# PSet6_PF_Evidencias

Está es la estructura del proyecto:

```
project/
│
├── docker-compose.yml
├── .env.example
├── feature-builder/
│      ├── Dockerfile
│      └── build_features.py
│
├── notebooks/
│      └── 01_ingesta_prices_raw.ipynb
          └── 02_build_features_prototipo.ipynb
          └── 03_verificacion.ipynb
  │
└── requirements.txt
```

En jupyter notebook hay un notebook llamado 01_ingesta_prices_raw.ipynb donde se crea el esquema raw, al crearlo en la celda 2, podemos verificar que efectivamente se creó ese esquema metiéndonos en el contenedor y examinando los esquemas, con:

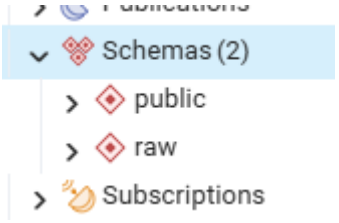
```
docker exec -it postgres bash
psql -U postgres -d marketdb
```

Con lo que ya se ingresa a la base de datos y ahí podemos verlo con

```
\dn
```

```
PS E:\Octavo Semestre\Data Mining\PSet6> docker exec -it postgres bash
root@36b1d2addd5c:/# psql -U postgres -d marketdb
psql (16.11 (Debian 16.11-1.pgdg13+1))
Type "help" for help.

marketdb=# \dn
      List of schemas
  Name  |       Owner
--------+-------------------
 public | pg_database_owner
 raw    | postgres
```

En la imagen anterior vemos que se creó el esquema raw, e igualmente podemos verlo en Pgadmin, así:




En el notebook se creó la tabla prices_daily en el esquema raw con las siguientes columnas:

Podemos verificar las columnas y su creación dentro del contenedor de postgres:

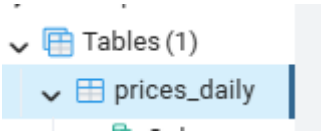
```
marketdb=# \dt raw.*
            List of relations
 Schema |     Name     | Type  |  Owner
--------+--------------+-------+----------
 raw    | prices_daily | table | postgres
(1 row)
```

Para raw.prices_daily:

```
\d raw.prices_daily
```

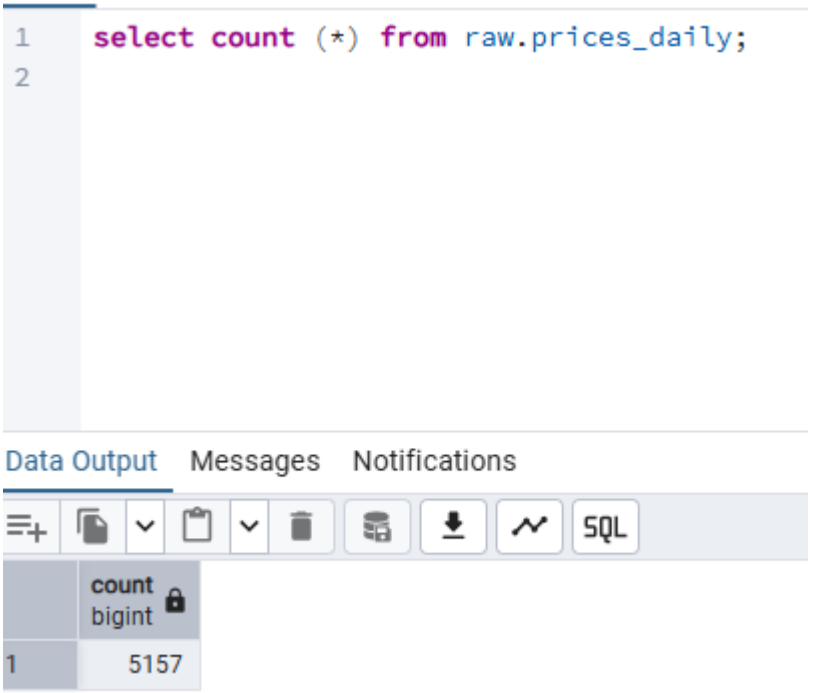
```
                    Table "raw.prices_daily"
     Column      |            Type             | Collation | Nullable | Default
-----------------+-----------------------------+-----------+----------+---------
 date            | date                        |           | not null |
 ticker          | character varying(20)       |           | not null |
 open            | double precision            |           |          |
 high            | double precision            |           |          |
 low             | double precision            |           |          |
 close           | double precision            |           |          |
 adj_close       | double precision            |           |          |
 volume          | bigint                      |           |          |
 run_id          | character varying(50)       |           |          |
 ingested_at_utc | timestamp without time zone |           |          |
 source_name     | character varying(50)       |           |          |
Indexes:
    "prices_daily_pkey" PRIMARY KEY, btree (date, ticker)
```

En Pgadmin:

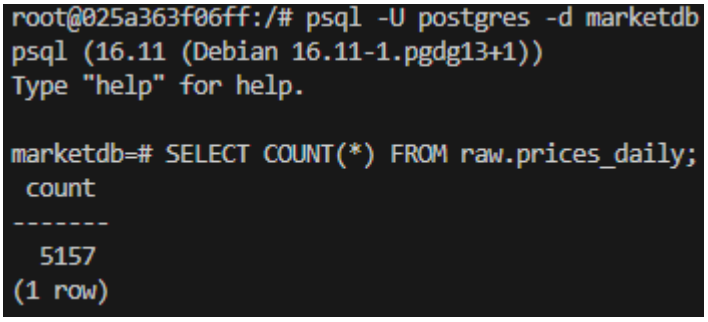



Después de la ingesta de datos en raw.prices_daily, verificamos que se haya insertado el número de filas correcto, que es 5157 filas.

En Pgadmin:




En el contenedor de postgres:



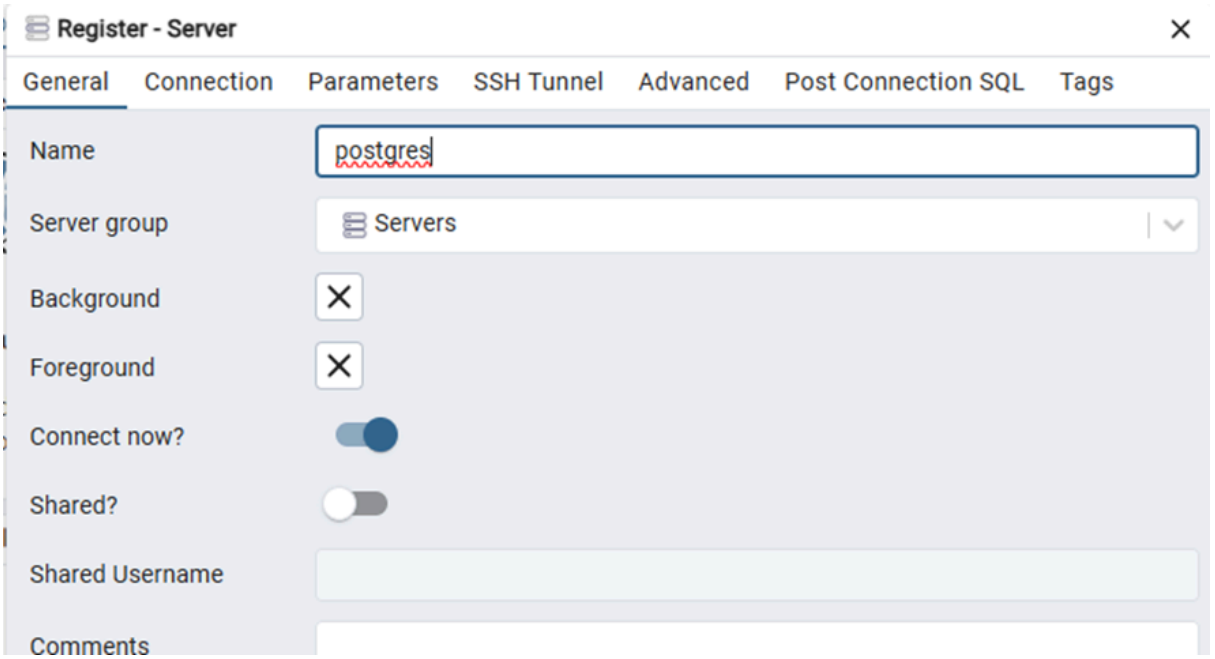

**Creación del esquema analytics y la tabla daily_features**

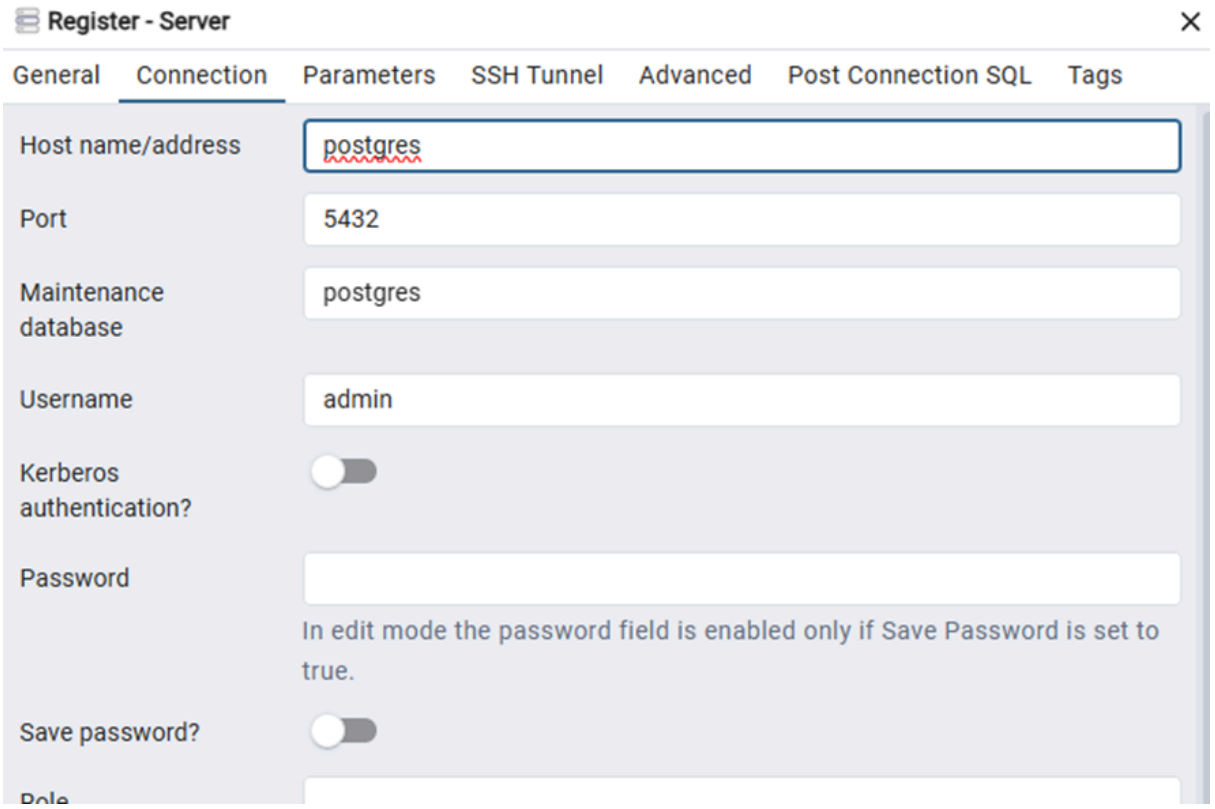
En Pgadmin se corre una sola vez el siguiente código:

```
CREATE SCHEMA IF NOT EXISTS analytics;

CREATE TABLE IF NOT EXISTS analytics.daily_features (
    date DATE NOT NULL,
    ticker VARCHAR(20) NOT NULL,

    -- Identificación temporal
    year INT,
    month INT,
    day_of_week INT,

    -- Mercado
    open DOUBLE PRECISION,
    high DOUBLE PRECISION,
    low DOUBLE PRECISION,
    close DOUBLE PRECISION,
    volume BIGINT,

    return_close_open DOUBLE PRECISION,
    return_prev_close DOUBLE PRECISION,
    volatility_5d DOUBLE PRECISION,

    -- Metadatos
    run_id VARCHAR(50),
    ingested_at_utc TIMESTAMPTZ,
```

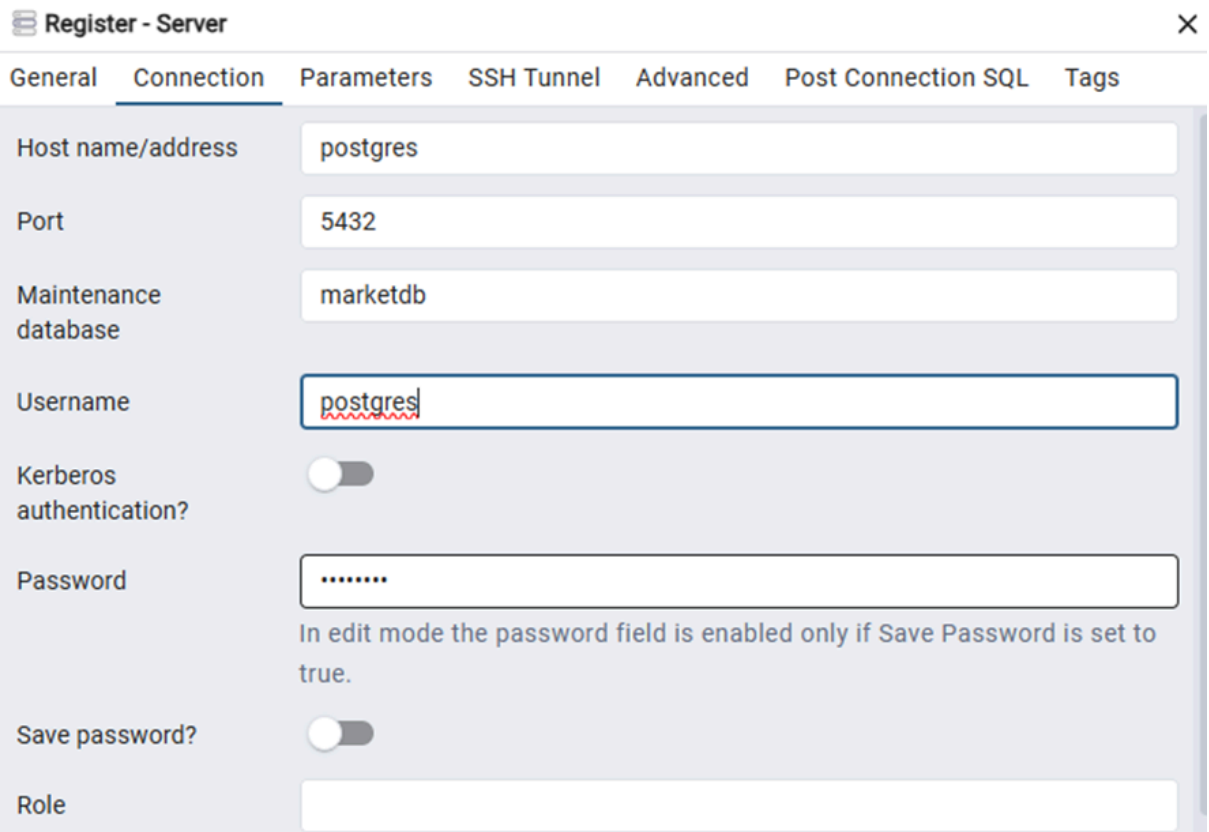
```
    PRIMARY KEY (date, ticker)
);
```

Para la conexión con postgres en Pgadmin:

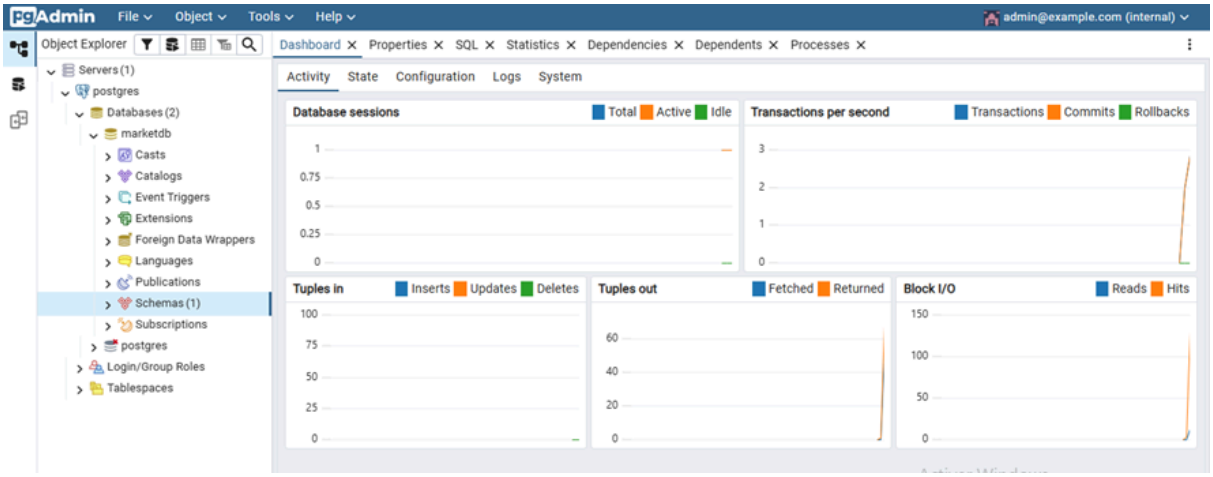



Lo siguiente es lo que tengo en las credenciales del archivo .env




Conexión exitosa de pgadmin con postgres:




Comandos para ingestar a la tabla analytics.daily_features

```
docker compose run feature-builder --mode full --ticker AAPL --overwrite t
rue
docker compose run feature-builder --mode full --ticker MSFT --overwrite
true
```

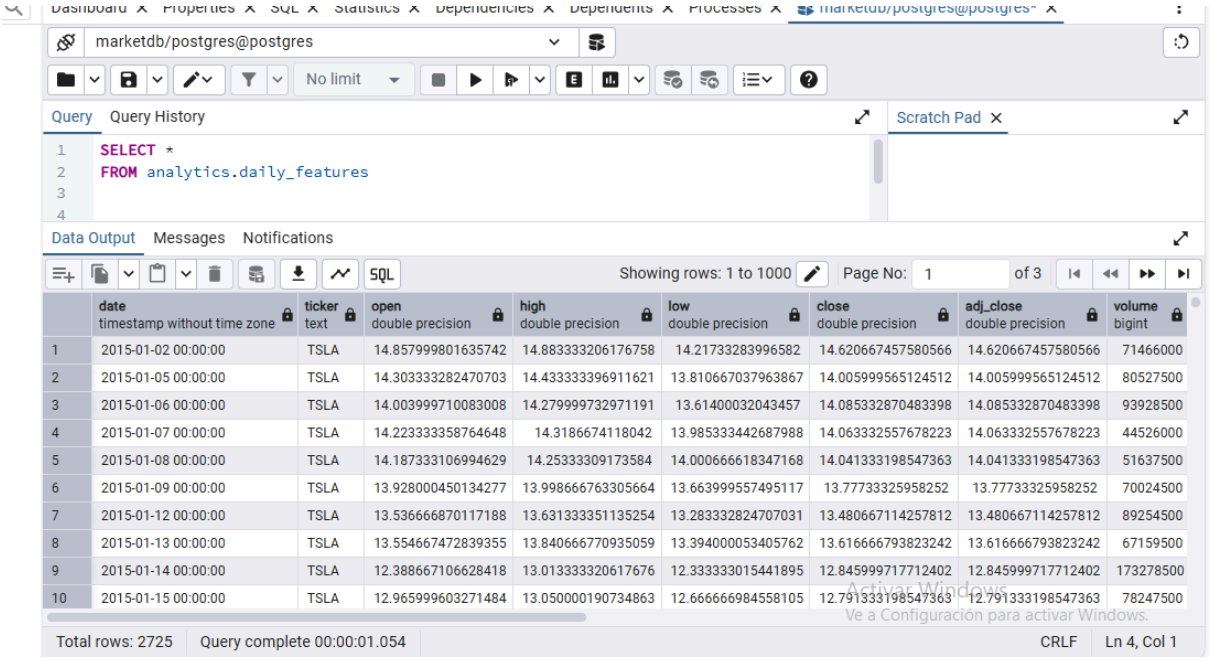
```
docker compose run feature-builder --mode full --ticker TSLA --overwrite t
rue
```

```
PS E:\Octavo Semestre\Data Mining\PSet6> docker compose run feature-builder --mode full --ticker
MSFT --overwrite false
time="2025-11-29T15:40:55-05:00" level=warning msg="Found orphan containers ([pset6-feature-build
er-run-86e86663a079 pset6-feature-builder-run-252ed8741bdd pset6-feature-builder-run-fbff605044d7
 pset6-feature-builder-run-a337cfcf2501]) for this project. If you removed or renamed this servic
e in your compose file, you can run this command with the --remove-orphans flag to clean it up."
[+] Creating 1/1
 ✓ Container postgres  Running                                                              0.0s
Insertando (idempotente)…
Procesadas 1719 filas para MSFT
Fecha min: 2019-01-02 00:00:00, max: 2025-10-31 00:00:00
```

Tabla analytics.daily_features

```
                    Table "analytics.daily_features"
      Column       |            Type             | Collation | Nullable | Default
-------------------+-----------------------------+-----------+----------+---------
 date              | timestamp without time zone |           |          |
 ticker            | text                        |           |          |
 open              | double precision            |           |          |
 high              | double precision            |           |          |
 low               | double precision            |           |          |
 close             | double precision            |           |          |
 adj_close         | double precision            |           |          |
 volume            | bigint                      |           |          |
 run_id            | text                        |           |          |
 ingested_at_utc   | timestamp with time zone    |           |          |
 source_name       | text                        |           |          |
 year              | integer                     |           |          |
 month             | integer                     |           |          |
 day_of_week       | integer                     |           |          |
 return_close_open | double precision            |           |          |
 return_prev_close | double precision            |           |          |
 volatility_5d     | double precision            |           |          |
```

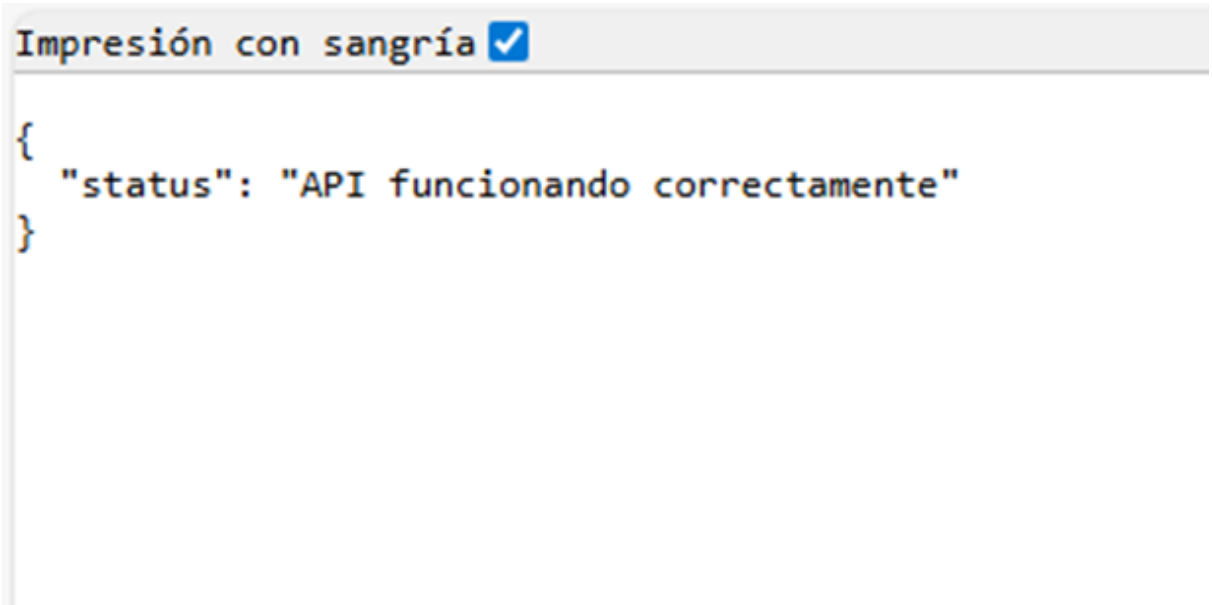
En este caso se insertó el ticker Tesla.




| | date timestamp without time zone | ticker text | open double precision | high double precision | low double precision | close double precision | adj_close double precision | volume bigint |
|---|---|---|---|---|---|---|---|---|
| 1 | 2015-01-02 00:00:00 | TSLA | 14.857999801635742 | 14.883333206176758 | 14.21733283996582 | 14.620667457580566 | 14.620667457580566 | 71466000 |
| 2 | 2015-01-05 00:00:00 | TSLA | 14.303333282470703 | 14.433333396911621 | 13.810667037963867 | 14.005999565124512 | 14.005999565124512 | 80527500 |
| 3 | 2015-01-06 00:00:00 | TSLA | 14.003999710083008 | 14.279999732971191 | 13.61400032043457 | 14.085332870483398 | 14.085332870483398 | 93928500 |
| 4 | 2015-01-07 00:00:00 | TSLA | 14.223333358764648 | 14.3186674118042 | 13.985333442687988 | 14.063332557678223 | 14.063332557678223 | 44526000 |
| 5 | 2015-01-08 00:00:00 | TSLA | 14.187333106994629 | 14.25333309173584 | 14.000666618347168 | 14.041333198547363 | 14.041333198547363 | 51637500 |
| 6 | 2015-01-09 00:00:00 | TSLA | 13.928000450134277 | 13.998666763305664 | 13.663999557495117 | 13.77733325958252 | 13.77733325958252 | 70024500 |
| 7 | 2015-01-12 00:00:00 | TSLA | 13.536666870117188 | 13.631333351135254 | 13.283332824707031 | 13.480667114257812 | 13.480667114257812 | 89254500 |
| 8 | 2015-01-13 00:00:00 | TSLA | 13.554667472839355 | 13.840666770935059 | 13.394000053405762 | 13.616666793823242 | 13.616666793823242 | 67159500 |
| 9 | 2015-01-14 00:00:00 | TSLA | 12.388667106628418 | 13.013333320617676 | 12.333333015441895 | 12.845999717712402 | 12.845999717712402 | 173278500 |
| 10 | 2015-01-15 00:00:00 | TSLA | 12.965999603271484 | 13.050000190734863 | 12.666666984558105 | 12.791333198547363 | 12.791333198547363 | 78247500 |

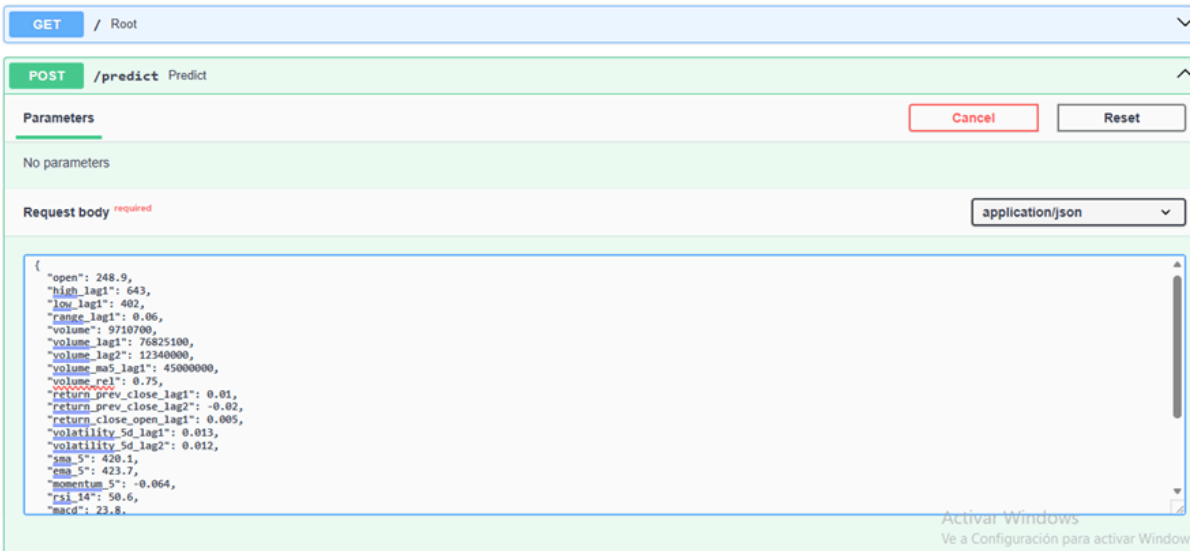


## Evidencias Funcionamiento de la API



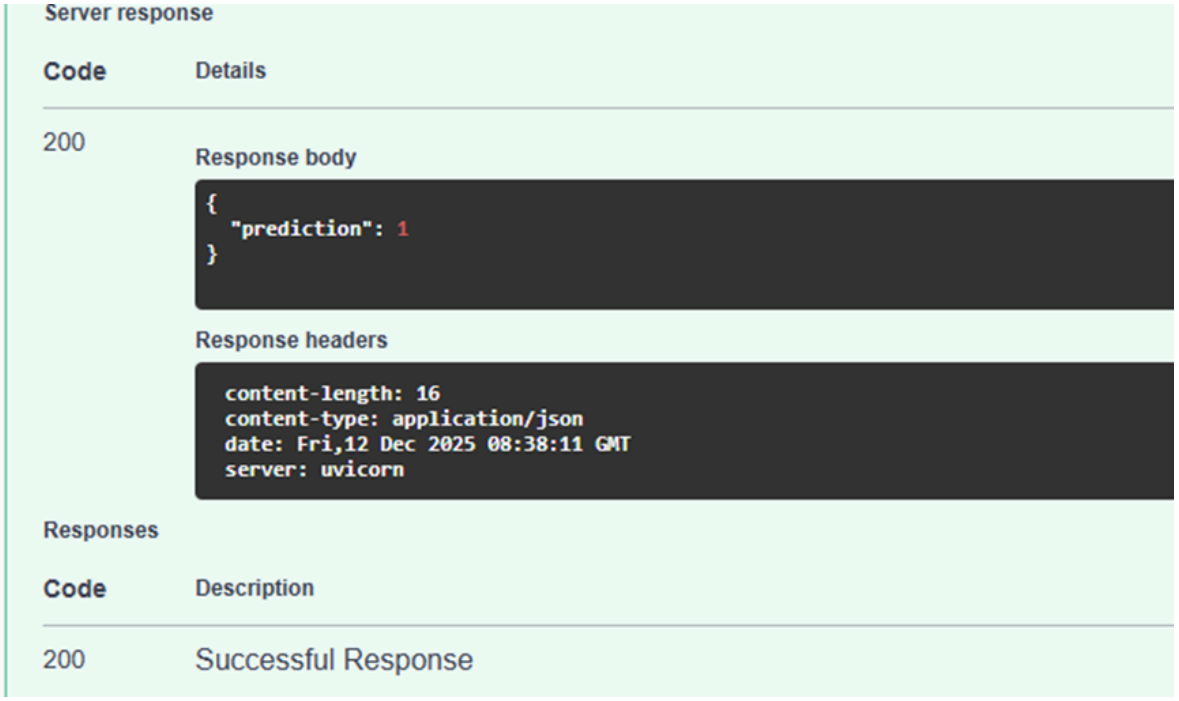

localhost:8000/docs#/default/predict_predict_post

Al colocar todos los parámetros de manera correcta:


GET / Root
POST /predict Predict
Parameters
Cancel
Reset
No parameters
Request body required
application/json
{
"open": 248.9,
"high_lag1": 643,
"low_lag1": 402,
"range_lag1": 0.06,
"volume": 9710700,
"volume_lag1": 76825100,
"volume_lag2": 12340000,
"volume_ma5_lag1": 45000000,
"volume_rel": 0.75,
"return_prev_close_lag1": 0.01,
"return_prev_close_lag2": -0.02,
"return_close_open_lag1": 0.005,
"volatility_5d_lag1": 0.013,
"volatility_5d_lag2": 0.012,
"sma_5": 420.1,
"ema_5": 423.7,
"momentum_5": -0.064,
"rsi_14": 50.6,
"macd": 23.8,
Activar Windows
Ve a Configuración para activar Window

Curl

```
curl -X 'POST' \
  'http://localhost:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "open": 248.9,
  "high_lag1": 643,
  "low_lag1": 402,
  "range_lag1": 0.06,
  "volume": 9710700,
  "volume_lag1": 76825100,
  "volume_lag2": 12340000,
  "volume_ma5_lag1": 45000000,
  "volume_rel": 0.75,
  "return_prev_close_lag1": 0.01,
  "return_prev_close_lag2": -0.02,
  "return_close_open_lag1": 0.005,
  "volatility_5d_lag1": 0.013,
  "volatility_5d_lag2": 0.012,
  "sma_5": 420.1,
  "ema_5": 423.7,
  "momentum_5": -0.064,
  "rsi_14": 50.6,
  "macd": 23.8,
  "macd_signal": 32.6,
  "boll_position": -0.21,
  "dist_max_5": -0.13,
  "dist_min_5": 0,
  "day_of_week": 3,
  "month": 1
}
'
```




Si uno de los campos no se incluye:

Curl

```
curl -X 'POST' \
  'http://localhost:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "open": 248.9,
  "high_lag1": 643,
  "low_lag1": 402,
  "range_lag1": 0.06,
  "volume": 9710700,
  "volume_lag1": 76825100,
  "volume_lag2": 12340000,
  "volume_ma5_lag1": 45000000,
  "volume_rel": 0.75,
  "return_prev_close_lag1": 0.01,
  "return_prev_close_lag2": -0.02,
  "return_close_open_lag1": 0.005,
  "volatility_5d_lag1": 0.013,
  "volatility_5d_lag2": 0.012,
  "sma_5": 420.1,
  "momentum_5": -0.064,
  "rsi_14": 50.6,
  "macd": 23.8,
  "macd_signal": 32.6,
  "boll_position": -0.21,
  "dist_max_5": -0.13,
  "dist_min_5": 0,
  "day_of_week": 3,
  "month": 1
}
'
```

Server response

| Code | Details |
| --- | --- |
| 422 | Error: Unprocessable Entity |

Response body


```
{
  "detail": [
    {
      "type": "missing",
      "loc": [
        "body",
        "macd"
      ],
      "msg": "Field required",
      "input": {
        "open": 248.9,
        "high_lag1": 643,
        "low_lag1": 402,
        "range_lag1": 0.06,
        "volume": 9710700,
        "volume_lag1": 76825100,
        "volume_lag2": 12340000,
        "volume_ma5_lag1": 45000000,
        "volume_rel": 0.75,
        "return_prev_close_lag1": 0.01,
        "return_prev_close_lag2": -0.02,
        "return_close_open_lag1": 0.005,
        "volatility_5d_lag1": 0.013,
        "volatility_5d_lag2": 0.012,
        "sma_5": 420.1,
        "ema_5": 423.7,
        "momentum_5": -0.064,
        "rsi_14": 50.6,
```