

Proyecto Integrado V – Línea de Énfasis Entrega Final

German Darío Arbeláez Ríos

Juan Carlos Presiga Montoya

GRUPO 13

PREICA2501B020128

Andrés Felipe Callejas



MEDELLÍN

Junio 2025

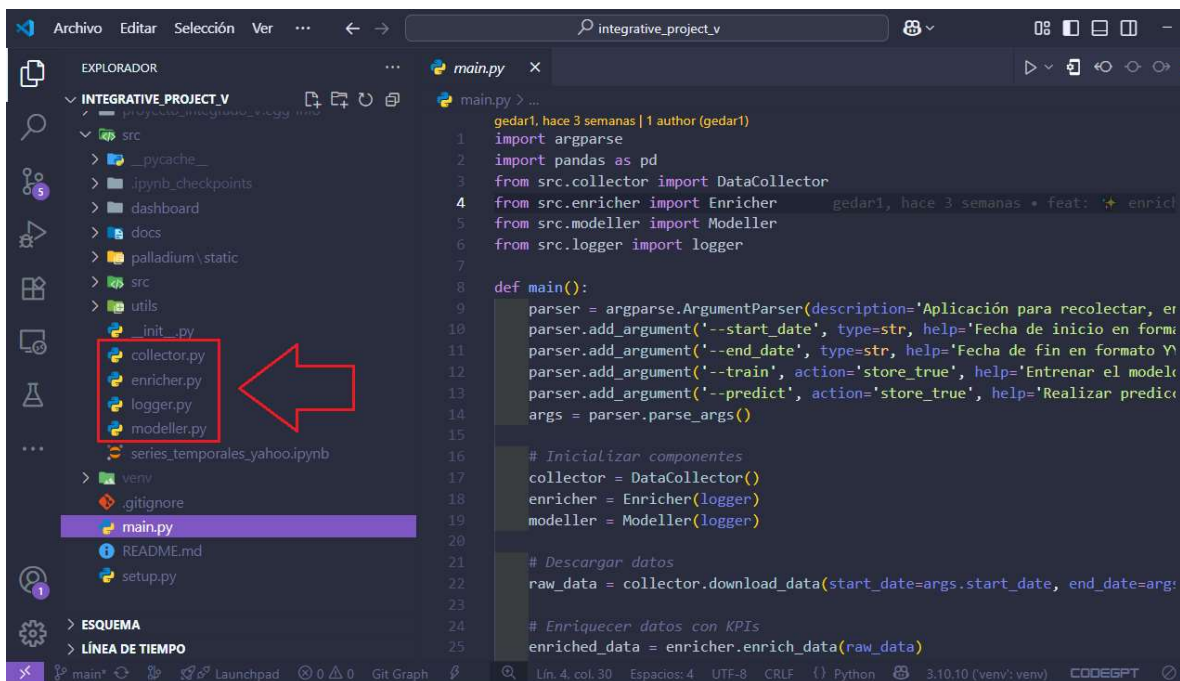
Actividades

Se realiza video con la explicación de cada tema propuesto para la actividad como es:

- Estructura y funcionamiento del código (collector, modeller, workflows, logging).
- Indicador económico seleccionado.
- Modelo predictivo y su métrica.
- Análisis de los KPI finales.

Estructura

En la imagen adjunta se puede observar los procesos de collector, enricher, logger y modeller con el cual se tomaron y modelaron los datos



The image shows a code editor interface with a file explorer on the left and a code editor on the right. The file explorer shows a project structure for 'INTEGRATIVE_PROJECT_V' with a 'src' directory containing 'collector.py', 'enricher.py', 'logger.py', and 'modeller.py'. A red box highlights these four files, and a red arrow points from the box to the code editor. The code editor shows the 'main.py' file with the following code:

```
1 import argparse
2 import pandas as pd
3 from src.collector import DataCollector
4 from src.enricher import Enricher
5 from src.modeller import Modeller
6 from src.logger import logger
7
8 def main():
9     parser = argparse.ArgumentParser(description='Aplicación para recolectar, en
10     parser.add_argument('--start_date', type=str, help='Fecha de inicio en form
11     parser.add_argument('--end_date', type=str, help='Fecha de fin en formato Y
12     parser.add_argument('--train', action='store_true', help='Entrenar el model
13     parser.add_argument('--predict', action='store_true', help='Realizar predic
14     args = parser.parse_args()
15
16     # Inicializar componentes
17     collector = DataCollector()
18     enricher = Enricher(logger)
19     modeller = Modeller(logger)
20
21     # Descargar datos
22     raw_data = collector.download_data(start_date=args.start_date, end_date=arg
23
24     # Enriquecer datos con KPIs
25     enriched_data = enricher.enrich_data(raw_data)
```

Indicador Económico

De la pagina de finance.yahoo.com se toma el indicador de Palladium, el cual es un mineral que se utiliza en diferentes procesos industriales como catalizador, en joyería, odontología, electrónica, medicina etc.



Modelo Predictivo

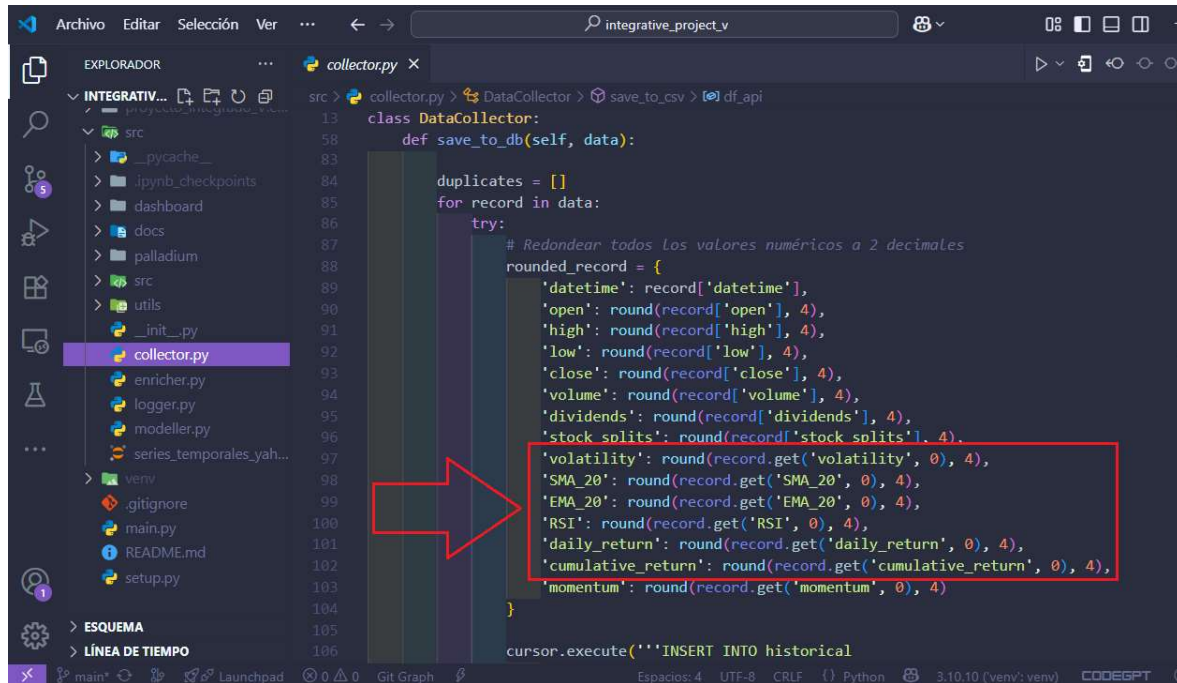
En modeller.py se realiza la creación y entrenamiento del modelo a través de randomforest

```
EXPLORADOR
  INTEGRATIV...
  src
    __pycache__
    .ipynb_checkpoints
    dashboard
    docs
    palladium
    src
    utils
    __init__.py
    collector.py
    enricher.py
    logger.py
    modeller.py
    series_temporales_yah...
    venv
    .gitignore
    main.py
    README.md
    setup.py
  ESQUEMA
  LÍNEA DE TIEMPO

modeller.py
14 class Modeller:
109 def entrenar_df(self, df=pd.DataFrame()):
114     # Asumimos que la última columna es el target (precio futuro)
115     X = df.iloc[:, :-1] # Todas las columnas excepto la última
116     y = df.iloc[:, -1] # Última columna como target
117
118     # Dividir los datos en conjuntos de entrenamiento y prueba
119     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_s
120
121     # Escalar los datos
122     scaler = StandardScaler()
123     X_train_scaled = scaler.fit_transform(X_train)
124     X_test_scaled = scaler.transform(X_test)
125
126     # Crear y entrenar el modelo
127     modelo = RandomForestRegressor(
128         n_estimators=100, # número de árboles
129         max_depth=10, # profundidad máxima de cada árbol
130         random_state=42 # para reproducibilidad
131     )
132     modelo.fit(X_train_scaled, y_train)
133
134     # Guardar el modelo y el scaler
135     self.guardar_modelo({
136         'model': modelo,
137         'scaler': scaler
138     })
```

KPI generados

En collector.py se realiza la generación de los KPI's implementados



```
class DataCollector:
    def save_to_db(self, data):
        duplicates = []
        for record in data:
            try:
                # Redondear todos los valores numéricos a 2 decimales
                rounded_record = {
                    'datetime': record['datetime'],
                    'open': round(record['open'], 4),
                    'high': round(record['high'], 4),
                    'low': round(record['low'], 4),
                    'close': round(record['close'], 4),
                    'volume': round(record['volume'], 4),
                    'dividends': round(record['dividends'], 4),
                    'stock_splits': round(record['stock_splits'], 4),
                    'volatility': round(record.get('volatility', 0), 4),
                    'SMA_20': round(record.get('SMA_20', 0), 4),
                    'EMA_20': round(record.get('EMA_20', 0), 4),
                    'RSI': round(record.get('RSI', 0), 4),
                    'daily_return': round(record.get('daily_return', 0), 4),
                    'cumulative_return': round(record.get('cumulative_return', 0), 4),
                    'momentum': round(record.get('momentum', 0), 4)
                }
                cursor.execute('INSERT INTO historical
```

Así mismo en la base de datos quedan registros los datos asociados a cada kpi



	volatility	SMA_20	EMA_20	RSI	daily_return	cumulative_return
30	55.0583	945.745	966.8895	66.113	0.0052	1.0112
31	53.5329	951.57	969.4143	64.0845	0.0059	1.0171
32	52.2326	955.035	969.3844	57.8704	-0.0245	0.9922
33	49.0107	958.5	967.2239	44.9029	-0.0231	0.9693
34	46.376	961.62	965.8693	53.8248	0.0067	0.9757
35	42.7211	964.965	964.4055	55.7014	-0.0026	0.9732
36	39.0693	967.235	960.7764	49.7529	-0.0255	0.9484
37	39.6157	966.92	954.9025	33.5833	-0.0294	0.9205
38	41.9485	965.475	948.6451	32.9787	-0.011	0.9104
39	43.6943	962.125	944.8789	24.3418	0.0224	0.9308
40	44.1706	961.54	942.7	23.9555	0.0142	0.944
41	45.93	960.105	938.0619	20.6782	-0.0304	0.9153
42	48.4441	957.97	932.9988	22.0653	-0.0102	0.906
43	50.7714	953.365	928.5037	25.6423	0.001	0.9069
251	52.2368	948.645	922.6028	22.7634	-0.0088	0.8980

En la carpeta dashboard quedan almacenadas las gráficas realizadas en Power BI de los KPI's y la predicción realizada a la volatilidad

