# **PROYECTO FINAL** - PROGRAMACIÓN II

**Letizia Rosa Laghi (2023-1485)** / Profesor Jean Charly Joseph Saint / ITLA / 2024-12-05

### Importación de librerías necesarias

In [51]:
```
import pandas as pd
from glob import glob
import duckdb
from random import randint
import os
import csv
from datetime import datetime
from dateutil.relativedelta import relativedelta
```

### Exploración

Detectar archivos de con datos de empleados (grupos 1 y 2)

In [111…
```
empleados = glob("Empleados/*.csv")
empleados

# Se encapsulan los archivos en la variable 'empleados'
```

Out[111…
```
['Empleados\\empleados_sucursal_A.csv',
 'Empleados\\empleados_sucursal_B.csv',
 'Empleados\\empleados_sucursal_C.csv',
 'Empleados\\empleados_sucursal_D.csv',
 'Empleados\\empleados_sucursal_E.csv',
 'Empleados\\empleados_sucursal_F.csv',
 'Empleados\\empleados_sucursal_G.csv',
 'Empleados\\empleados_sucursal_H.csv',
 'Empleados\\empleados_sucursal_I.csv',
 'Empleados\\empleados_sucursal_J.csv',
 'Empleados\\empleados_sucursal_K.csv',
 'Empleados\\empleados_sucursal_L.csv',
 'Empleados\\empleados_sucursal_M.csv',
 'Empleados\\empleados_sucursal_N.csv',
 'Empleados\\empleados_sucursal_O.csv']
```

Ver encabezados de archivos de empleados

In [112…
```
def ver_encabezados(ruta):
    """
    Esta función sirve para detectar archivos .csv
    en un directorio e imprimir sus encabezados.

    Args:
        ruta (_str_): ruta del directorio
    """
    archivos = glob(os.path.join(ruta, '*.csv'))

    for i in archivos:
        with open(i, 'r', newline='', encoding='utf-8') as archivo:
            lector = csv.reader(archivo)
            encabezados = next(lector)
            print(f"{os.path.basename(i)}: {encabezados}")

ver_encabezados('Empleados/')
```

```
empleados_sucursal_A.csv: ['id', 'sucursal', 'nombre', 'apellido', 'sexo', 'nacimiento', 'nacionalida
d', 'departamento', 'salario', 'comision', 'correo', 'telefono', 'puesto', 'direccion', 'contratacio
n']
empleados_sucursal_B.csv: ['id', 'sucursal', 'nombre', 'apellido', 'sexo', 'nacimiento', 'nacionalida
d', 'departamento', 'salario', 'comision', 'correo', 'telefono', 'puesto', 'direccion', 'contratacio
n']
empleados_sucursal_C.csv: ['id', 'sucursal', 'nombre', 'apellido', 'sexo', 'nacimiento', 'nacionalida
d', 'departamento', 'salario', 'comision', 'correo', 'telefono', 'puesto', 'direccion', 'contratacio
n']
empleados_sucursal_D.csv: ['id', 'sucursal', 'nombre', 'apellido', 'sexo', 'nacimiento', 'nacionalida
d', 'departamento', 'salario', 'comision', 'correo', 'telefono', 'puesto', 'direccion', 'contratacio
n']
empleados_sucursal_E.csv: ['id', 'sucursal', 'nombre', 'apellido', 'sexo', 'nacimiento', 'nacionalida
d', 'departamento', 'salario', 'comision', 'correo', 'telefono', 'puesto', 'direccion', 'contratacio
n']
empleados_sucursal_F.csv: ['id', 'sucursal', 'nombre', 'apellido', 'sexo', 'nacimiento', 'nacionalida
d', 'departamento', 'salario', 'comision', 'correo', 'telefono', 'puesto', 'direccion', 'contratacio
n']
empleados_sucursal_G.csv: ['id', 'sucursal', 'nombre', 'apellido', 'sexo', 'nacimiento', 'nacionalida
d', 'departamento', 'salario', 'comision', 'correo', 'telefono', 'puesto', 'direccion', 'contratacio
n']
empleados_sucursal_H.csv: ['id', 'sucursal', 'nombre', 'apellido', 'sexo', 'nacimiento', 'nacionalida
d', 'departamento', 'salario', 'comision', 'correo', 'telefono', 'puesto', 'direccion', 'contratacio
n']
empleados_sucursal_I.csv: ['id', 'sucursal', 'nombre', 'apellido', 'sexo', 'nacimiento', 'nacionalida
d', 'departamento', 'salario', 'comision', 'correo', 'telefono', 'puesto', 'direccion', 'contratacio
n']
empleados_sucursal_J.csv: ['id', 'sucursal', 'nombre', 'apellido', 'sexo', 'nacimiento', 'nacionalida
d', 'departamento', 'salario', 'comision', 'correo', 'telefono', 'puesto', 'direccion', 'contratacio
n']
empleados_sucursal_K.csv: ['id', 'nombre', 'apellido', 'nacimiento', 'nacionalidad', 'departamento',
'sucursal', 'sexo', 'telefono_personal', 'codigo_postal', 'salario', 'comision', 'vehiculo_asignado',
'hijos', 'flota']
empleados_sucursal_L.csv: ['id', 'nombre', 'apellido', 'nacimiento', 'nacionalidad', 'departamento',
'sucursal', 'sexo', 'telefono_personal', 'codigo_postal', 'salario', 'comision', 'vehiculo_asignado',
'hijos', 'flota']
empleados_sucursal_M.csv: ['id', 'nombre', 'apellido', 'nacimiento', 'nacionalidad', 'departamento',
'sucursal', 'sexo', 'telefono_personal', 'codigo_postal', 'salario', 'comision', 'vehiculo_asignado',
'hijos', 'flota']
empleados_sucursal_N.csv: ['id', 'nombre', 'apellido', 'nacimiento', 'nacionalidad', 'departamento',
'sucursal', 'sexo', 'telefono_personal', 'codigo_postal', 'salario', 'comision', 'vehiculo_asignado',
'hijos', 'flota']
empleados_sucursal_O.csv: ['id', 'nombre', 'apellido', 'nacimiento', 'nacionalidad', 'departamento',
'sucursal', 'sexo', 'telefono_personal', 'codigo_postal', 'salario', 'comision', 'vehiculo_asignado',
'hijos', 'flota']
```

### Detectar archivos de ventas (grupo 3)

In [113…

```
ventas = glob("Ventas/*.csv")
ventas

# Se encapsulan los archivos en la variable 'ventas'
```

Out[113…

```
['Ventas\\ventas_especiales.csv',
 'Ventas\\ventas_generales.csv',
 'Ventas\\ventas_pormayor.csv',
 'Ventas\\ventas_premium.csv',
 'Ventas\\ventas_promocion.csv']
```

### Ver encabezados de archivos de ventas

In [114…

```
ver_encabezados('Ventas/')
```

```
ventas_especiales.csv: ['id_venta', 'fecha_pedido', 'fecha_envio', 'nombre_cliente', 'apellido_client
e', 'correo_cliente', 'telefono_cliente', 'nacionalidad_cliente', 'vendedor', 'producto', 'cantidad',
'monto', 'impuesto', 'tipo_tarjeta', 'no_tarjeta']
ventas_generales.csv: ['id_venta', 'fecha_pedido', 'fecha_envio', 'nombre_cliente', 'apellido_client
e', 'correo_cliente', 'telefono_cliente', 'nacionalidad_cliente', 'vendedor', 'producto', 'cantidad',
'monto', 'impuesto', 'tipo_tarjeta', 'no_tarjeta']
ventas_pormayor.csv: ['id_venta', 'fecha_pedido', 'fecha_envio', 'nombre_cliente', 'apellido_client
e', 'correo_cliente', 'telefono_cliente', 'nacionalidad_cliente', 'vendedor', 'producto', 'cantidad',
'monto', 'impuesto', 'tipo_tarjeta', 'no_tarjeta']
ventas_premium.csv: ['id_venta', 'fecha_pedido', 'fecha_envio', 'nombre_cliente', 'apellido_cliente',
'correo_cliente', 'telefono_cliente', 'nacionalidad_cliente', 'vendedor', 'producto', 'cantidad', 'mo
nto', 'impuesto', 'tipo_tarjeta', 'no_tarjeta']
ventas_promocion.csv: ['id_venta', 'fecha_pedido', 'fecha_envio', 'nombre_cliente', 'apellido_client
e', 'correo_cliente', 'telefono_cliente', 'nacionalidad_cliente', 'vendedor', 'producto', 'cantidad',
'monto', 'impuesto', 'tipo_tarjeta', 'no_tarjeta']
```

**Selección de variables relevantes**

Selección y concatenación de variables relevantes de archivos de empleados

In [122…

```python
dataset = []

for i in empleados:
    df = pd.read_csv(i, usecols=[
        'id',
        'sucursal',
        'nombre',
        'apellido',
        'sexo',
        'nacimiento',
        'nacionalidad',
        'departamento',
        'comision',
        'salario'])
    dataset.append(df)

df1 = pd.concat(dataset)

"""
   Se crea un dataset vacío y se van agregando
   los DataFrames resultantes de la lectura de
   cada archivo de empleados.
"""

df1
```

Out[122…

| | id | sucursal | nombre | apellido | sexo | nacimiento | nacionalidad | departamento | salario | comision |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 99-4094901 | A | Clarisse | Vuittet | F | 1977-06-18 | United States | Training | 24672 | 0.43 |
| **1** | 26-6370584 | A | Ludovico | Priestner | M | 2006-04-19 | United States | Sales | 27779 | 0.18 |
| **2** | 52-7669688 | A | Hunfredo | Carwithen | M | 1979-10-05 | United States | Product Management | 20198 | 0.90 |
| **3** | 09-6282213 | A | Stacie | Rzehor | F | 1986-06-17 | United States | Accounting | 75526 | 0.96 |
| **4** | 98-6095003 | A | Alison | Ikringill | F | 1984-12-26 | United States | Business Development | 42595 | 0.41 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **5** | 58-3268041 | O | Kristal | Kingcote | F | 2001-10-11 | Dominican Republic | Research and Development | 19715 | 0.76 |
| **6** | 90-3827618 | O | Otho | Caselli | M | 1985-11-15 | United States | Services | 29267 | 0.58 |
| **7** | 78-9151673 | O | Rupert | Barsby | M | 1986-04-15 | Haiti | Legal | 55874 | 0.98 |
| **8** | 70-8330847 | O | Dmitri | Delucia | M | 1993-10-24 | United States | Marketing | 48442 | 0.67 |
| **9** | 38-7030185 | O | Jocelyn | Crees | F | 1982-04-16 | United States | Accounting | 20398 | 0.32 |

150 rows × 10 columns

### Selección y concatenación de variables relevantes en archivos de ventas

In [117…

```
dataset = []

for i in ventas:
    df = pd.read_csv(i, usecols=['vendedor', 'monto', 'id_venta'])
    dataset.append(df)

df2 = pd.concat(dataset)

"""
   Se crea un dataset vacío y se van agregando
   los DataFrames resultantes de la lectura de
   cada archivo de ventas.
"""

df2.head()
```

Out[117…

| | id_venta | vendedor | monto |
|---|---|---|---|
| **0** | 1 | 79-3781698 | 11171 |
| **1** | 2 | 56-6267907 | 12727 |
| **2** | 3 | 90-3763598 | 5845 |
| **3** | 4 | 46-2060669 | 11100 |
| **4** | 5 | 35-1306564 | 7431 |

### Agrupación de montos de ventas y cantidad de ventas por vendedores en archivos de ventas

In [ ]:

```
df2 = df.groupby('vendedor').agg(
    monto_ventas=('monto', 'sum'),
    cant_ventas=('id_venta', 'count')
```

```
).reset_index()

"""
    El DataFrame se agrupa por vendedor y
    se agrega a cada uno la suma de los
    montos de todas sus ventas y el conteo
    de las mismas.
"""

df2.head()
```

Out[ ]:

| | vendedor | monto_ventas | cant_ventas |
|---|---|---|---|
| **0** | 04-2721911 | 5704 | 1 |
| **1** | 04-3513015 | 12810 | 1 |
| **2** | 05-5140393 | 9127 | 1 |
| **3** | 05-6134170 | 3973 | 1 |
| **4** | 08-3359811 | 5822 | 1 |

### Unificación de los tres grupos en un dataframe

In [ ]:
```
df3 = pd.merge(df1, df2, left_on='id', right_on='vendedor', how ='left').drop(columns='vendedor')

"""
    La función merge() actúa como un join
    para unir el DataFrame de ventas y el de
    empleados por la columna 'vendedor' y
    'id', respectivamente.

    Se utiliza 'left_on' y 'right_on' porque
    las columnas no tienen el mismo nombre.
"""

df3.head()
```

Out[ ]:

| | id | sucursal | nombre | apellido | sexo | nacimiento | nacionalidad | departamento | salario | comision |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 99-4094901 | A | Clarisse | Vuittet | F | 1977-06-18 | United States | Training | 24672 | 0.43 |
| **1** | 26-6370584 | A | Ludovico | Priestner | M | 2006-04-19 | United States | Sales | 27779 | 0.18 |
| **2** | 52-7669688 | A | Hunfredo | Carwithen | M | 1979-10-05 | United States | Product Management | 20198 | 0.90 |
| **3** | 09-6282213 | A | Stacie | Rzehor | F | 1986-06-17 | United States | Accounting | 75526 | 0.96 |
| **4** | 98-6095003 | A | Alison | Ikringill | F | 1984-12-26 | United States | Business Development | 42595 | 0.41 |

## TAREAS ADICIONALES

### Unificar todos los archivos en un único CSV

In [108…
```
dataset = []

for i in empleados:
    df = pd.read_csv(i)
    dataset.append(df)

dfx = pd.concat(dataset)

dfx.to_csv('Unico_csv/empleados_unificado.csv', index=False)
```

```
# Todos los archivos de empleados son concatenados a
# un unico dataset que es convertido en un archivo CSV
```

In [109…
```
dataset = []

for i in ventas:
    df = pd.read_csv(i)
    dataset.append(df)

dfx = pd.concat(dataset)

dfx.to_csv('Unico_csv/ventas_unificado.csv', index=False)

# Todos los archivos de ventas son concatenados a un
# unico dataset que es convertido en un archivo CSV
```

**Generar libro de Excel en hojas separadas con los datos correspondientes a personas de entre 18 y 30 años de edad**

In [ ]:
```
hoy = datetime.now()
minimo = hoy - relativedelta(years=18)
maximo = hoy - relativedelta(years=30)

# A la fecha actual se le restan 18 y 30 años

minimo = minimo.strftime('%Y-%m-%d')
maximo = maximo.strftime('%Y-%m-%d')

# Formateo de fechas para hacerlas compatibles
# con las fechas de los datos de los archivos
```

In [110…
```
df4 = df3[(df3['nacimiento'] > maximo) & (df3['nacimiento'] < minimo)]
df4.head()

# DataFrame filtrado por fechas
```

Out[110…

| | id | sucursal | nombre | apellido | sexo | nacimiento | nacionalidad | departamento | salario | comisio |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 26-6370584 | A | Ludovico | Priestner | M | 2006-04-19 | United States | Sales | 27779 | 0.1 |
| **5** | 97-1407392 | A | Broddy | Featherston | M | 2000-12-22 | United States | Training | 14791 | 0.4 |
| **6** | 65-8192629 | A | Bonni | Mattevi | F | 2006-07-06 | United States | Training | 79834 | 0.7 |
| **7** | 04-1781862 | A | Grenville | House | M | 2001-08-15 | United States | Training | 31139 | 0.4 |
| **15** | 65-5729756 | B | Mead | Laffoley-Lane | M | 2006-10-17 | United States | Human Resources | 57369 | 0.2 |

In [ ]:
```
sucursales = df4['sucursal'].unique()

with pd.ExcelWriter(f'Libro_Excel/Informe_empleados(18-30).xlsx', engine='openpyxl') as writer:
    for i in sucursales:
        df = df4[(df4['sucursal'] == i)]
        df.to_excel(writer, sheet_name=f"Empleados {i}", index=False)

# Se hace una lista con los nombres de cada sucursal
# Se crea un libro de Excel segmentando las hojas por sucursal
```

**Unificación en una base de datos DuckDB con tabla para el análisis**

In [ ]:
```
for i in sucursales:
    df = df3[(df3['sucursal'] == i)]
    df.to_csv(f'BaseDatos/Informe_Sucursal_{i}.csv', index=False)
```

```
# A partir de los datos combinados de los tres grupos
# se crea un archivo CSV para cada sucursal
```

In [ ]:
```
conn = duckdb.connect(database='BaseDatos/InformeGeneral.db')

# Conexión con Duckdb y creación de base de datos
```

In [ ]:
```
conn.execute("""
CREATE TABLE IF NOT EXISTS Empleados (
    ID_EMPLEADO VARCHAR(10),
    SUCURSAL CHAR(1),
    NOMBRE VARCHAR(20),
    APELLIDO VARCHAR(25),
    SEXO CHAR(1),
    NACIMIENTO DATE,
    NACIONALIDAD VARCHAR(30),
    DEPARTAMENTO CHAR(50),
    SALARIO BIGINT,
    COMISION DOUBLE,
    MONTO_VENTAS DOUBLE,
    CANT_VENTAS DOUBLE
);
""")

conn.execute("""
INSERT INTO Empleados
SELECT * FROM read_csv_auto('BaseDatos/*.csv');
""")

# Generación de tabla para el análisis e inserción masiva de datos
# provenientes de los archivos CSV generados para cada sucursal
```

Out[ ]:
```
<duckdb.duckdb.DuckDBPyConnection at 0x21d55ccb4f0>
```

In [ ]:
```
conn.execute("SELECT * FROM Empleados;").fetchdf()

# Consulta de prueba
```

Out[ ]:

| | ID_EMPLEADO | SUCURSAL | NOMBRE | APELLIDO | SEXO | NACIMIENTO | NACIONALIDAD | DEPARTAMENTO |
|---|---|---|---|---|---|---|---|---|
| **0** | 99-4094901 | A | Clarisse | Vuittet | F | 1977-06-18 | United States | Training |
| **1** | 26-6370584 | A | Ludovico | Priestner | M | 2006-04-19 | United States | Sales |
| **2** | 52-7669688 | A | Hunfredo | Carwithen | M | 1979-10-05 | United States | Product Management |
| **3** | 09-6282213 | A | Stacie | Rzehor | F | 1986-06-17 | United States | Accounting |
| **4** | 98-6095003 | A | Alison | Ikringill | F | 1984-12-26 | United States | Business Development |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **145** | 58-3268041 | O | Kristal | Kingcote | F | 2001-10-11 | Dominican Republic | Research and Development |
| **146** | 90-3827618 | O | Otho | Caselli | M | 1985-11-15 | United States | Services |
| **147** | 78-9151673 | O | Rupert | Barsby | M | 1986-04-15 | Haiti | Legal |
| **148** | 70-8330847 | O | Dmitri | Delucia | M | 1993-10-24 | United States | Marketing |
| **149** | 38-7030185 | O | Jocelyn | Crees | F | 1982-04-16 | United States | Accounting |

150 rows × 12 columns

In [ ]:
```
conn.close()

# Cierre de la base de datos
```