



CARGA DE DATOS

Una vez que las funciones de recolección de precios (web scrapping) y de detección se realizan, dejarán los resultados dentro de un bucket generado en Google Cloud Storage con ficheros de formatos CSV o XLSX. Es entonces cuando se ejecutará la función para cargar estos datos en una Base de Datos generada dentro un servicio Google Cloud SQL.

La estrategia de contar con una Base de Datos para esta información es la de tener de manera versátil el acceso a la información. También se necesitará contar con un historial de dicha información para su utilización en diferentes puntos de la aplicación.

La base de datos se ha generado dentro de un servicio Google Cloud SQL para contar con las ventajas que ofrece un servicio de estas características, tales como Accesibilidad, Escalabilidad y Seguridad.

ID de instancia	Tipo	Dirección IP pública	Dirección IP privada	Nombre de la conexión con la instancia	Alta disponibilidad	Ubicación	Almacenamiento usado	Acciones
bd1-kcp	PostgreSQL 12	34.79.94.25		keepcoding-projectfinal-01.eur...	HABILITADA	europe-west1-d	71	

La elección de PostgreSQL se debe, en primer lugar, a que se trata de un sistema de gestión de Base de Datos con licencia libre. Este sistema también cuenta con funciones para trabajar con información JSON, aspecto que en futuras evoluciones de la aplicación será muy conveniente.

Nombre	Intercalación	Grupo de caracteres
postgres	en_US.UTF8	UTF8
proyecto	en_US.UTF8	UTF8

ALGORITMO DE DETECCIÓN



Actualmente se cuentan con 4 tablas para la aplicación:

- **Tabla Store.** En esta tabla se almacena la información de precio obtenida a través de la función de scraping.

The screenshot shows the pgAdmin interface with a query executed in the 'Query Editor'. The query is: `SELECT supermercado, tipo, fecha, categoria, category, subcategoria, precio, peso_gr, precio_kg FROM public.stores;` The results are displayed in a table with 9 columns: supermercado, tipo, fecha, categoria, category, subcategoria, precio, peso_gr, and precio_kg. The data shows 16 records for 'Carrefour' with various fruits and their prices.

	supermercado	tipo	fecha	categoria	category	subcategoria	precio	peso_gr	precio_kg
1	Carrefour	fruta	2022-03-24	banana	banana	granel	1.25	1000	1.25
2	Carrefour	fruta	2022-03-24	fresón	strawberry	tarina	2.29	1000	2.29
3	Carrefour	fruta	2022-03-24	aguacate	avocado	carrefour	1.99	1000	1.99
4	Carrefour	fruta	2022-03-24	uva	grape	blanca	2.4	1000	2.40
5	Carrefour	fruta	2022-03-24	manzana	apple	golden	1.95	1000	1.95
6	Carrefour	fruta	2022-03-24	naranja	orange	zumo	4.76	4000	1.19
7	Carrefour	fruta	2022-03-24	plátano	banana	canarias	2.0	5000	0.40
8	Carrefour	fruta	2022-03-24	manzana	apple	royal	1.69	1000	1.69
9	Carrefour	fruta	2022-03-24	arándanos	blueberries	carrefour	2.79	1000	2.79
10	Carrefour	fruta	2022-03-24	pera	pear	conferencia	1.99	1000	1.99
11	Carrefour	fruta	2022-03-24	pera	pear	conferencia	1.69	1000	1.69
12	Carrefour	fruta	2022-03-24	naranjas	oranges	mesa	5.34	1000	5.34
13	Carrefour	fruta	2022-03-24	frambuesa	raspberry	carrefour	2.89	1000	2.89
14	Carrefour	fruta	2022-03-24	manzana	apple	golden	1.65	1000	1.65
15	Carrefour	fruta	2022-03-24	kiwi	kiwi	círculo	3.19	1000	3.19
16	Carrefour	fruta	2022-03-24	limón	lemon	ecológico	1.49	1000	1.49

- **Tabla Detección.** Para cada detección realizada por la función correspondiente se genera un fichero que será almacenado en esta tabla. Se indica un ID por cada detección, y por cada clase detectada en una imagen se genera un registro con el número de elementos detectados de dicha clase.

The screenshot shows the pgAdmin interface with a query executed in the 'Query Editor'. The query is: `SELECT id_deteccion, product, qty_detected, image_url, date_created FROM public.detection;` The results are displayed in a table with 5 columns: id_deteccion, product, qty_detected, image_url, and date_created. The data shows 17 records of product detections with their respective IDs, quantities, image URLs, and creation dates.

	id_deteccion	product	qty_detected	image_url	date_created
1	35	potato	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_35.jpg	2022-03-26
2	35	orange	2	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_35.jpg	2022-03-26
3	35	pineapple	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_35.jpg	2022-03-26
4	35	apple	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_35.jpg	2022-03-26
5	35	lemon	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_35.jpg	2022-03-26
6	35	pear	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_35.jpg	2022-03-26
7	35	banana	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_35.jpg	2022-03-26
8	35	pepper	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_35.jpg	2022-03-26
9	35	carrot	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_35.jpg	2022-03-26
10	39	potato	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_39.jpg	2022-03-30
11	39	orange	8	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_39.jpg	2022-03-30
12	39	pineapple	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_39.jpg	2022-03-30
13	39	apple	10	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_39.jpg	2022-03-30
14	39	lemon	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_39.jpg	2022-03-30
15	39	pear	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_39.jpg	2022-03-30
16	39	banana	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_39.jpg	2022-03-30
17	39	pepper	0	https://storage.googleapis.com/kcp-bucket-01/datos_imagenes_out/deteccion_39.jpg	2022-03-30

ALGORITMO DE DETECCIÓN



- Tabla Shoplist. Esta tabla almacena la lista de compra de los productos que el usuario requiere para una determinada fecha.

The screenshot shows the pgAdmin interface with the 'shoplist' table selected in the left sidebar. The 'Query Editor' tab is active, displaying a SQL query that selects data from the 'shoplist' table. The 'Data Output' tab shows the results of the query.

date_created	product	qty_pieces_wished
date	text	numeric
2022-03-22	lemon	10
2022-03-22	potato	11
2022-03-22	apple	6
2022-03-22	banana	7
2022-03-22	pear	5
2022-03-22	carrot	12
2022-03-22	pepper	5
2022-03-22	orange	20
2022-03-22	pineapple	2
2022-03-24	lemon	15
2022-03-24	potato	7
2022-03-24	apple	7
2022-03-24	banana	7
2022-03-24	pear	7
2022-03-24	carrot	10
2022-03-24	pepper	7
2022-03-24	orange	15
2022-03-24	pineapple	2
2022-03-28	lemon	10
2022-03-28	potato	10
2022-03-28	apple	8

- Tabla Equivalencias. Dado que los precios de la tabla Store se determinan por kilogramos, esta tabla se genera para determinar la equivalencia en piezas de cada producto por kilo.

The screenshot shows the pgAdmin interface with the 'equivalencias' table selected in the left sidebar. The 'Query Editor' tab is active, displaying a SQL query that selects data from the 'equivalencias' table. The 'Data Output' tab shows the results of the query.

date_created	product	qty_kg	qty_pieces
date	text	numeric	numeric
2022-03-31	lemon	1	10
2022-03-31	apple	1	6
2022-03-31	carrot	1	8
2022-03-31	orange	1	6
2022-03-31	pepper	1	5
2022-03-31	pear	1	6
2022-03-31	potato	1	5
2022-03-31	banana	1	5
2022-03-31	pineapple	2	1



ALGORITMO DE DETECCIÓN

La función de Carga de Datos obtendrá los datos desde un bucket generado en Cloud Storage. Para cada tabla se tiene una carpeta correspondiente dentro del bucket donde los datos serán descargados, se procesaran y se realizará la inserción de estos mismos en la tabla.

COPIAMOS LA CARPETA DATOS DE NUESTRO BUCKET A NUESTRO LOCAL PARA TRATARLOS ANTES DE REALIZAR LA CARGA

```
[ ] ! gsutil -m cp -r gs://$BUCKET_NAME/datos_deteccion /content/

Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220323.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220327.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220326.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220331.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220332.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220328.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220324.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220325.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220329.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220330.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220401.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220402.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220403.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220404.csv...
Copying gs://kcp-bucket-01/datos_deteccion/datos_deteccion_datos_deteccion_20220408.csv...
```

LEEMOS EL FICHERO CSV DE DETECCION A UN DATAFRAME

```
[ ] import os
import pandas as pd
import glob

files_joined = os.path.join('/content/datos_deteccion', '*.csv')
list_files_store = glob.glob(files_joined)
df_data_deteccion = pd.concat(map(pd.read_csv, list_files_store), ignore_index=True)
```

COMPROBAMOS LOS DATOS DEL DATAFRAME

```
print(df_data_deteccion.columns)
df_data_deteccion.head()
```

```
Index(['id_deteccion', 'product', 'qty_detected', 'image_url', 'date_created'], dtype='object')
```

	id_deteccion	product	qty_detected	image_url	date_created
0	42	potato	0	https://storage.googleapis.com/kcp-bucket-01/d...	20220402
1	42	orange	2	https://storage.googleapis.com/kcp-bucket-01/d...	20220402
2	42	pineapple	0	https://storage.googleapis.com/kcp-bucket-01/d...	20220402
3	42	apple	11	https://storage.googleapis.com/kcp-bucket-01/d...	20220402
4	42	lemon	0	https://storage.googleapis.com/kcp-bucket-01/d...	20220402

Una vez realizada la carga de los datos en la tabla, la información de la carpeta correspondiente se moverá a la carpeta de datos_processed para dejar así libre las carpetas de datos para el siguiente envío de información por parte las funciones de recolección de precios y detección.

En futuras actualizaciones, se generaría una función para la carga de información para las tablas sholist y equivalencias.