

DOCUMENTACION SCRIPT GENERADOR DE TICKETS
SINTÉTICOS – COCHE ELECTRICO

GRUPO 1

Contenido

Introducción.....	3
Entradas, salidas y estructura de carpetas.	3
Archivos de entrada	3
Archivos de salida	3
Parámetros globales y aleatoriedad	4
Normalización, limpieza y detección de columnas	4
Utilidades clave.....	4
Detección flexible de columnas	5
Lógica de precios EV	5
Geolocalización y validación de coordenadas	5
Identificadores y fiscalidad	5
Modelo de datos de salida (Ticket JSON)	6
Flujo de generación	7
Funciones principales (índice breve).....	7
Validaciones y asunciones.....	8

Introducción

Este script genera **tickets sintéticos de recarga eléctrica** (facturas simplificadas) para varias empresas de transporte y sus usuarios, usando:

- Un catálogo de **puntos de carga** (provincia, coordenadas, operador, potencia máxima).
- Un **mapa de CIF/NIF por operador** (empresa/compañía del punto de carga).
- Una **tabla de precios EV** por provincia, mes y **banda AC/DC** (opcional).
Si no se aporta tabla de precios, el script **inyecta precios por defecto** por provincia/banda/mes.

Los resultados se guardan en:

- `data/tickets_ev_sinteticos.jsonl` (un ticket por línea).
- `data/tickets_ev_sinteticos.json` (lista JSON completa).

Entradas, salidas y estructura de carpetas.

Archivos de entrada

Ubicados en /data

- `PuntosCarga.csv`
 - Columnas autodetectadas: id punto, operador/empresa, provincia, municipio, dirección, lat, lon, potencia máxima (kW)
- `CIFs_puntos_carga.csv`
 - Columnas para empresa/operador y CIF/NIF (alias admitidos)
- `ocm_agg_2025.csv`
 - base de precios EV para construir índices por estación y/o provincia, por mes y AC/DC.

El lector es tolerante a diferentes encoding.

Archivos de salida

Se crean automáticamente

- `Data/`
 - `PuntosCarga.csv`
 - `CIFs_puntos_carga.csv`
 - `ocm_agg_2025.csv`
 - `Tickets_ev_sinteticos.jsonl`
 - `Tickets_ev_sinteticos.json`

Parámetros globales y aleatoriedad

- Replicabilidad: Se añade una semilla para asegurar que los tickets obtenidos son coherentes con los empleados
- Rango temporal:
 - YEAR = 2025
 - FECHA_INI = 2025-01-01, FECHA_FIN = 2025-07-31
- Población sintética:
 - N_EMPRESAS_TRANSP = 3
 - USUARIOS_POR_EMPRESA = 3
 - TICKETS_POR_USUARIO = 50
- Métodos de pago:
 - PESO_TARJETA = 0.85 → 85% “Tarjeta crédito”, 15% “Efectivo”.
- Consumo kWh:
 - Distribución triangular: kwh_min=10, kwh_mode=35, kwh_max=80 (redondeo a 2 decimales).
- Banda potencia
 - DC_THRESHOLD_KW = 2 → DC si potencia es mayor de 20 kW, sino es AC
- Precio por estación y día:
 - Base mensual por (provincia normalizada, producto canonizado, mes)
 - Offset por estación: id_estacion: [-0.02, 0.02] kW/h.
 - Ruido diario: [-0.01, 0.01] kW/h.

Normalización, limpieza y detección de columnas

Utilidades clave

- norm_txt(x)
Quita tildes, pasa a minúsculas y recorta espacios.
- _simp/pick
Heurísticas para detectar alias de columnas (id, dirección, provincia, municipioi, lat/lon, potencia, operador)
- to_float_locale(series)
Convierte números con coma decimal a float.
- prov_key(x)
Normaliza nombres de provincia (corrige sinónimos: vizcaya→bizkaia, guipuzcoa→gipuzkoa, la coruña→a coruña, orense→ourense, etc.).
- Autocorrección lat/lon → Si parecen invertidas, intenta permutarlas; valida rango España.

Detección flexible de columnas

- Id estación: múltiples alias (`idestacion`, `stationid`, `poi`, `ocmid`, `codigo`, `id...`). Si no existe, se genera `_idgen`.
- Operador/empresa: busca entre (`operador`, `operator`, `operatorinfo`, `operatorname...`).
- Provincia/municipio/dirección: varios alias.
- Lat/Lon: admite `lat`, `latitude`, `addressinfo` `latitude`, y equivalentes para `lon`.
- Potencia: alias como `powerkw`, `maxpowerkw`, `potencia_max`, etc.
- Caso empaquetado: si una columna contiene campos concatenados con `|`, se descompone y se mapea por posición (`id`, `dirección`, `provincia`, `municipio`, `lon`, `lat`, `potencia...`).

Lógica de precios EV

- Banda por potencia
`band_by_power(potencia_max_kw) → AC (≤20 kW) o DC (>20 kW).`
- Precio base mensual
`precio_base_mes(prov_norm, mes, band, id_estacion=None)`
intenta, en orden:
- Precio por estación/mes/banda (`precios_station_idx`), si existe.
- Precio por provincia/mes/banda (`precios_prov_idx`), si existe.
- Si no hay datos, en `precio_diario` se usa la media disponible para el mismo mes/banda o, como último recurso, 0.45 €/kWh.
- Precio diario
`precio_diario(prov_norm, fecha, id_estacion, band) =`
`base_mes + station_offset(id_estacion) + ruido_diario`
Redondeo a 3 decimales y mínimo 0.15 €/kWh.

Geolocalización y validación de coordenadas

- `_coords_from_row(row):`
 - Intenta `lat`, `lon`; si no parecen válidas en el rango España (`lat 27-44.5`, `lon -20-5.5`), prueba invertidas `lon`, `lat`.
 - Si siguen inválidas, devuelve `NaN`, `NaN`.
- `generar_ticket()` reintenta hasta 5 veces muestreando estaciones aleatorias hasta conseguir coordenadas válidas; si no, puede dejar `lat/lon = null`.

Identificadores y fiscalidad

- CIF/NIF de operador
 - `get_cif(empresa)` consulta el diccionario mapeado desde `CIFs_puntos_carga.csv`.
 - Si no hay, genera un CIF plausible con `cif_generate()` (control de dígito/letra basado en reglas españolas simplificadas).

- Impuestos.
- IVA: $\text{IVA_TIPO} = 0.21$
- Impuesto Especial de Electricidad (IEE): $\text{IEE_TIPO} = 0.0511$
- `calcular_importes(kwh, precio_unit, unit_price_includes_taxes=False)` soporta tanto precios base (sin impuestos) como precios finales (si `unit_price_includes_taxes=True`).
- Ajuste de céntimos para asegurar que $\text{base} + \text{IEE} + \text{IVA} = \text{total}$.

Modelo de datos de salida (Ticket JSON)

- Cada ticket (`dict`) contiene:

```
{
  "idTicket": "T-EFFAB4C5059B",
  "idEmpresa": "EMP001",
  "empresaNombre": "Transporte_01 S.L.",
  "idUsuario": "EMP001-U28",
  "fechaEmision": "2025-04-13",
  "horaEmision": "04:23:15",
  "metodoPago": "Tarjeta crédito",
  "estacion": {
    "id": "ES*WEN*ESGASVALDEMORO14",
    "provincia": "Madrid",
    "municipio": "Valdemoro",
    "direccion": "Av De Andalucía, 519",
    "lat": 40.17221,
    "lon": -3.665604,
    "empresa": "WENEA EMOBILITY SERVICES S.L.",
    "nifEmpresa": "B16895971",
    "potenciaMaxKW": 150.0,
    "tarifa": "DC"
  },
  "lineas": [
    {
      "producto": "Electricidad",
      "kwh": 76.92,
      "precioUnitarioSinImpuestos": 0.544,
      "precioUnitario": 0.692,
      "importe": 53.22
    }
  ],
  "baseImponible": 41.84,
  "iee": 2.14,
  "iva": 9.24,
  "total": 53.22,
  "moneda": "EUR",
  "tipoDocumento": "Factura simplificada"
}
```

Flujo de generación

1. Carga/normalización de puntos de carga (`est_std`), y maca CIF (`map_grupo_to_cif`), carga y definición de precios (`precios_station_idx / precios_prov_idx`).
2. Empresas y usuarios:
 - o empresas: EMP001...EMP003
 - o usuarios: por empresa, desde un contado inicial (`usuario_cont=28`)
3. Pool de estaciones:
 - o Conversión robusta a float de lat/lon y potencia_max_kw.
 - o Normalización de provincia → provincia_norm.
 - o Cálculo de banda AC/DC.
4. Generación:
 - o `generar_todos()` itera empresas→usuarios→TICKETS_POR_USUARIO y llama a `generar_ticket()`.
 - Fecha/hora aleatoria (rango definido).
 - Muestreo de estación válida (coords plausibles).
 - Determinación de banda, precio diario, kWh y método de pago.
 - Cálculo de base, IEE, IVA, total.
 - Ensamblado del ticket JSON.
5. Escritura a disco:
 - o JSONL y JSON en data

Funciones principales (índice breve)

- **Lectura/Limpieza**
 - o `load_csv_guess(path)` – lector robusto CSV.
 - o `norm_txt, prov_key, to_float_locale, _simp, pick`
- **Precios**
 - o `band_by_power, precio_base_mes, station_offset, precio_diario`.
- **Identificación fiscal**
 - o `get_cif, cif_generate`.
- **Aleatoriedad y formato**
 - o `random_fecha, random_hora, random_kwh, elegir_metodo_pago, str_fecha, str_hora, round2, round3`.
- **Coordenadas**
 - o `_to_float_locale, _coords_from_row`.
- **Modelo y generación**
 - o `Empresa, Usuario, generar_ticket(empresa, usuario), generar_todos()`.

Validaciones y asunciones

- Se asume que meses disponibles en Precios cubren enero–julio (1–7).
- Banda AC/DC por potencia máxima del punto ($>20 \text{ kW} \rightarrow \text{DC}$).
- Precios:
 - Preferencia por índice estación→mes→banda, luego provincia→mes→banda.
 - Si no hay datos, se usan precios por defecto (AC/DC) por provincia y mes.
- Coordenadas: intento de autocorrección si están invertidas; se exigen rangos plausibles para España; si no, `null`.
- Impuestos: IVA 21% e IEE 5.11%; ajuste de céntimos para cuadrar totales.
- Producto único: "Electricidad" (línea única por ticket).
- Usuarios por empresa: en este script se mantiene el valor inicial (`USUARIOS_POR_EMPRESA = 3`).
- Robustez de columnas: detección por alias y, si procede, desempaquetado de columnas con valores |.