

## Consigna - Sprint 4

### Procesamiento de Cheques Bancarios

El objetivo de este proyecto es desarrollar un script de Python llamado `listado_cheques.py` que permita procesar y consultar información de cheques bancarios almacenados en un archivo CSV. El script debe permitir a los usuarios filtrar y visualizar los datos de los cheques emitidos y depositados por un cliente específico, considerando diferentes criterios de filtrado, como el estado del cheque y el rango de fechas. Además, se debe gestionar la exportación de datos a un archivo CSV si es necesario.

En el siguiente link podrán encontrar más detalle de este tipo de archivos: <https://www.geeknetic.es/Archivo-CSV/que-es-y-para-que-sirve>

Dado este tipo de archivo se sabe que contiene los siguientes campos con la siguiente información:

- **NroCheque:** Número de cheque, este debe ser único por cuenta.
- **CodigoBanco:** Código numérico del banco, entre 1 y 100.
- **CodigoScurusal:** Código numérico de la sucursal del banco va entre 1 y 300.
- **NumeroCuentaOrigen:** Cuenta de origen del cheque.
- **NumeroCuentaDestino:** Cuenta donde se cobra el cheque.
- **Valor:** float con el valor del cheque.
- **FechaOrigen:** Fecha de emisión: (En timestamp)
- **FechaPago:** Fecha de pago o cobro del cheque (En timestamp)
- **DNI:** String con DNI del cliente donde se permite identificarlo
- **Estado:** Puede tener 3 valores pendiente, aprobado o rechazado.
- **Requisitos Específicos:**
- **Nombre del Archivo:** El script de Python se debe llamar `listado_cheques.py`.

#### Argumentos de Línea de Comando:

- **Nombre del archivo CSV:** Se debe proporcionar el nombre del archivo CSV que contiene los registros de los cheques.
- **DNI del Cliente:** Se debe proporcionar el DNI del cliente para el cual se realizará la consulta.
- **Salida (PANTALLA o CSV):** El usuario puede elegir si desea ver los resultados en la pantalla o exportarlos a un archivo CSV.
- **Tipo de Cheque (EMITIDO o DEPOSITADO):** El usuario debe especificar si desea consultar cheques emitidos o depositados.
- **Estado del Cheque (Opcional):** El usuario puede proporcionar un estado de cheque (PENDIENTE, APROBADO, RECHAZADO) como criterio de filtrado.

- Rango de Fechas (Opcional): El usuario puede especificar un rango de fechas para filtrar los cheques.

### Manejo de Errores:

Si se encuentra un número de cheque repetido en la misma cuenta para un DNI dado, mostrar un mensaje de error indicando el problema.

### Salida de Datos:

- Si el parámetro "Salida" es PANTALLA, imprimir por pantalla todos los valores correspondientes a la consulta.
- Si el parámetro "Salida" es CSV, exportar los resultados a un archivo CSV con el nombre en el formato "<DNI><TIMESTAMP\_ACTUAL>.csv". El archivo CSV debe contener las siguientes columnas: NroCheque, CodigoBanco, CodigoSucursal, NumeroCuentaOrigen, NumeroCuentaDestino, Valor, FechaOrigen, FechaPago, DNI, Estado.
- Filtrado por Estado (Opcional): Si el estado del cheque no se proporciona como parámetro, se deben imprimir los cheques sin filtrar por estado.

Documentación y Comentarios: Agregar comentarios descriptivos en el código para explicar su funcionalidad y proporcionar una documentación clara de cómo usar el script.

Validación de Parámetros: Asegurarse de que los parámetros proporcionados por el usuario sean válidos y manejar posibles errores de entrada.

Optimización del Código: Considerar la optimización del código para cargar y procesar grandes conjuntos de datos de manera eficiente.

Manejo de Fechas y Formato CSV: Asegurarse de que las fechas se manejen correctamente y se formateen adecuadamente al exportar a CSV.

### Ejemplo de archivo

```
NroCheque,CodigoBanco,CodigoSucursal,NumeroCuentaOrigen,NumeroCuentaDestino,Valor,FechaOrigen,FechaPago,DNI,Estado
1,15,102,12345678,98765432,1500.50,1631066400,1631854800,12345678,aprobado
2,22,203,87654321,56789012,750.25,1631152800,1631941200,98765432,pendiente
3,8,155,11111111,22222222,3000.75,1631066400,1631941200,12345678,rechazado
4,35,275,33333333,44444444,250.80,1631152800,1631854800,55555555,aprobado
5,45,55,11111111,56789012,1000.00,1631066400,1631854800,55555555,pendiente
6,12,44,33333333,98765432,500.25,1631152800,1631941200,98765432,aprobado
```

## EVOLUCIÓN CONTINUA

```
7,85,186,87654321,22222222,750.75,1631152800,1631941200,12345678,pendient
e
8,3,45,87654321,44444444,200.50,1631066400,1631854800,55555555,rechazado
9,70,101,33333333,56789012,450.60,1631152800,1631854800,12345678,aprobado
10,40,220,87654321,98765432,900.30,1631066400,1631941200,55555555,pendiente
```

Ejemplo 1: Consultar Cheques Emitidos de un Cliente Específico

```
python listado_cheques.py cheques.csv 12345678 PANTALLA EMITIDO
```

Resultado Esperado (PANTALLA):

NroCheque	CodigoBanco	CodigoScurusal	NumeroCuentaOrigen	NumeroCuentaDestino	Valor	FechaOrigen	FechaPago	DNI	Estado
1	15	102	12345678	98765432	1500.50	2021-09-08 10:00:00	2021-09-17 10:00:00	12345678	aprobado
3	8	155	11111111	22222222	3000.75	2021-09-08 10:00:00	2021-09-17 10:00:00	12345678	rechazado
7	85	186	87654321	22222222					

Ejemplo 2: Consultar Cheques Depositados de un Cliente con Filtro por Fecha

```
python listado_cheques.py cheques.csv 98765432 CSV DEPOSITADO --fecha 2021-09-12:2
```

Resultado Esperado (Archivo CSV): 98765432\_TIMESTAMP.csv

NroCheque	CodigoBanco	CodigoScurusal	NumeroCuentaOrigen	NumeroCuentaDestino	Valor	FechaOrigen	FechaPago	DNI	Estado
2,22,203,87654321,56789012,750.25,2021-09-09 12:00:00,2021-09-18 12:00:00,98765432,pendiente									
6,12,44,33333333,98765432,500.25,2021-09-09 12:00:00,2021-09-18 12:00:00,98765432,aprobado									

Ejemplo 3: Consultar Cheques Emitidos de un Cliente con Filtro por Estado

```
python listado_cheques.py cheques.csv 55555555 PANTALLA EMITIDO RECHAZADO
```

Resultado Esperado (PANTALLA):



NroCheque	CodigoBanco	CodigoScurusal	NumeroCuentaOrigen	NumeroCuentaDestino	Valor	FechaOrigen	FechaPago	DNI	Estado
8	3	45	87654321	44444444	200.50	2021-09-08 10:00:00	2021-09-17 10:00:00	55555555	rechazado

