



**Asignatura:**  
**75.08 Sistemas Operativos**

**Profesor: Ing. Osvaldo Clúa**  
**JTP: Lic. Sandra Abraham**  
**Ayudante: Lic. Adrian Muccio**  
**Ayudante: Lic. Guido Fernandez**

### **Trabajo práctico 1**

Nombre y apellido	Padrón	Correo electrónico
Julian Mejliker	100866	jmejliker@gmail.com
Lautaro Javier Ituarte	93639	lautaro.javier.ituarte@gmail.com
Joel Nicolas Saidman	99730	Joelsaidman1@gmail.com

**Segundo cuatrimestre 2019**

# Índice

<b>1. Autoevaluación</b>	<b>2</b>
1.1. Documentación conteniendo . . . . .	2
1.2. Paquete . . . . .	2
1.3. Script de Instalación . . . . .	2
1.4. Inicialización . . . . .	3
1.5. Proceso . . . . .	4
1.6. Start y Stop . . . . .	6
1.7. Evidencias . . . . .	6
<b>2. Instalador</b>	<b>7</b>
2.1. ¿Como instalar el programa? . . . . .	7
2.2. Reinstalación . . . . .	7
2.3. Pruebas del instalador . . . . .	8
2.4. Dificultades afrontadas . . . . .	8
<b>3. Inicialización</b>	<b>8</b>
<b>4. Start / Stop</b>	<b>9</b>
4.1. Hipótesis . . . . .	9
<b>5. Proceso</b>	<b>9</b>
5.1. Hipótesis . . . . .	9

## 1. Autoevaluación

### 1.1. Documentación conteniendo

1. Describa todas las hipótesis y aclaraciones que han asumido en la resolución.
2. Describa la lista de los problemas relevantes que se hayan presentado durante el desarrollo y como lograron resolverlos.
3. Liste la estructura resultante luego de una instalación estándar.
4. Describa los comandos auxiliares con una explicación de para qué y donde lo usa.
5. Describa los archivos auxiliares con una explicación de para qué y donde lo usa
6. Liste los nombres de archivos surgidos de los datos remitidos por la practica
7. Liste los nombres de los archivos de prueba creados por Uds. y para que se usan (que casos de prueba representan)
8. Cualquier otro ítem que quieran agregar (observaciones, recomendaciones, comentarios) que facilite el testeo

Cada parte del trabajo practico realizado tiene una sección en el presente informe que responderá estos ítems.

### 1.2. Paquete

1. Test: ¿Al extraer el paquete, se verifica la existencia de los directorios de trabajo los comandos y los archivos maestros?  
Si.
2. Test: comprobar que Ningún comando puede leer o grabar información del sistema fuera del directorio \$Grupo  
Si.
3. Test: comprobar que reserva los scripts y archivos originales en algún subdirectorio. Indique donde lo reserva

### 1.3. Script de Instalación

1. ¿Hay indicación correcta en el readme de cómo se ejecuta este script?  
Si.
2. Test: ¿el instalador detecta cuando el sistema nunca fue instalado y lo instala?  
Si.
3. Test: ¿el instalador detecta cuando el sistema está bien instalado y no lo instala?  
Si.
4. Test: ¿el instalador detecta cuando el sistema se debe reparar y lo repara?  
Si.

5. Test: ¿se aclara en las Hipótesis que controla el script para determinar que no está instalado, que está bien instalado, cuando es posible reparar? ¿cual es el nombre reservado para salvaguardar el original?  
Si, se aclara en la sección correspondiente al instalador en el presente informe.
6. Test: ¿el instalador permite Configurar los nombres de los directorios?  
Si.
7. Test: ¿el instalador propone valores default?  
Si.
8. Test: ¿el instalador evita nombres duplicados?  
Si.
9. Test: el instalador evita /conf y todo otro directorio reservado que se haya definido (esto debe estar aclarado en las hipótesis)  
Si.
10. Test: ¿el instalador procesa bien nombres de directorios con espacios intermedios?  
Si.
11. Test: ¿el instalador solicita confirmación de la instalación? ¿Confirma la instalación? (SI-NO): -  
Si. En caso que se ingrese algo distinto toma por defecto NO.
12. Test: ¿si la instalación no se confirma, reinicia el proceso y se tiene memoria de lo que fue cargando previamente el usuario?  
Si.
13. Test: ¿si la instalación se confirma, realiza la instalación y se depositan los comandos y maestros en los directorios correctos?  
Si.
14. Test: cuando la instalación se confirma, ¿recién allí crea los directorios?  
Si.
15. En el log: ¿se registra toda la interacción entre el usuario y el script tanto cuando la instalación es exitosa como cuando la instalación es cancelada o reparada? muestra el paso a paso de la creación de directorios y el movimiento de archivos  
Si.
16. ¿Se crea el archivo de configuración con el contenido solicitado?  
Si.

#### **1.4. Inicialización**

1. ¿Hay Indicación correcta en el readme de como se ejecuta este script?
2. Test: ¿el comando detecta cuando el sistema nunca fue inicializado y lo inicializa?  
Si
3. Test: ¿detecta cuando el sistema ya fue inicializado exitosamente y no reinicializa?  
Si

4. Test: ¿detiene la inicialización cuando falta algún componente? Directorios / archivos  
Si
5. Test: ¿corrige permisos?  
Si
6. Test: ¿setea las variables de ambiente y estas se exportan al siguiente script?  
Si
7. Test: ¿arranca el proceso? ¿Se muestra el process id?  
Si
8. En el log, ¿hay líneas del progreso de la inicialización?  
Si
9. ¿En el log, graba el pid cuando arranca el proceso? ESTO ES MUY IMPORTANTE PARA REALIZAR LOS TESTEOS  
Si
10. ¿En el log, hay líneas de “inicialización cancelada” con el motivo de cancelación?  
Si

### 1.5. Proceso

1. ¿Hay hipótesis indicando que separador de campos se emplea en el input / output ?  
Si, se utiliza la coma.
2. Test: ¿el proceso no ejecuta cuando el sistema nunca fue inicializado?  
No se ejecuta si no fue inicializado.
3. ¿Aclara en Hipótesis donde se realiza este control?  
Si, en el start.
4. Test: ¿se previene la ejecución de dos procesos simultáneos?  
Si.
5. ¿Aclara en Hipótesis como se realiza este control?  
Si.
6. Test: ¿cuenta los ciclos?
7. Test: ¿detecta las novedades?  
Si.
8. Test: ¿si el nombre es incorrecto porque tiene espacios el filename lo mueve a rechazados?  
Si.
9. Test: ¿si el nombre es incorrecto por cualquier otro motivo lo mueve a rechazados?  
Si.
10. Test: ¿si el nombre está bien pero el archivo esta vacío, lo mueve a rechazados?  
Si.
11. Test: si el nombre está bien pero el archivo no es un archivo regular, de texto (es una imagen, un pdf, etc) ¿lo mueve a rechazados?

12. Test: ¿si el nombre esta bien pero ya existe en /procesados un archivo con idéntico nombre (duplicado) lo mueve a rechazados?  
Si.
13. Test: ¿si e nombre esta bien y no viene vacío, es regular y no esta duplicado, lo mueve a aceptados?  
Si.
14. Test: ¿Cuándo procesa archivos? Siempre que haya algo para procesar (esto es lo correcto) o solo cuando el script acepta algo Siempre que haya algo para procesar.
15. Test: ¿acepta un archivo con solo el tráiler y 0 transacciones cuando esto es lo que indica el cierre de lote?  
Si.
16. Test: ¿rechaza un archivo sin tráiler?  
Si.
17. Test: ¿rechaza un archivo con tráiler incorrecto?  
Si, si el número de transacciones no coincide se rechaza.
18. Test: si rechaza el archivo ¿lo saca de aceptados y lo mueve a rechazados?  
Si.
19. Test: ¿arma el registro de cierre de lote correctamente?  
Si.
20. Test: ¿graba cierre de lote?  
Si.
21. Test: ¿arma el registro de transacciones correctamente?  
Si.
22. Test: el nombre externo de los archivos de transacciones ¿son correctos?  
Si.
23. Test: ¿graba transacciones?  
Si.
24. Test: ¿valida bien que el Código de Respuesta ISO 8583 exista en la tabla?  
Si.
25. Test: si el código no existe en la tabla, ¿explica en la hipótesis que se hace?  
Si.
26. Test: ¿mueve de aceptados a procesados?  
Si.
27. Test: ¿duerme y arranca nuevamente otro ciclo?  
Si.
28. En el log, ¿hay líneas del progreso de la ejecución?  
Si.

### 1.6. Start y Stop

1. ¿Hay Indicación correcta en el readme de cómo se ejecuta START?  
Si
2. ¿Hay Indicación correcta en el readme de cómo se ejecuta STOP?  
Si
3. ¿Funciona el Start?  
Si
4. ¿Funciona el Stop?  
Si
5. En el log, hay evidencia que realizaron varias pruebas de arranque - detención  
Si

### 1.7. Evidencias

1. ¿En el log, hay evidencia del proceso de los siguientes archivos de novedades? (todos deben contener al menos 1 registro con cualquier formato)

- Loter.01
- Lote 01
- Lote .001
- Lote .00
- Lote .199
- Lote01
- Lote01
- lote.01
- Miarchivo.Lote\_01
- Lote.02\_archi

Si, se puede encontrar en "test/evidencia/evidencia.1/log.txt"

2. ¿En el log, hay evidencia del proceso de los siguientes archivos de novedades? (con el contenido indicado)

- Lote\_02 vacío
- Lote\_03 conteniendo un pdf

Si, se puede encontrar en "test/evidencia/evidencia.2/log.txt"

3. ¿En el log, hay evidencia del proceso de los siguientes archivos de novedades? (con el contenido indicado)

- Lote\_04 conteniendo 3 registros sin trailer
- Lote\_05 conteniendo 2 registros de transacciones con tráiler indicando que tiene 3 registros

Si, se puede encontrar en "test/evidencia/evidencia\_3/log.txt"

4. ¿En el log, hay evidencia del proceso de los siguientes archivos de novedades? casos de prueba contruados por el grupo en donde
  - Se manda a procesar un lote donde un registro tiene un Código de Respuesta ISO 8583 inexistente
  - Se manda a procesar un lote donde el campo 4 viene sin datos
  - Se manda a procesar un lote donde el campo 5 viene sin datos
  - Se manda a procesar un lote donde el campo 6 viene sin datos
  - Se manda a procesar un registro con formato incorrecto

Si, se puede encontrar en "test/evidencia/evidencia\_4/log.txt"

5. ¿Envío todos los casos de prueba usados en las pruebas 1 2 3 4? Si.
6. ¿En el log, hay evidencia del proceso de los siguientes archivos de novedades? casos de prueba remitidos por la cátedra. Lote\_11 al 20  
Si, se puede encontrar en "test/evidencia/evidencia\_5/log.txt"
7. Verificar Líneas de log del procesamiento correcto de estos archivos
8. Verificar líneas de log del NO procesamiento de estos archivos  
El único que no proceso correctamente fue el 17 debido a que no cumple con nuestra hipótesis de tener comas solo para separar columnas

## 2. Instalador

### 2.1. ¿Como instalar el programa?

Para realizar la instalación del programa se deben seguir los siguientes pasos:

- Clonar el repositorio del trabajo.
- Abrir una terminal en la carpeta descargada y ejecutar el comando:

```
bash install.sh
```

- Seguir los pasos que indica el instalador una ves que se ejecuta

### 2.2. Reinstalación

En caso que el programa ya se encuentre instalado de manera defectuosa se puede iniciar la reparación mediante el siguiente comando:

```
bash install.sh -r
```

Si no esta seguro de que el programa se encuentre instalado correctamente puede correr el comando:



```
bash install.sh
```

y el instalador detectara automáticamente si se debe reparar el programa o se encuentra bien instalado. Al finalizar dará los detalles de la instalación.

Los errores que detecta son:

- Si faltan carpetas dentro de directorio GRUPO01
- Si el archivo config.txt no existe dentro del directorio GRUPO01/conf
- Si el archivo config.txt dentro del directorio GRUPO01/conf tiene información incorrecta en cuanto a la cantidad de variables (nombres de los directorios) que debe contener
- Si faltan archivos dentro del directorio GRUPO01/bin (el nombre bin puede variar dependiendo del que halla elegido el usuario).

### 2.3. Pruebas del instalador

Se realizaron test del instalador donde se tienen distintos escenarios para probar el mismo. Para correr estos test se debe abrir una terminal en el directorio:

```
../so7508\TP1\Grupo1\test\test\_instalador
```

y ejecutar el comando:

```
bash correr_pruebas.sh
```

Este script correrá todas las pruebas que hay disponibles. Las carpetas serán modificadas a lo largo de la ejecución por lo que servirá únicamente para la primera corrida.

### 2.4. Dificultades afrontadas

Durante el desarrollo del instalador las mayores dificultades fueron, por un lado tener que familiarizarse con la sintaxis de bash, y por otro lado probar los casos en que el instalador debía detectar y reparar la instalación del programa.

## 3. Inicialización

El proceso de inicialización, consiste en generar el ambiente adecuado para la corrida del proceso de su mejor manera. En esta instancia, se recopila la información de la instalación del archivo config.txt. Al obtener la información, se evalúa si los directorios, están correctos, o si falta alguno. También se evalúan los permisos de los archivos ejecutables, en el caso de ser incorrectos, se les asigna los permisos correspondientes.

Se puede dar la situación, de que el sistema deba reiniciarse, el sistema de manera autónoma, lo realizara, y le avisar al usuario. Si se da otro problema, se le avisa al usuario, cual es la forma de arreglar el problema presente.

Si la inicialización es correcta, el proceso se lanza, haciendo uso del proceso start, que sera explicado a continuación.

## 4. Start / Stop

El comando start, lo que realiza, es una evaluación de la inicialización, es decir que el ambiente este correctamente inicializado, para poder iniciar con el proceso general. También evalúa que no haya ningún otro proceso corriendo, si existe, notifica que no se puede ejecutar el proceso. Este comando es complementario con stop, el cual se encarga de detener la ejecución del proceso. Con respecto al segundo comando, se resolvió, buscando las ejecuciones en el momento, y filtrando por el proceso, y así poder matar el proceso. Este otro comando, también evalúa si es que el proceso esta corriendo o no.

Estos comandos, fueron fácil de implementar, no se me presento ninguna situación de alta complejidad que me trabo en el desarrollo de los mismos.

En el comando start, se evalúa si el sistema fue inicializado, a través de una variable de ambiente llamada init. Si esta está seteada en uno, significa la correcta inicialización. De forma contraria, no se correrá el proceso. No se evalúa.

En los logs correspondientes se puede ver distintas corridas que muestran las posibles situaciones con respecto a los comandos en cuestión. Por ejemplo, que nos e corrió debido a que ya hay alguno corriendo o que no se pudo parar el proceso, ya que no había ninguno corriendo.

### 4.1. Hipótesis

- El comando start, no evalúa ningún cambio en el ambiente, con esto me refiero a que se da por sentado que todo esta igual a cuando se hizo la inicialización.

## 5. Proceso

El procesamiento se lleva a cabo en dos partes, lo que llamamos 'pre procesamiento' y el 'procesamiento de salida'. El procesamiento solo comenzará si el ambiente esta inicializado.

El pre procesamiento hace las validaciones básicas sobre los lotes que llegan a 'Novedades' (validaciones de nombre de archivo, que no este vacío, tipo de archivo) y en base a eso traslada los lotes a 'Aceptados' o a 'Rechazados'.

El procesamiento de salida toma los lotes que quedaron en 'Aceptados' para hacer validaciones de contenido del archivo (formato del archivo, existencia de un trailer) y en base a eso puede enviarlos a 'Rechazados' o generar los archivos de salida (Cierre de lote y archivo de transacciones) y pasar el lote a 'Procesados'.

### 5.1. Hipótesis

- Si se encuentra un problema con una transacción en particular el proceso desechará dicha transacción y continuará con el procesamiento del resto.
- Se utiliza la coma como separador de columnas.
- Al momento de buscar el código en el archivo, si este no existe, se continua el procesamiento del resto de las transacciones del documento.