

## Entregable – Taller Microservicios V4

Brayan Estiven Salcedo Suarez

David Alejandro Calderon Pineda

### 1. Despliegue del microservicio Places.

En esta sección se documenta el proceso de implementación del nuevo microservicio Places, donde se configuró una instancia VM independiente con Ubuntu 18.04 y se desplegó el servicio usando FastAPI. La imagen adjunta muestra la estructura del microservicio conectado a su propia base de datos MongoDB, cumpliendo con el patrón "DB por servicio". Se evidencia la correcta configuración de los endpoints CRUD para gestionar lugares.

```
is deprecated. A suggested replacement is 'projects/ubuntu-os-cloud/global/images/ubuntu-2004-robot-v20250508'

NAME: msd-places-db
TYPE: compute.v1.instance
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: msd-places-ms
TYPE: compute.v1.instance
STATE: COMPLETED
ERRORS: []
INTENT:

NAME: msd-service-db
TYPE: compute.v1.firewall
STATE: COMPLETED
ERRORS: []
dav_calderonpi@cloudshell:~/ISIS2503-Microservices-AppDjango (isis2503-talleres-451901) $
```

A continuación se ve la instancia desplegada de places

filter <small>filter property name or value</small>								
<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	✓	<a href="#">msd-kong-instance</a>	us-central1-a			10.128.0.81 ( <a href="#">nic0</a> )	34.44.164.128 ( <a href="#">nic0</a> )	SSH ▾ ⋮
<input type="checkbox"/>	✓	<a href="#">msd-measurements-db</a>	us-central1-a			10.128.0.83 ( <a href="#">nic0</a> )		SSH ▾ ⋮
<input type="checkbox"/>	✓	<a href="#">msd-measurements-ms</a>	us-central1-a			10.128.0.85 ( <a href="#">nic0</a> )	34.172.48.152 ( <a href="#">nic0</a> )	SSH ▾ ⋮
<input type="checkbox"/>	✓	<a href="#">msd-places-db</a>	us-central1-a			10.128.0.86 ( <a href="#">nic0</a> )		SSH ▾ ⋮
<input type="checkbox"/>	✓	<a href="#">msd-places-ms</a>	us-central1-a			10.128.0.87 ( <a href="#">nic0</a> )	104.198.233.225 ( <a href="#">nic0</a> )	SSH ▾ ⋮
<input type="checkbox"/>	✓	<a href="#">msd-variables-db</a>	us-central1-a			10.128.0.82 ( <a href="#">nic0</a> )		SSH ▾ ⋮
<input type="checkbox"/>	✓	<a href="#">msd-variables-ms</a>	us-central1-a			10.128.0.84 ( <a href="#">nic0</a> )	35.239.80.43 ( <a href="#">nic0</a> )	SSH ▾ ⋮

Related actions

### Ejecución del microservicio:

En esta sección podemos ver que la aplicación se ejecuto correctamente

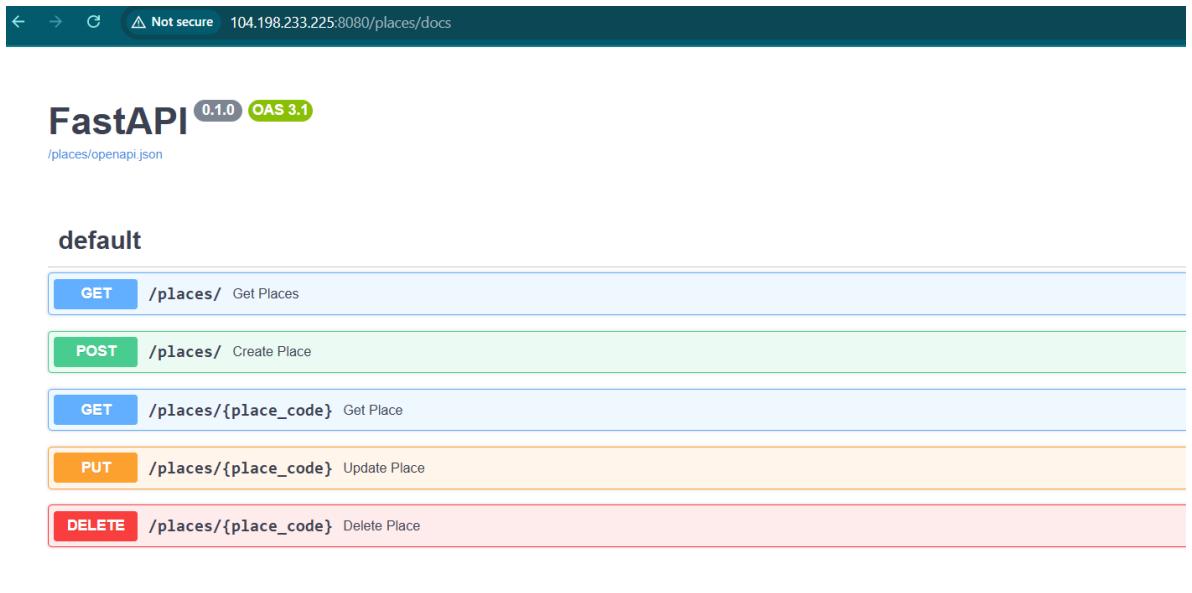
```

dav_calderonpi@msd-places-ms:~$ cd /home/labs/ISIS2503-Microservices-AppDjango/places
dav_calderonpi@msd-places-ms:/home/labs/ISIS2503-Microservices-AppDjango/places$ sudo python3.12 main.py
/home/labs/ISIS2503-Microservices-AppDjango/places/main.py:14: DeprecationWarning:
  on_event is deprecated, use lifespan event handlers instead.

  Read more about it in the
  [FastAPI docs for Lifespan Events](https://fastapi.tiangolo.com/advanced/events/).

@app.on_event("startup")
INFO: Started server process [3343]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)

```



## 2. Modificación de configuración de Kong

Aquí se realizó la actualización del API Gateway Kong para integrar el nuevo microservicio Places al ecosistema existente. La imagen muestra la configuración del archivo YAML donde se agregó el nuevo upstream (places-ms) y se definieron las rutas correspondientes, permitiendo que el gateway enrute correctamente las peticiones al nuevo servicio mientras mantiene el tráfico de los servicios existentes.

```

- name: measurements_service
  host: measurements_upstream
  protocol: http
  routes:
    - name: measurements
      paths:
        - /measurements
      strip_path: false

- name: places_service
  host: places_upstream
  protocol: http
  routes:
    - name: places
      paths:
        - /places
      strip_path: true

upstreams:
- name: variables_upstream
  targets:
    - target: 10.128.0.84:8080
      weight: 100

- name: measurements_upstream
  targets:
    - target: 10.128.0.85:8080
      weight: 100

- name: places_upstream
  targets:
    - target: 10.128.0.87:8080
      weight: 100

```

### 3. Modificación de microservicio Measurements

Esta parte demuestra las adaptaciones necesarias en el servicio Measurements para validar la existencia de lugares antes de crear mediciones. Las imágenes muestran el nuevo código que implementa llamadas HTTP al servicio Places, verificando la integridad referencial distribuida.

```

PLACES_HOST = os.environ.get('PLACES_HOST', '10.128.0.87')
PLACES_PORT = os.environ.get('PLACES_PORT', '8080')
PATH_PLACES = f"http://{PLACES_HOST}:{PLACES_PORT}/places/"

"monitoring/settings.py" 144L, 4079C

```

Se añade la función que verifica el lugar:

```

def check_place(data):
    r = requests.get(settings.PATH_PLACES, headers={"Accept": "application/json"})
    places = r.json()
    for place in places:
        if data["place"] == place["code"]:
            return True
    return False

```

se modifican las funciones de measurement para usar la función

```
def MeasurementCreate(request):
    if request.method == 'POST':
        data = request.body.decode('utf-8')
        data_json = json.loads(data)
        if check_variable(data_json) and check_place(data_json):
            measurement = Measurement()
            measurement.variable = data_json['variable']
            measurement.value = data_json['value']
            measurement.unit = data_json['unit']
            measurement.place = data_json['place']
            measurement.save()
            return HttpResponse("successfully created measurement")
        else:
            return HttpResponse("unsuccessfully created measurement. Variable or place does not exist")

def MeasurementsCreate(request):
    if request.method == 'POST':
        data = request.body.decode('utf-8')
        data_json = json.loads(data)
        measurement_list = []
        for measurement in data_json:
            if check_variable(measurement) and check_place(measurement):
                db_measurement = Measurement()
                db_measurement.variable = measurement['variable']
                db_measurement.value = measurement['value']
                db_measurement.unit = measurement['unit']
                db_measurement.place = measurement['place']
                measurement_list.append(db_measurement)
            else:
                return HttpResponse("unsuccessfully created measurement. Variable or place does not exist")

        Measurement.objects.bulk_create(measurement_list)
```

#### 4. Creación de Lugares de Prueba

Mediante la interfaz Swagger UI autogenerada por FastAPI, se crearon lugares de prueba para validar el funcionamiento del microservicio. La captura muestra ejemplos de creación exitosa de lugares con diferentes capacidades y tipos (aula, laboratorio, etc.), demostrando la correcta persistencia en MongoDB y la disponibilidad de los datos.

INGRESO A : <http://104.198.233.225:8080/places/docs>

Parameters

No parameters

Request body <sup>required</sup>

application/json

```
{
  "capacity": 50,
  "code": "HL515",
  "type": "classroom"
}
```

Execute Clear

## Parameters

No parameters

## Request body required

```
{  
  "capacity": 80,  
  "code": "ML213",  
  "type": "classroom"  
}
```

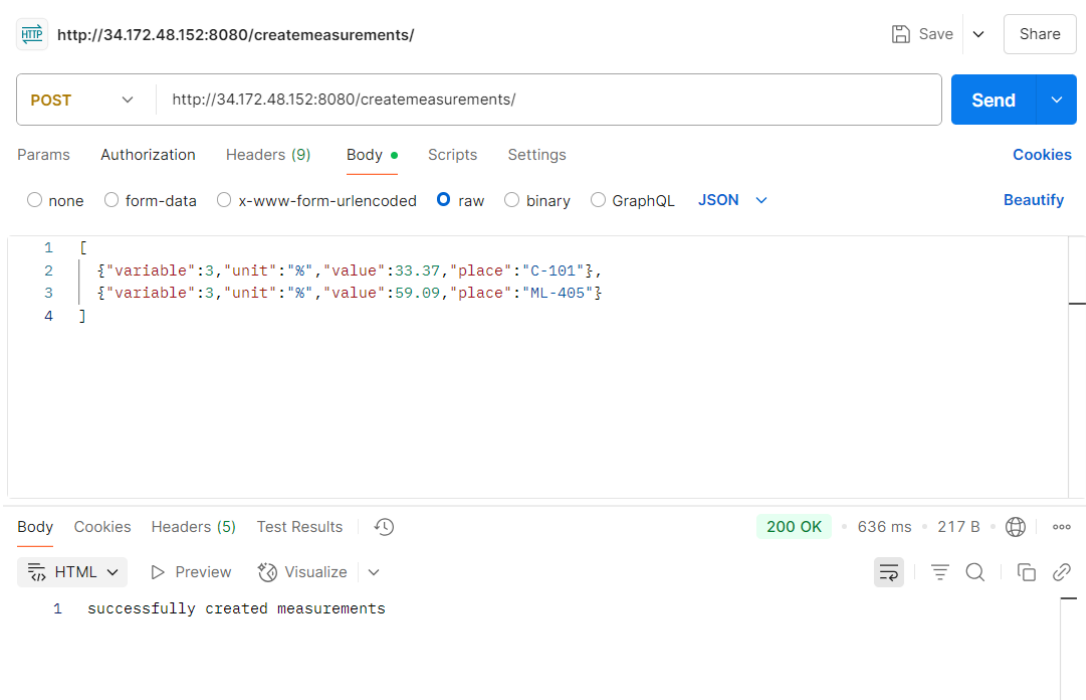
Execute

## Responses



## 5. Creación de *Cloud Function* para datos de oxígeno

En esta sección se documenta el desarrollo de una nueva función serverless que consume datos de oxígeno de una API externa. Las imágenes muestran la configuración en GCP Cloud Functions, incluyendo variables de entorno y el código Python que realiza las peticiones HTTP tanto al API externo como al servicio Measurements para almacenar los datos.



## Creación oxygen-consumption

oxygen-consumption Region: us-central1 URL: <https://oxygen-consumption-1008673593258.us-central1.run.app> Scaling: Auto (Min: 0)

Metrics SLOs Logs Revisions **Source** Triggers Networking Security YAML

Source Base image: Python 3.9 (Ubuntu 18 Full) Function entry point: hello\_http Edit source

main.py
requirements.txt

```

1 import requests
2 import json
3 import os
4
5 def hello_http(request):
6     data = requests.get(os.environ.get('API_PATH'), headers={"Accept": "application/json"})
7     json_data = data.json()
8
9     # Modificar el ID de la variable en el JSON
10    for measurement in json_data:
11        measurement["variable"] = 3 # Cambiar de 2 a 3 para usar la variable "Oxigeno"
12
13    response = requests.post(os.environ.get('MS_PATH'), json=json_data, headers={'Content-type': 'application/json', "charset": "utf-8"})
14    return "The function was successfully executed"
15
16

```

Source - oxygen-consumption - Cloud x oxygen-consumption-1008673593258.us-central1.run.app

The function was successfully executed

Posteriormente se realiza una verificación creación del *Cloud Scheduler* para ejecutar la función:

Google Cloud

ISIS2503-talleres

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

Search

60

Cloud Scheduler / Jobs

Jobs

CREATE JOB

REFRESH

FORCE RUN

EDIT

COPY

PAUSE

RESUME

DELETE

LEARN

SCHEDULER JOBS

APP ENGINE CRON JOBS

Filter

Filter jobs

	Name	Status of last execution	Region	State	Description	Frequency	Target	Last run	Next run	Last updated	Actions
<input type="checkbox"/>	<a href="#">api-consumption-scheduler</a>	Has not run yet	us-central1	Paused		*/* * * * * (America/Guayaquil)	URL : https://api-consumption-1008673593258.us-central1.run.app/			May 20, 2025, 4:20:22 PM	
<input checked="" type="checkbox"/>	<a href="#">oxygen-consumption-scheduler</a>	Success	us-central1	Enabled		*/* * * * * (America/Guayaquil)	URL : https://oxygen-consumption-1008673593258.us-central1.run.app/	May 20, 2025, 7:30:00 PM	May 20, 2025, 7:40:00 PM	May 20, 2025, 7:22:15 PM	

**Microservicio Measurements:**  
A continuación, se observa, en primer lugar, el resultado del microservicio Measurements accediendo al endpoint /measurements/ desde la dirección IP 104.198.233.225:8000. En esta

primera imagen se muestra una respuesta en formato JSON, en la que cada objeto representa una medición capturada por el sistema. Cada medición contiene atributos como id (identificador único), variable (ID del tipo de variable medida, como temperatura u oxígeno), value (valor numérico registrado), unit (unidad de medida como °C o ppm), place (lugar o sensor que reporta la medición), y dateTime (fecha y hora de captura).

[illegible]

### Agregación de datos con el tiempo:

Posteriormente, se observa una segunda imagen correspondiente a la IP 34.172.48.152:8080, donde el mismo endpoint devuelve una respuesta mucho más extensa, lo cual sugiere que este entorno ha recibido un mayor volumen de mediciones. Esto puede ser producto del funcionamiento del Cloud Scheduler y la Cloud Function que inyecta datos automáticamente, demostrando así la integración entre componentes del sistema. Ambas imágenes confirman que el microservicio está correctamente desplegado y operativo, permitiendo la visualización y gestión de mediciones en diferentes entornos de la arquitectura.

[illegible]



```
← → ↻ ⚠ Not secure 34.172.48.152:8080/measurements/ ☆ 🏠 🔍 ⋮
Pretty-print
163, "variable": 3, "value": 23.37, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:07:47.611Z"}, {"id": 164, "variable": 3, "value": 57.09, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:07:47.611Z"}, {"id": 165, "variable": 3, "value": 33.37, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:14:21.299Z"}, {"id": 166, "variable": 3, "value": 59.09, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 167, "variable": 3, "value": 15.2, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 168, "variable": 3, "value": 45.0, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 169, "variable": 3, "value": 34.75, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 170, "variable": 3, "value": 51.32, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 171, "variable": 3, "value": 71.2, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 172, "variable": 3, "value": 2.35, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 173, "variable": 3, "value": 25.03, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 174, "variable": 3, "value": 27.21, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 175, "variable": 3, "value": 51.32, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 176, "variable": 3, "value": 53.61, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 177, "variable": 3, "value": 65.88, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 178, "variable": 3, "value": 75.72, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 179, "variable": 3, "value": 81.17, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 180, "variable": 3, "value": 13.04, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 181, "variable": 3, "value": 7.77, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 182, "variable": 3, "value": 94.24, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 183, "variable": 3, "value": 51.87, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 184, "variable": 3, "value": 99.34, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:14:21.300Z"}, {"id": 185, "variable": 3, "value": 33.37, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 186, "variable": 3, "value": 59.09, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 187, "variable": 3, "value": 15.2, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 188, "variable": 3, "value": 45.0, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 189, "variable": 3, "value": 34.75, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 190, "variable": 3, "value": 51.32, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 191, "variable": 3, "value": 71.2, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 192, "variable": 3, "value": 2.35, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 193, "variable": 3, "value": 25.03, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 194, "variable": 3, "value": 27.21, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 195, "variable": 3, "value": 51.32, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 196, "variable": 3, "value": 53.61, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 197, "variable": 3, "value": 65.88, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 198, "variable": 3, "value": 75.72, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 199, "variable": 3, "value": 81.17, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 200, "variable": 3, "value": 13.04, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 201, "variable": 3, "value": 7.77, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 202, "variable": 3, "value": 94.24, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 203, "variable": 3, "value": 51.87, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 204, "variable": 3, "value": 99.34, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:30:04.097Z"}, {"id": 205, "variable": 3, "value": 33.37, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 206, "variable": 3, "value": 59.09, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 207, "variable": 3, "value": 15.2, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 208, "variable": 3, "value": 45.0, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 209, "variable": 3, "value": 34.75, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 210, "variable": 3, "value": 51.32, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 211, "variable": 3, "value": 71.2, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 212, "variable": 3, "value": 2.35, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 213, "variable": 3, "value": 25.03, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 214, "variable": 3, "value": 27.21, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 215, "variable": 3, "value": 51.32, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 216, "variable": 3, "value": 53.61, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 217, "variable": 3, "value": 65.88, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 218, "variable": 3, "value": 75.72, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 219, "variable": 3, "value": 81.17, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 220, "variable": 3, "value": 13.04, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 221, "variable": 3, "value": 7.77, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 222, "variable": 3, "value": 94.24, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 223, "variable": 3, "value": 51.87, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 224, "variable": 3, "value": 99.34, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T00:50:07.275Z"}, {"id": 225, "variable": 3, "value": 33.37, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 226, "variable": 3, "value": 59.09, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 227, "variable": 3, "value": 15.2, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 228, "variable": 3, "value": 45.0, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 229, "variable": 3, "value": 34.75, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 230, "variable": 3, "value": 51.32, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 231, "variable": 3, "value": 71.2, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 232, "variable": 3, "value": 2.35, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 233, "variable": 3, "value": 25.03, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 234, "variable": 3, "value": 27.21, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 235, "variable": 3, "value": 51.32, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 236, "variable": 3, "value": 53.61, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 237, "variable": 3, "value": 65.88, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 238, "variable": 3, "value": 75.72, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 239, "variable": 3, "value": 81.17, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 240, "variable": 3, "value": 13.04, "unit": "K", "place": "C-101", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 241, "variable": 3, "value": 7.77, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 242, "variable": 3, "value": 94.24, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 243, "variable": 3, "value": 51.87, "unit": "K", "place": "ML-405", "dateTime": "2025-05-21T01:00:07.351Z"}, {"id": 244, "variable": 3, "value": 99.34, "unit": "K", "place": "RGD-101", "dateTime": "2025-05-21T01:00:07.351Z"}]
```

## Microservicio Variables:

En tercer lugar vemos el funcionamiento de los distintos tipos de variables cada una con su id y nombre

```
← → ↻ ⚠ Not secure 104.198.233.225:8000/variables/
Pretty-print
[{"id": 1, "name": "Temperatura"}, {"id": 2, "name": "Temperatura"}, {"id": 3, "name": "Oxigeno"}]
```

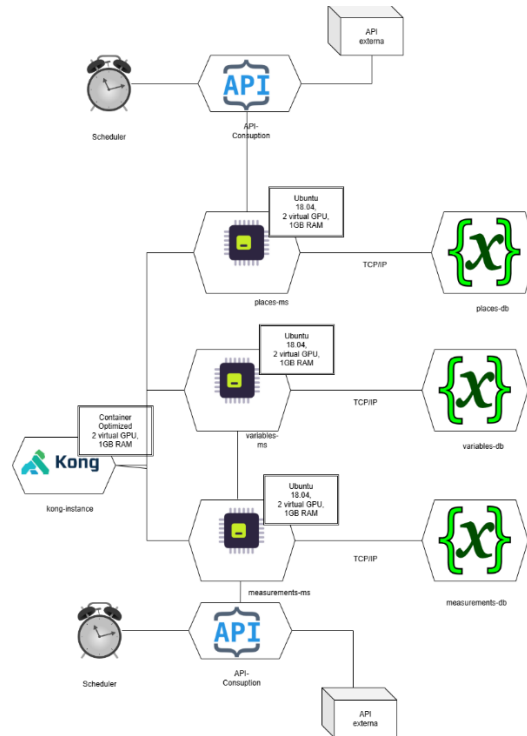
## Microservicio Places:

```
← → ↻ ⚠ Not secure 34.44.164.128:8080/places/
Pretty-print
{"places":[{"code":"ML515","capacity":50,"type":"classroom","id":"682c17a5a8e8964c8d5abba3"}, {"code":"ML213","capacity":80,"type":"classroom","id":"682c1864a8e8964c8d5abba5"}, {"code":"RGD202","capacity":40,"type":"classroom","id":"682c18a8a8e8964c8d5abba6"}, {"code":"LL105","capacity":35,"type":"classroom","id":"682c18caa8e8964c8d5abba7"}, {"code":"ML510","capacity":25,"type":"classroom","id":"682c31143d01536bb09e9e33"}, {"code":"R112","capacity":30,"type":"classroom","id":"682cb75088e5c2e95fddfde4"}, {"code":"LL112","capacity":40,"type":"classroom","id":"682cb75988e5c2e95fddfde5"}, {"code":"ML313","capacity":40,"type":"Laboratory","id":"682cb7a7b88e5c2e95fddfde6"}, {"code":"ML-405","capacity":80,"type":"classroom","id":"682cbdf588e5c2e95fddfde7"}, {"code":"RGD-101","capacity":30,"type":"classroom","id":"682cbdf188e5c2e95fddfde8"}, {"code":"C-101","capacity":80,"type":"classroom","id":"682cbdf588e5c2e95fddfde9"}]}
```

## Diagrama de despliegue

El diagrama fue actualizado incorporando un nuevo microservicio llamado places-ms, desplegado sobre una máquina virtual con Ubuntu 18.04, 2 vGPU y 1 GB de RAM, y conectado a su propia base de datos places-db, alojada en una máquina optimizada para contenedores. Además, se añadió una nueva Cloud Function que se comunica directamente con places-ms y está representada con un ícono distintivo. Esta función es invocada de forma periódica por un nuevo Scheduler, al cual también fue agregado al diagrama. Todos los componentes originales,

como kong-instance, variables-ms, measurements-ms y sus respectivas bases de datos, se mantuvieron sin cambios, conservando la estructura inicial del sistema.



### Análisis del resultado del experimento

A partir del experimento realizado, se puede concluir que la arquitectura implementada benefició directamente al Atributo de Calidad (ASR) planteado, el cual exige que, una vez desplegado el sistema, la integración de un nuevo servicio no tome más de dos horas. Este objetivo se cumplió gracias al enfoque de microservicios adoptado, el cual permite desarrollar y desplegar servicios de manera independiente, reduciendo significativamente los tiempos de integración. La incorporación del microservicio *Places* y su vinculación con servicios existentes como *Measurements* y la *Cloud Function* se realizó sin requerir modificaciones sustanciales en el sistema base, lo que evidencia el nivel de desacoplamiento alcanzado. Asimismo, el uso de tecnologías como Docker para contenedorización, Kong como API Gateway, y servicios gestionados como Cloud Run y Cloud Scheduler facilitaron una orquestación eficiente y ágil de los componentes. Esta infraestructura estandarizada no solo permitió una integración rápida, sino también garantizó la estabilidad del sistema en su conjunto. En resumen, la arquitectura no solo respondió al requerimiento funcional del ASR, sino que además demostró su capacidad para escalar y adaptarse a nuevas necesidades sin comprometer tiempos ni calidad del servicio.

### Créditos:

A continuación, se observan los créditos disponibles en la cuenta al finalizar el laboratorio:

Créditos

Todos los créditos

Consulta y descarga los detalles de crédito aquí. Los descuentos por compromiso de uso activos no están incluidos aquí y se pueden ver en la [página Compromisos](#).

Filtro Filtrar créditos										
Nombre del crédito	Estado ↑	Porcentaje restante	Valor restante	Valor original	Tipo	ID de crédito	Permiso ⓘ			
Free Trial	⚠ Vence en 8 días	<div><div></div></div> 92%	\$1,167,087.43	\$1,262,802.70	De un solo uso	FreeTrialUpgrad...	Uso específico; consulta la			✓
Free Trial	❌ Vencido	—	\$1,262,741.14	\$1,263,338.00	De un solo uso	FreeTrial:Credit-...	Uso específico; consulta la			✓