

Amining TT

César Chirino

Tabla de Contenidos

- **Amining TT**
 - **Tabla de Contenidos**
 - **Descripción**
 - [Características](#)
 - [Estructura](#)
 - [Componentes](#)
 - [Requisitos](#)
 - [Uso](#)
 - [Licencia](#)
 - [Contacto](#)

Descripción

Este proyecto es una aplicación de análisis de datos desarrollada en Python y Dash. La aplicación permite a los usuarios visualizar diferentes métricas y tendencias a través de gráficos interactivos.

El principal objetivo de este proyecto es proporcionar una herramienta sencilla para la visualización de indicadores clave de rendimiento (KPI), la generación de informes y el manejo eficiente de conjuntos de datos, todo ello a través de una interfaz amigable para el usuario. Además, se ha hecho énfasis en mantener un código bien estructurado y escalable.

Características

- **Dashboards Dinámicos:** Dashboards interactivos para la visualización de datos en tiempo real.
- **Diseño Modular::** Estructura bien organizada con componentes reutilizables.
- **Informes Exhaustivos::** Generación de informes detallados con visualizaciones, accesibles a través de la aplicación.

Estructura

assets/

├─ css/

│ ├─ lineatiempo_styles.css

│ └─ resume_styles.css

├─ img/

│ ├─ Chancado_Correas_Icon.png

│ ├─ Concentradora_Icon.png

│ ├─ Dot_Circle.png

│ ├─ Horizontal_Line.png

│ ├─ Pipe_Icon.png

│ ├─ Ship_Icon.png

│ ├─ Truck_Icon.png

│ └─ Vertical_Line.png

data/

├─ autogestion.csv

├─ comments.csv

└─ graficos.csv

src/

├─ config.py

├─ dash_components/

│ ├─ callbacks/

│ ├─ home_callbacks.py

│ ├─ lineatiempo_callbacks.py

│ ├─ register_callbacks.py

│ ├─ resumen_callbacks.py

│ ├─ sidebar_callback.py

│ └─ __init__.py

│ └─ graphs/

│ ├─ resumen_comments.py

│ └─ resumen_graphs.py

│ └─ layouts/

│ ├─ app_layout.py

│ ├─ home_layout.py

│ ├─ lineatiempo_layout.py

│ ├─ resumen_layout.py

│ ├─ sidebar_layout.py

│ └─ __init__.py

└─ styles/

```
|       |— home_styles.py
|       |— sidebar_style.py
|— data_processing/
|   |— data_cleaning.py
|   |— data_loader.py
|   |— __init__.py
|— utils/
|   |— data_config.py
|   |— __init__.py

tests/
```

Componentes

- **Assets:** Contiene archivos de diseño, animación y las imágenes utilizadas en la aplicación.
- **Data:** Almacena los archivos de datos locales que se utilizan en la aplicación.
- **Src:** Incluye las funciones principales de la aplicación.
- **Dash Components:** Componentes individuales que conforman la estructura de la aplicación.
- **Callbacks:** Gestiona la lógica de los callbacks, que permiten la interacción y actualización dinámica de los componentes.
- **Graphs:** Encargado de la creación de gráficos utilizados en la aplicación.
- **Layouts:** Define la estructura y disposición visual de cada página dentro de la aplicación.
- **Styles:** Contiene los estilos personalizados aplicados a las visualizaciones.
- **Data Processing:** Gestiona la lógica para la lectura, escritura y procesamiento de diversas bases de datos.
- **Utils:** Funciones auxiliares que se utilizan en diferentes partes de la aplicación para tareas diversas.

Requisitos

- Python 3.8+
- Las dependencias necesarias están listadas en el archivo `requirements.txt` .

Uso

1. Manual:

- Crear y activar un entorno virtual:

```
python -m venv venv
source venv/bin/activate
venv\Scripts\activate
```

- Instalar dependencias

```
pip install -r requirements.txt
```

- Iniciar la aplicación

```
python main.py
```

- Acceder a la aplicación en el navegador:

```
http://127.0.0.1:8050/
```

2. Automático (Windows):

- Ejecutar instalador

```
setup_win.bat
```

Licencia

Este proyecto está bajo la Licencia MIT. Consulta el archivo LICENSE para más detalles.

Contacto

Si tienes alguna pregunta o comentario sobre este proyecto, no dudes en contactarme a través del correo: chirinocesar.me@gmail.com.