Standard

Public access: Not public

Protection: Soft Delete

OBJECTS

CONFIGURATION

PERMISSIONS

PROTECTION

LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

OPERATIONS

Folder browser

sc\_entrega3\_files

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER

TRANSFER DATA

MANAGE HOLDS

EDIT RETENTION

DOWNLOAD

DELETE

Filter by name prefix only

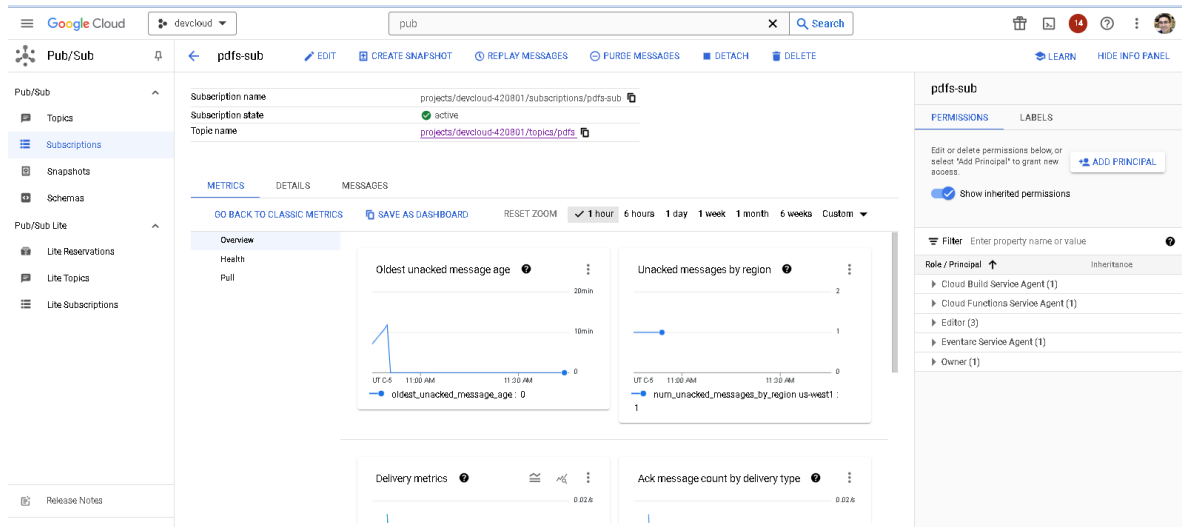
Filter

Filter objects and folders

Show Live objects only

Name	Size	Type	Created	Storage class	Last modified	Public acc
Actualización Reportes I04.pdf	26.6 KB	application/pdf	May 25, 2024, 7:18:22 PM	Standard	May 25, 2024, 7:18:22 PM	Not public
Actualización Reportes I04.xlsx	9.8 KB	text/plain	May 25, 2024, 7:16:55 PM	Standard	May 25, 2024, 7:16:55 PM	Not public
Laboratorio2.docx	12.9 KB	text/plain	May 25, 2024, 7:05:00 PM	Standard	May 25, 2024, 7:05:00 PM	Not public
Laboratorio2.pdf	1.2 KB	application/pdf	May 25, 2024, 7:10:21 PM	Standard	May 25, 2024, 7:10:21 PM	Not public
pe.pdf	29 KB	application/pdf	May 26, 2024, 10:49:32 AM	Standard	May 26, 2024, 10:49:32 AM	Not public
pe.xlsx	10.8 KB	text/plain	May 26, 2024, 10:36:47 AM	Standard	May 26, 2024, 10:36:47 AM	Not public

## Pub/Sub



## Consideraciones para la Escalabilidad:

Para asegurar que la aplicación pueda escalar eficientemente y manejar cientos de usuarios concurrentes, se deben tener en cuenta las siguientes consideraciones:

- **Autoscaling:** Configurar políticas de escalado automático en Cloud Run y Cloud SQL para manejar incrementos en la demanda sin intervención manual.
- **Load Balancing:** Implementar balanceadores de carga para distribuir el tráfico entrante de manera equitativa entre las instancias de servicio.
- **Optimización de Base de Datos:** Revisar y optimizar las consultas a la base de datos. Considerar el uso de índices y particionamiento de tablas para mejorar el rendimiento.
- **Almacenamiento en Caché:** Implementar mecanismos de caché utilizando servicios como Memorystore para reducir la carga en la base de datos y mejorar la velocidad de respuesta.
- **Monitoreo y Alerta:** Utilizar Cloud Monitoring para rastrear el rendimiento de los servicios y configurar alertas para detectar y reaccionar a problemas antes de que afecten a los usuarios.

## Limitaciones del Desarrollo Realizado

Limitaciones Identificadas:

- **Dependencia de Servicios PaaS:** La aplicación depende significativamente de servicios PaaS, lo que puede limitar la flexibilidad en la personalización de ciertos aspectos del entorno de ejecución.
- **Costos Operativos:** Aunque los servicios en la nube proporcionan escalabilidad, los costos pueden aumentar considerablemente con el incremento en la cantidad de usuarios concurrentes.
- **Procesamiento Asíncrono:** Tuvimos el inconveniente en el despliegue del converter-worker en Cloud Run, el cual no quedaba activo, pero si procesaba cuando se hace Deploy al servicio. Quizá debimos explorar más a fondo la posibilidad de usar Cloud Functions para ese componente para que le podamos configurar un trigger de eventos para el encolamiento en el Pub/Sub.