

# Trabajo Práctico 1

[7508/9503] Sistemas Operativos

Curso 1 - Clua

Segundo cuatrimestre de 2020

Alumno:	CAROL LUGONES, Ignacio
Número de padrón:	100073
Email:	icarol@fi.uba.ar
Alumno:	MININO, Nahuel
Número de padrón:	99599
Email:	nminino@fi.uba.ar
Alumno:	TORRESETTI, Lisandro
Número de padrón:	99846
Email:	ltorresetti@fi.uba.ar

Repositorio: <https://github.com/IgnacioCarol/SisopTp>

## Índice

<b>1. Documentación</b>	<b>2</b>
1.1. Aclaraciones . . . . .	2
1.2. Descripción de problemas . . . . .	2
1.3. Archivos adicionales . . . . .	2
1.4. Estructura de directorios . . . . .	3
1.4.1. Parte 2 . . . . .	3

## 1. Documentación

### 1.1. Aclaraciones

Para la realización del trabajo práctico se realizaron las siguientes hipótesis:

#### 1. Chequeo de TFH:

Primero se analiza que el campo *RECORD\_TYPE* contenga las siglas 'TFH'. Luego que el *MERCHANT\_CODE* del registro coincida con el merchant code del archivo. Por último se analiza si el *NUMBER\_OF\_TRX\_RECORDS* es válido, lo cual sucede cuando todos sus dígitos son distintos de cero y el número indicado coincide con la cantidad de registros de transacciones que vienen a continuación del registro TFH.

En caso de que no se cumpla uno de los casos mencionados se descarta el archivo y se informa en el log cual fue el motivo de error. Al analizar el registro de esta forma si, por ejemplo, el registro posee un record type inválido y el merchant code tampoco es correcto, ya que se analiza primero el record type, lo que se informará en el log es este error y no que el merchant code es incorrecto. Por lo tanto, siempre se indica un error de todos los posibles que puede tener el registro TFH.

#### 2. Chequeo de TFD:

Al igual que para el TFH, los posibles errores en este tipo de registros se analizaron de la siguiente manera: primero se analiza que el campo *RECORD\_TYPE* contenga las siglas 'TFD'. En segundo lugar, se verifica que el *RECORD\_NUMBER* se corresponda con el número de registro correcto. Luego se analiza el *PROCESSING\_CODE* sea válido, lo cual sucede cuando su valor es 000000 ó 111111. Finalmente se analiza si el *ID\_PAYMENT\_METHOD* indica un valor que existe en la tabla de tarjetas homologadas.

Esto se repite para todos los registros TFD que posea el archivo, y en caso de que una condición no se cumpla se lo descarta y se informa en el log el motivo. Al igual que antes, si tenemos un registro TFD que posee más de un error sólo se marcará el primero que sea identificado según el orden que fue planteado para analizar los campos.

#### 3. Otros errores en los registros:

En caso de haber otro error en los registros TFH o TFD, como por ejemplo que un campo se encuentre vacío, no se descarta el archivo ya que solamente se analizan los casos detallados anteriormente.

### 1.2. Descripción de problemas

Ningún integrante del grupo se encontró con problemas al realizar la parte que le correspondía del trabajo práctico. Se realizó una división de tareas lo más equitativa posible para tener una mejor organización con respecto a los tiempos. Ante la aparición de dudas de implementación se realizó una puesta en común con los integrantes del grupo para poder decidir que hacer con ella. Además se consultaron distintas fuentes, como por ejemplo el material de clase, internet y libros para saber el funcionamiento de ciertos comandos.

### 1.3. Archivos adicionales

Se crearon dos archivos adicionales *salida1* y *salida2*. El motivo por el cual se crearon estos archivos es que se buscó modularizar más el código para hacerlo más legible y entendible. Además, al dividir las tareas, trabajar con archivos distintos permitió que cada integrante del grupo no dependa de los otros para realizar la parte que le fue otorgada.

Estos archivos se encargan de generar la salida 1 (grabar el archivo de liquidaciones) y la salida 2 (grabar el archivo de comisiones). Como estas salidas poseen diferencias es otra razón por la cual se decidió dividirlo y no hacer un único script.

## 1.4. Estructura de directorios

### 1.4.1. Parte 2

Los archivos de prueba propios se encuentran en la carpeta **proprios**. Dentro de esta carpeta se encuentran otras carpetas con archivos que contienen distintos tipos de errores, a continuación se mencionan dichas carpetas y los archivos que contienen.

#### 1. Filenames \_erroneos:

Esta carpeta contiene archivos con nombres erroneos ya sea porque no cumplen con el formato *C<MerchantCode>\_Lote<BatchNumber>*, porque el merchant code no existe en la tabla maestra de comercios ó porque el merchant code o el batch number no poseen la cantidad de dígitos correcta, 8 y 4 respectivamente. Los archivos que se encuentran dentro de esta carpeta son:

- C123456791011\_Lote1012: archivo para probar error en merchant code.
- c12345680\_Lote1016: archivo para probar error en el formato del nombre, la c se encuentra en minúscula.
- C12345680\_lote1016: archivo para probar error en el formato del nombre, la l se encuentra en minúscula.
- C12345681Lote1018: archivo para probar error en el formato del nombre, no posee '\_ '.
- Cabcdefg\_Lote1012: archivo para probar error en merchant code.
- C12345680\_Lote1: archivo para probar error en número de lote.
- C\_Lote1016: archivo para probar error en el merchant code.
- C12345680\_Lote: archivo para probar error en número de lote.

#### 2. TFDs \_erroneos:

Esta carpeta contiene archivos con errores en los registros TFD. A continuación se listan los archivos utilizados para probar estos casos negativos:

- C12345678\_Lote1014: archivo para probar error en el campo record type, posee un registro con valor 'TFG'.
- C12345679\_Lote1012: archivo para probar error en el record number. Un registro posee valor 69 en vez de 18.
- C12345680\_Lote1001: archivo para probar error en el id payment method, posee un valor 999.
- C12345680\_Lote1016: archivo para probar error en processing code. Un registro posee el valor 999999.
- C12345681\_Lote1018: archivo para probar error en el record type. Posee un registro con valor 'TGF'.
- C23456790\_Lote1000: archivo para probar error en el record number. Un registro posee valor 12 en vez de 11.
- C23456790\_Lote1002: archivo para probar error en el id payment method, posee un valor 'abc'.
- C23456790\_Lote1003: archivo para probar error en el record type. Posee un registro con valor 'TFH'.
- C23456790\_Lote1100:

- C12345679\_Lote1012: archivo para probar error en el record number. Un registro posee valor 75 en vez de 12.

### 3. TFHs \_erroneos:

Esta carpeta contiene archivos con errores en el registro TFH. A continuación se listan los archivos utilizados para probar estos casos negativos:

- C12345678\_Lote1014: archivo para probar error en el record type. El record type es 'TFC'.
- C12345679\_Lote1012: archivo para probar error en el record type. No posee un registro TFH.
- C12345680\_Lote1001: archivo para probar error en el merchant code. El merchant code del nombre del archivo es '12345680' mientras que el respectivo campo del registro TFH contiene el valor '11111111'.
- C12345680\_Lote1016: archivo para probar error en el number\_of\_trx\_records. El campo posee el valor '00000'.
- C12345681\_Lote1018: archivo para probar error en el number\_of\_trx\_records. El campo posee el valor '25' mientras que la cantidad de registros TFD son 20.
- C23456790\_Lote1000: archivo para probar error en el record type. El record type es 'TF'.
- C23456790\_Lote1002: archivo para probar error en el number\_of\_trx\_records. El campo posee el valor '11' mientras que la cantidad de registros TFD son 10.
- C23456790\_Lote1003: archivo para probar error en el merchant code. El merchant code del nombre del archivo es '23456790' mientras que el respectivo campo del registro TFH contiene el valor 'wrong'.
- C23456790\_Lote1100: archivo para probar error en el number\_of\_trx\_records. El campo posee el valor '100' mientras que la cantidad de registros TFD son 15.