

DOCUMENTACIÓN TÉCNICA: REPARTO DE INGRESOS

1.- TECNOLOGÍAS USADAS Y VERSIONES

Este script está desarrollado en Python y utiliza las siguientes librerías principales:

- Pandas '2.2.3': utiliza la manipulación y análisis de datos, especialmente para la gestión de DataFrames.
- SQLAlchemy '2.0.41': usada como *ORM* para la conexión y comunicación con la base de datos *PostgreSQL*. Abstrae la lógica de conexión.
- Psycopg2-binary '2.9.10': driver de *PostgreSQL* para Python que utiliza *SQLAlchemy* de forma interna para conectarse a la base de datos.
- Numpy '2.2.6': dependencia de pandas y es utilizada para hacer cálculos numéricos eficientes.
- Python-dateutil '2.9.2.post0': dependencia de pandas, proporciona extensiones potentes para el manejo de fechas y horas.

2. FUNCIONALIDAD

El propósito principal de la aplicación es procesar los ingresos por matrículas de cursos y repartirlos a lo largo del tiempo según un conjunto de reglas del negocio.

El flujo de trabajo es el siguiente:

- 1) Conexión a la base de datos: el script se conecta a una base de datos *PostgreSQL* para obtener los datos necesarios.
- 2) Extracción de datos: se extraen datos de las tablas '**fct_matricula**' (información de matrículas) y '**dim_producto**' (detalles de los cursos).
- 3) Procesamiento y reparto: para cada matrícula, el script determina un rango de fechas de reparto basándose en una jerarquía de reglas:
 - a) *Modalidad ONLINE*: el importe total se asigna a la fecha de la matrícula.

- b) *Otras modalidades*: se sigue una prioridad para encontrar las fechas de inicio y fin:
 - i) Fecha de reconocimiento del curso.
 - ii) Fechas de inicio y fin del producto.
 - iii) Fecha de matrícula más la duración en meses del curso (aproximación).
- 4) Generación de archivos: el script genera tres archivos CSV como salida:
 - a) *reparto_ingresos_output.csv*: contiene el reparto detallado de ingresos por día para cada matrícula.
 - b) *reparto_ingresos_debug.csv*: un archivo de depuración que muestra los datos de entrada y las fechas calculadas para cada matrícula, facilitando la validación de lógica.
 - c) *resumen_por_matricula_y_curso.csv*: un resumen que agrupa por curso y por matrícula, mostrando el importe total repartido para cada una.

3. ESTRUCTURA DE FICHEROS

Aplicación que se compone de un único script principal y los archivos de salida que se genera.

- 1) *reparto_ingresos.py*: el script principal que contiene toda la lógica de conexión, extracción, procesamiento y guardado de datos.
- 2) *requeriments.txt*: archivo que define las dependencias de Python para asegurar la reproducibilidad del entorno.
- 3) *reparto_ingresos_output.csv*: fichero con el reparto diario de ingresos.
- 4) *reparto_ingresos-debug.csv*: fichero de depuración.
- 5) *resumen_por_matricula_y_curso.csv*: fichero de resumen de reparto.

4. COSAS PARA TENER EN CUENTA

- 1) *Credenciales de la base de datos*: las credenciales de acceso a la base de datos ('**DB_HOST**', '**DB_NAME**', '**DB_USER**' y '**DB_PASSWORD**') están definidas como constantes al inicio del script. Para un entorno de producción, se recomienda gestionarlas de forma segura.
- 2) *Manejo de fechas inválidas*: si para una matrícula no se puede determinar un rango de fechas válido (ej. fechas inconsistentes, nulas, o una fecha de inicio posterior a la de fin), el script emite una advertencia (Advertencia:

No se pudieron determinar fechas válidas...) y, como fallback, reparte el importe íntegro en la `fec_matricula`. Esto asegura que la suma total de importes repartidos siempre coincida con la suma original, pero es importante revisar estas advertencias para detectar posibles errores en los datos de origen.

- 3) *Codificación de CSV*: Los archivos CSV se guardan con codificación 'utf-8-sig' para garantizar la compatibilidad y correcta visualización de caracteres especiales (como tildes o 'ñ') en programas como Microsoft Excel.