



Proyecto: Trabajo Practico Sistemas Operativos	Grupo N°: x	Turno: 22/05/2018 hh hs
Integrantes: Padrón, Nombre y Apellido	Correo Electrónico	

		OK	NOK	OBS
<b>Documentación</b>				
Punto 1.	Hipótesis			
Punto 2.	Readme			
Punto 3.	Auxiliares			
Punto 4.	Carpeta			
<b>Paquete Instalación</b>				
Punto 1.	paquete de instalación			
Punto 2.	directorio de trabajo			
Punto 3.	Subdir de configuración			
Punto 4.	Archivo Readme			
<b>InstalO</b>				
Punto 1.	Verificar Perl			
Punto 2.	opción -r			
Punto 3.	Configurar directorios			
Punto 4.	confirmación de instalación			
Punto 5.	Archivo de configuración			
Punto 6.	Grabar archivo de log			
<b>IniciO</b>				
Punto 1.	Verificar instalación completa			
A.	VERIFICAR DIRECTORIOS			
B.	VERIFICAR / CORREGIR PERMISOS			
Punto 2.	Setear variables de ambiente			
Punto 3.	arrancar el demonio			
Punto 4.	Grabar archivo de Log			
<b>DetectO</b>				
Punto 1.	Ambiente			



Punto 2.	Grabar archivo de Log			
Punto 3.	Demonio			
Punto 4.	Contar los Ciclos			
Punto 5.	Ver si hay novedades			
PUNTO 6.	Validar el nombre			
Punto 7.	Verificación del archivo			
PUNTO 8.	Rechazar			
PUNTO 9.	Aceptar			
PUNTO 10.	Invocar al Intérprete			
PUNTO 11.	Movimiento de archivos			
InterpretO				
PUNTO 1.	Ambiente			
PUNTO 2.	Grabar archivo de Log			
PUNTO 3.	Leer archivos a procesar			
PUNTO 4.	No procesar el mismo archivo			
PUNTO 5.	Interpretar cada registro de input			
PUNTO 6.	Armar el registro de Salida			
PUNTO 7.	Grabar la Salida			
PUNTO 8.	Conservar el archivo procesado			
PUNTO 9.	Movimiento de archivos			
ReportO				
PUNTO 1.	Recomendación			
PUNTO 2.	Listar Divergencias en tanto %			
PUNTO 3.	Listar Divergencias en tantos \$			
Datos	TRAER NOVEDADES DE PRUEBA?			



## INDICE

<b>Enunciado – Tema O .....</b>	<b>3</b>
Producto: Documentación .....	3
Producto: Paquete de Instalación.....	4
Producto: Instalación: <b>Instalo</b> .....	4
Producto: Inicializador de ambiente: <b>IniciO</b> .....	7
Producto: Detector de novedades <b>DetectO</b> .....	9
Producto: Intérprete de archivos: <b>Interpreto</b> .....	12
Producto: Consultas y Reportes: <b>ReportO</b> .....	14
<b>Archivos y Estructuras.....</b>	<b>17</b>

NOTA: El texto incluido en rectángulos grises *y el exhibido en cursiva azul* se incluye con el fin de proporcionar una guía y debe ser eliminado antes de publicar / imprimir el documento

## Enunciado – Tema O

El área de control de inversiones necesita un sistema software denominado CONTROLLO que le permita procesar archivos enviados por las subsidiarias de la empresa (ubicadas en distintos países, que usan distintos sistemas) con información de préstamos personales impagos.

Estos archivos no poseen un formato uniforme, por lo tanto, se deben interpretar antes de poder procesarlos.

Al procesarlos se debe generar un archivo único por país que luego se emplea para reportes.

### **Producto: Documentación**

El día de corrección traer la documentación

#### **Punto 1. HIPÓTESIS**

Indique aquí todas las hipótesis que han asumido en la resolución del TP

Indique aquí todas las aclaraciones que considere necesarias

Problemas relevantes: Comparta aquí los problemas relevantes que se hayan presentado durante el desarrollo y como lograron resolverlos.

#### **Punto 2. README**

Copie aquí el contenido del archivo README solicitado como parte del paquete de instalación

#### **Punto 3. AUXILIARES**

Opcionalmente, si ha creado nuevos comandos, escriba aquí el listado de comandos auxiliares con una explicación de para que y donde lo usa

Opcionalmente, si ha creado nuevos archivos, escriba aquí el listado de archivos auxiliares con una explicación de para qué y donde lo usa

#### **Punto 4. CARPETA**

Complete la primera hoja y el pie de página con los datos del grupo de TP

Complete los puntos antes mencionados

Actualice la tabla de índice

Imprima las primeras dos Hojas DOS veces

Imprima el resto 1 sola vez

Entregue todo en una Carpeta



## **Producto: Paquete de Instalación**

### **Punto 1. PAQUETE DE INSTALACIÓN**

El paquete de instalación deberá estar contenido en un **único archivo instalable** en formato ".tgz" con todos los archivos y directorios empaquetados en un archivo "tar" y luego comprimido con "gzip".

El paquete debe tener los comandos y los datos, tanto los datos provistos por la catedra, como otros juegos de datos de prueba generados por el grupo.

No debe contener resultados de ejecuciones previas.

### **Punto 2. DIRECTORIO DE TRABAJO**

Toda la instalación debe realizarse bajo un mismo **directorio de trabajo** llamado Grupoxx donde xx es su número de grupo

Ningún comando puede leer o grabar información del sistema fuera de este directorio

Todo el camino (path) que va desde la raíz hasta Grupoxx lo denominaremos genéricamente en esta explicación \$GRUPO

### **Punto 3. SUB DIRECTORIO DE CONFIGURACIÓN**

La extracción del paquete debe **generar al menos** en el directorio trabajo un subdirectorio denominado **dirconf**

Este será el único directorio del sistema con nombre prefijado

### **Punto 4. ARCHIVO README**

Generar un archivo de texto llamado README con las instrucciones de descarga, los requisitos del sistema, las instrucciones de instalación, las instrucciones de ejecución y cualquier aclaración que se considere necesaria para asegurar el éxito de la revisión.

- Una explicación de cómo descargar el paquete
- Una explicación de cómo descomprimir, crear directorio del grupo, etc
- Una explicación de lo que se crea a partir de la descompresión
- Una explicación sobre que se requiere para poder instalar y/o ejecutar el sistema
- Instrucciones de instalación del sistema
- Una explicación de cómo se hace una instalación o reparación de la instalación
- Que nos deja la instalación y dónde
- Cuáles son los primeros pasos para poder ejecutar el sistema
- Como arrancar o detener comandos
- Cualquier otra indicación, diagrama, cuadro que considere adecuada

## **Producto: Instalación: *InstalO***

ANTES DE EMPEZAR: asegúrese de contar con la última versión del sistema antes de iniciar la instalación

El propósito de este comando es efectuar la instalación del sistema

Debe ser desarrollado en Shell script

El comando debe soportar los siguientes escenarios

- El sistema nunca fue instalado (se deberá **instalar el sistema**)
- El sistema está bien instalado (no se deberá instalar el sistema, solo mostrar datos de la instalación)
- El sistema está instalado, pero está incompleto (se deberá **reparar<sup>1</sup> la instalación** siempre que se pueda)
  - Puede asumir: si el archivo de configuración no existe, el sistema no está instalado, y si existe el sistema está instalado. Indicarlo en Hipótesis

<sup>1</sup> cree un subdirectorio reservado para la instalación para salvaguardar el paquete original (completo) así podrá reparar el sistema. También le servirá para tener el set de datos de prueba. Adviértalo en las Hipótesis.



### **Punto 1. VERIFICAR PERL**

Chequear que perl versión 5 o superior este instalado, si no está instalado informarlo y terminar la instalación con error.

### **Punto 2. OPCIÓN -R**

Permitir ejecutar el script de instalación con la opción -r para reparar la instalación.  
Puede definir más opciones de instalación, las cuales debe documentar adecuadamente en el README

### **Punto 3. CONFIGURAR LOS NOMBRES DE LOS DIRECTORIOS**

Solicitar al usuario que defina

- 1) El directorio de ejecutables
- 2) El directorio de archivos maestros y tablas del sistema (aquí no van los datos de prueba!!)
- 3) El directorio de arribo de archivos externos, es decir, los archivos que remiten las subsidiarias
- 4) El directorio donde se depositan temporalmente las novedades aceptadas
- 5) El directorio donde se depositan todos los archivos rechazados
- 6) El directorio donde se depositan los archivos procesados
- 7) El directorio donde se depositan los reportes
- 8) El directorio donde se depositan los logs de los comandos<sup>2</sup>

En todos los casos, cuando se interactúa con el usuario se debe proponer un valor por default para el nombre del directorio mostrándolo entre paréntesis. Ejemplo:

*Defina el directorio de ejecutables (\$grupo/bin):*

Evitar nombres de directorios duplicados (dos directorios con igual nombre)

El directorio `dirconf` (creado en la descompresión del paquete) se considera **nombre reservado**, es decir que el usuario no puede escoger este nombre para un directorio de instalación.

Permitir subdirectorios, pe: *\$grupo/scripts/ejecutables*

No ir creando directorios hasta no tener **CONFIRMACIÓN DE LA INSTALACIÓN**

Todo lo que se muestra al usuario por pantalla y sus respuestas debe también grabarlas en el log

### **Punto 4. CONFIRMACIÓN DE INSTALACIÓN**

Cuando el usuario termina de configurar los nombres de directorios, mostrar los valores que ingresó, como va a quedar la estructura, y solicitarle que confirme para continuar

*Ejemplo:*

*TP SO7508 Primer Cuatrimestre 2018. Tema O Copyright © Grupo xx*

*Librería del Sistema: dirconf*

*Ejecutables en: ....*

*Directorio para los archivos maestros: ...*

*Directorio para el arribo de archivos externos: ...*

*Directorio para los archivos aceptados...*

*Directorio para los archivos rechazados...*

*Directorio para Archivos procesados...*

*Directorio para los reportes*

*Logs de auditoria del Sistema: ...*

*Estado de la instalación: LISTA*

---

<sup>2</sup> Todos los logs se graban en este directorio excepto el log de la instalación que se graba en `dirconf`



¿Confirma la instalación? (SI-NO): \_

**Si el usuario indica No**, volver a solicitar los directorios, pero esta vez mostrando como default lo recientemente ingresado, en lugar de los propuestos inicialmente (es decir, que **el script debe tener memoria** de los valores ingresados previamente)

**Si el usuario indica Si** crear la estructura de directorios definida y mover al directorio de maestros, los archivos maestros y las tablas, al de ejecutables los ejecutables

A medida que se hace la creación de directorios y su llenado **informarlo mediante mensajes** (por pantalla y en el log de la instalación)

### **Punto 5. GRABAR ARCHIVO DE CONFIGURACIÓN**

Nombre a elección, extensión: .conf directorio \$GRUPO/dirconf

*Ejemplo: \$GRUPO/dirconf/instalo.conf*

Como resultado de la instalación además de crear los directorios y mover los archivos se debe grabar un nuevo archivo con la configuración seteada. Este archivo es el input para el script de inicialización. Se debe grabar un registro por cada nombre de directorio usado por la instalación con el siguiente formato: Identificador\_del\_directorio-Valor-Usuario-Fecha  
Puede incluir la variable GRUPO para registrar el valor de \$grupo

Ejemplo:

*ejecutables-/usr/sandra/temp/grupo01/bintp-Sandra-09/09/2017 10:03 p.m*

*maestros-/usr/sandra/temp/grupo01/maetp-Sandra-09/09/2017 10:03 p.m*

*aceptados-/usr/sandra/temp/grupo01/aceptp-Sandra-09/09/2017 10:03 p.m*

*rechazados-/usr/sandra/temp/grupo01/rechtp-Sandra-09/09/2017 10:03 p.m*

*arribos-/usr/sandra/temp/grupo01/ftp-Sandra-09/09/2017 10:03 p.m*

*reportes-/usr/sandra/temp/grupo01/repotp-Sandra-09/09/2017 10:03 p.m*

*logs-/usr/sandra/temp/grupo01/logtp-Sandra-09/09/2017 10:03 p.m*

Luego de estos registros puede agregar todos los registros que necesite (para otros directorios, para variables de secuencia, para \$grupo, etc.)

Estructura del archivo de configuración		Separador de campos: - (guion medio)
ID	campo	Descripción/fuente/valor
1.	Id	Identificador del directorio, el identificador debe ser explicativo de su contenido, es decir, no usar nombres tales como VAR1, VAR2 etc.,
2.	Valor	Ruta del directorio
3.	Usuario	Es el login del usuario que graba el registro
4.	Fecha	Formato a Elección. Es la fecha y hora en el momento de grabación del registro.

### **Punto 6. GRABAR ARCHIVO DE LOG<sup>3</sup>**

Durante la instalación se debe grabar un archivo de log, es decir, desde que empieza a ejecutar la instalación debe ir registrando todos los eventos en un log.

Indicar claramente el inicio y el fin del comando grabando un mensaje en cada caso

<sup>3</sup> ¿Qué es un log?

Un log es un registro oficial de eventos durante un periodo de tiempo en particular.

Es usado para registrar información sobre cuándo, quién, dónde, qué y por qué un evento ocurre para una aplicación, proceso o dispositivo.

A estos 5 valores se los llama estándar W5, por su origen en inglés: when, who, where, what and why.



El log de la instalación debe ser sumamente exhaustivo y registrar que se está ejecutando (*instalación, reinstalación*), que paso se está ejecutando (*creando estructuras, moviendo archivos*), que resultado se obtuvo luego de algún control (*alerta! nombre de directorio duplicado*) y todo el diálogo entre el script y el usuario (*Defina el directorio de ejecutables (\$grupo/bin): /ejecutables*)

**El log de la instalación no debe reescribirse**, aun cuando se produce cancelación del script o reinstalación

Nombre a elección, extensión .log directorio \$GRUPO/dirconf

*Ejemplo: \$GRUPO/dirconf/instalo.log*

FORMATO: Cada registro de log debe cumplir con el estándar w5. Se debe grabar un registro por cada mensaje

Estructura del archivo de log		Separador de campos: - guion medio
	campo	Descripción/fuente/valor
1.	Fecha	Formato a Elección. Es la fecha y hora en el momento de grabación del registro.
2.	Usuario	Es el login del usuario que graba el registro
3.	Origen	Es el nombre del Comando, función o rutina que produce el evento que se registra en el log
4.	Tipo de Error	Indica que tipo de error es: Informativo (INF), Alerta (ALE) Error (ERR)
5.	Mensaje	Mensaje de log propiamente dicho

*Ejemplo: 20170905 19:53:22-Sandra-InstalO-INF-Defina el directorio de ejecutables (\$grupo/bin): /ejecutables*

NOTA: esta estructura de archivo es estándar para todos los logs del sistema

### **Producto: Inicializador de ambiente: *IniciO***

El propósito de este comando es preparar el entorno de ejecución del TP (ambiente) y asegurar que estén dadas todas las condiciones para la ejecución del sistema

Debe ser desarrollado en Shell script

Preveer los siguientes escenarios:

- El sistema nunca fue inicializado (se deberá inicializar el sistema y ejecutar el demonio)
- El sistema ya fue inicializado exitosamente (se deberá ejecutar el demonio siempre y cuando ya no esté corriendo)
- El sistema no puede ser inicializado (se deberá explicar el motivo y terminar)

#### **Punto 1. VERIFICAR QUE LA INSTALACIÓN ESTE COMPLETA**

El Proceso se inicia con el aseguramiento de la disponibilidad de la información para llevar adelante el proceso total: Es indispensable contar con los comandos, archivos maestros y las tablas de configuración con los permisos adecuados

##### **a. VERIFICAR DIRECTORIOS**

Leer el archivo de configuración que deja la instalación y verificar que todos los directorios requeridos existan: Ejecutables, Maestros, Arribos, Aceptados, Rechazados, Procesados, Reportes, Logs



Si se detecta que alguno de estos directorios no existe:

- Informa al usuario de la situación, que es lo que está faltando
- le brinda una explicación de los pasos que debe seguir para reparar la instalación
- termina

#### **b. VERIFICAR / CORREGIR PERMISOS**

Verifica que los archivos del directorio maestro tengan permisos de lectura, sino los corrige  
Verifica que los archivos del directorio ejecutables tengan permisos de ejecución, sino los corrige

mostrar (y grabar en el log) los permisos que ya estaban ok  
mostrar (y grabar en el log) los permisos que fueron corregidos

#### **Punto 2. SETEAR VARIABLES DE AMBIENTE**

a partir de la información de configuración que deja la instalación crear las variables de ambiente.  
Estas variables deben permanecer durante toda la ejecución del sistema y **ser accesibles por todos los comandos desencadenados a partir de éste.**

El único que lee el archivo de configuración es este script, el resto de los comandos deben poder leer las variables de ambiente creadas aquí.

mostrar (y grabar en el log) las variables creadas

#### **Punto 3. ARRANCAR EL DEMONIO**

Invocar al script **DetectO**, mostrar (y grabar en el log) el process id del demonio, mostrar al usuario como debe hacer para detener el demonio → crear un comando llamado **StopO** para esto

Nunca Invocar al script **DetectO**, si ya hay otro corriendo; **siempre mostrar** (y grabar en el log) **el process id** del demonio que está corriendo.

#### **Punto 4. GRABAR ARCHIVO DE LOG<sup>4</sup>**

Cada registro de log debe cumplir con el estándar w5. Cada mensaje es un nuevo registro, los mensajes deben servir para guiarnos sobre el curso de ejecución del script (loguear que se está haciendo: *permisos seteados, variables de ambiente definidas, Demonio invocado*) y mostrar los resultados obtenidos (*demonio corriendo bajo el numero xx*), que resultado tuvo algún control (*error! no existe el comando de validación, error! falta el maestro de cuentas*)

mostrar (y grabar en el log) cuando se terminó el IniciO

varios comandos deben grabar en el log

Si crea una función donde indica nombre del comando, tipo de mensaje y mensaje puede usarla en varios comandos

Nombre a elección, extensión .log directorio indicado por el usuario en el momento de la instalación  
*Ejemplo: \$GRUPO/logtp/inicio.log*

Estructura del archivo de log		Separador de campos: - guion medio
	campo	Descripción/fuente/valor

<sup>4</sup> la estructura de archivo es estándar para todos los logs del sistema





6.	Fecha	Formato a Elección. Es la fecha y hora en el momento de grabación del registro.
7.	Usuario	Es el login del usuario que graba el registro
8.	Origen	Es el nombre del Comando, función o rutina que produce el evento que se registra en el log
9.	Tipo de Error	Indica que tipo de error es: Informativo (INF), Alerta (ALE) Error (ERR)
10.	Mensaje	Mensaje de log propiamente dicho

### **Producto: Detector de novedades DetectO**

El propósito de este comando es inspeccionar el directorio de arribos, si hay algún archivo analizarlo superficialmente (sin leerlo) y determinar si lo acepta o rechaza.

Además, debe invocar (si corresponde) al comando que interpreta la estructura de los registros de novedades, dormir un tiempo x, y volver a empezar.

Es un proceso de ejecución eterna. Si se lo quiere detener, se usa el comando **StopO**

Debe ser desarrollado en Shell script

Prever los siguientes escenarios:

- El sistema nunca fue inicializado (explicar y terminar)
- El sistema está bien inicializado (ejecutar)

### **Punto 1. AMBIENTE**

Ningún comando puede ser ejecutado sin la inicialización del ambiente

### **Punto 2. GRABAR ARCHIVO DE LOG<sup>5</sup>**

Como el demonio es un script de ejecución eterna, se debe **evitar que este log crezca indefinidamente**

No se exige ningún método específico, adopte el de su preferencia, aclarándolo debidamente en las hipótesis generales del sistema. Si desea implementar la misma política de control para todos los logs, también puede hacerlo

Evite borrar todo todo el tiempo, permita que alguna pista de los últimos eventos quede registrada

A continuación, un par de ejemplos que puede usar para hacer el control:

- 1) cada 100 ciclos se truncan 90 ciclos, es decir nunca hay más de 100 ciclos registrados
- 2) cuando el log supera un tamaño x (x = 300 KB por ejemplo, así es fácil evaluarlo en clase) se trunca dejando las últimas 50 líneas y escribiendo el mensaje "log truncado"

Los mensajes deben servir para guiarnos sobre el curso de ejecución del script (loguear el número de ciclo y qué se está haciendo: *Ciclo Numero 1, Novedad rechazada: <nombre> + motivo del rechazo, Novedad aceptada <nombre>, Validador invocado: process id xxx, Invocación del Validador pospuesta para el siguiente ciclo*)

varios comandos deben grabar en el log

<sup>5</sup> Un log es un registro oficial de eventos durante un periodo de tiempo en particular.

Es usado para registrar información sobre cuándo, quién, dónde, qué y por qué un evento ocurre para una aplicación, proceso o dispositivo.

A estos 5 valores se los llama estándar W5, por su origen en inglés: when, who, where, what and why.



Si crea una función donde indica nombre del comando, tipo de mensaje y mensaje puede usarla en varios comandos

Nombre a elección, extensión .log directorio indicado por el usuario en el momento de la instalación

*Ejemplo: \$GRUPO/logtp/detecto.log*

Estructura del archivo de log		Separador de campos: - guion medio
	campo	Descripción/fuente/valor
11.	Fecha	Formato a Elección. Es la fecha y hora en el momento de grabación del registro.
12.	Usuario	Es el login del usuario que graba el registro
13.	Origen	Es el nombre del Comando, función o rutina que produce el evento que se registra en el log
14.	Tipo de Error	Indica que tipo de error es: Informativo (INF), Alerta (ALE) Error (ERR)
15.	Mensaje	Mensaje de log propiamente dicho

### **Punto 3. DEMONIO**

Este comando es un proceso del tipo demonio, daemon o dæmon (de sus siglas en inglés Disk And Execution Monitor).

Se ejecuta en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo).

Cada vez que termina su trabajo duerme un tiempo "x" y vuelve a empezar, es decir, que a menos que se detenga con **StopO**, este proceso no tiene condición de fin.

### **Punto 4. CONTAR LOS CICLOS**

Se debe mantener un contador de ciclos y grabarlo en el log

### **Punto 5. VER SI HAY NOVEDADES**

buscar las novedades en el directorio de arribos indicado en la variable de ambiente correspondiente

Si existen archivos, validar el nombre, sino seguir en **INVOCAR AL INTERPRETE**

Las novedades consisten en archivos remitidos por las subsidiarias de la empresa con información de préstamos personales impagos. Pueden arribar archivos provenientes de varios países, sistemas, años o meses

### **Punto 6. VALIDAR EL NOMBRE**

Para determinar si un archivo es válido o no, se debe verificar su nombre. Los archivos válidos poseen nombres con este formato: <país>-<sistema>-<año>-<mes>

- <pais >, corresponde a un código de país
- <sistema>, corresponde a un código de sistema
- La combinación código de país + código de sistema debe existir en el **maestro de Países y Sistemas (p-s.mae)**
- <año>, corresponde a un año mayor a 2016. El formato del año es aaaa
- <mes>, corresponde al valor de un mes (entre 1 y 12), El formato del mes es mm.
- A la combinación año-mes se la denomina: PERIODO, el periodo no puede ser mayor al periodo corriente



Ejemplos de nombres de archivos validos

[A-6-2017-06](#)

[C-7-2017-06](#)

[A-6-2017-12](#)

[A-7-2018-04](#)

Si el nombre del archivo es válido, realizar la VERIFICACIÓN DEL ARCHIVO

Si el nombre del archivo no es válido, RECHAZAR

### **Punto 7. VERIFICACIÓN DEL ARCHIVO**

Si el archivo viene vacío, RECHAZAR

Si el archivo no es un archivo regular, de texto (si es una imagen, un comprimido, etc), RECHAZAR  
si no está vacío y es un archivo regular, ACEPTAR

### **Punto 8. RECHAZAR**

Si el archivo NO supera las validaciones precedentes, **mover el archivo rechazado** al directorio de rechazados indicado en la variable de ambiente correspondiente  
grabar en el log el nombre del archivo que no superó la validación y cuál ha sido el motivo del rechazo, ejemplos:

- *Novedad rechazada: <nombre>*
- *Motivo Archivo vacío, Motivo Tipo de archivo invalido, Motivo Formato de filename incorrecto, Motivo país inexistente, Motivo sistema inexistente, Motivo año invalido, Motivo mes invalido, Motivo periodo adelantado*

### **Punto 9. ACEPTAR**

Si el archivo supera las validaciones precedentes, **mover el archivo aceptado** al directorio de aceptados indicado en la variable de ambiente correspondiente  
grabar en el log el nombre del archivo aceptado y la leyenda "Aceptado"

### **Punto 10. INVOCAR AL INTÉRPRETE**

Invocar al script [InterpretO](#) sólo si hay archivos en el directorio de aceptados y sólo si no hay otro comando [InterpretO](#) ejecutándose (evitar la múltiple invocación)

Si se invoca, grabar en el log el process id del interprete

Si había que invocar (porque hay archivos en el directorio de aceptados) pero no se puede porque ya hay otro corriendo, registrar en el log "*Invocación del Interprete pospuesta para el siguiente ciclo*"

### **Punto 11. MOVIMIENTO DE ARCHIVOS**

En líneas generales, para todo el TP, nada se borra, se lo mueve de un lugar a otro: A aceptados, a rechazados, a procesados.

Siempre que movemos archivos, lo hacemos con el mismo nombre con el que vino.

¿Pero qué sucede si al moverlo ya existe en ese destino otro igual?

Cuando se intenta mover un archivo y ya existe en ese lugar otro del mismo nombre, no se debe sobrescribir ni debe fracasar la operación, se deben poder conservar ambos.

No se exige ningún método específico, adopte el de su preferencia, aclarándolo debidamente en las hipótesis generales del sistema.

A continuación, un ejemplo que puede usar para lograr esto:

En la primera colisión, crear un subdirectorio /dup para depositar los duplicados, y mover allí el archivo, pero con el nombre modificado: modificar el nombre del archivo agregándole una secuencia numérica conservada, es decir, una secuencia que no se destruye al cerrar la sesión, sino que se conserva durante toda la existencia del sistema

De esta forma, todos los archivos con nombre repetidos se pueden conservar.



El comando InterpretO también debe mover de un lugar a otro  
Si crea una función donde indica nombre del archivo, origen, destino, puede usarla en ambos

### **Producto: Intérprete de archivos: InterpretO**

El propósito principal de este comando es INTERPRETAR CADA REGISTRO DE INPUT, ARMAR EL REGISTRO DE SALIDA y GRABAR LA SALIDA

Debe ser desarrollado en Shell script

Prever los siguientes escenarios:

- El sistema nunca fue inicializado (terminar)
- El sistema está bien inicializado (ejecutar)
- El archivo no fue procesado (procesar)
- El archivo ya fue procesado (informar que esta duplicado)

Los archivos aceptados contienen información de préstamos personales impagos generada por los diferentes sistemas de las subsidiarias que operan en distintos países. Debido a esto, **no poseen una estructura estándar**. La misma se encuentra **definida a través de las tablas de separadores y campos que tiene el sistema**. Para realizar la lectura de los archivos se debe interpretar su información a través de estas tablas y generar una salida que pueda ser usada para las consultas, reportes y recálculos

#### **Punto 1. AMBIENTE**

Ningún comando puede ser ejecutado sin la inicialización del ambiente

#### **Punto 2. GRABAR ARCHIVO DE LOG<sup>6</sup>**

Cada registro de log debe cumplir con el estándar W5. Cada mensaje es un nuevo registro, los mensajes deben servir para guiarnos sobre el curso de ejecución del script (loguear que archivo se está leyendo, procesando, rechazando: *Archivo rechazado <nombre> + motivo del rechazo*, *Archivo procesado <nombre>* y mostrar los resultados obtenidos a nivel de registro (*registro n°1: aceptado*, *registro n°2: error! y motivo por el cual el registro es erróneo*). Cuando se termina de procesar un archivo mostrar total de registros leídos, cuantos fueron aceptados y cuantos rechazados.

El archivo de log se debe grabar en el directorio de log indicado por el usuario. El nombre debe ser InterpretO.log

#### **Punto 3. LEER ARCHIVOS A PROCESAR**

los archivos que procesar son los que están en el directorio de aceptados indicado en la variable de ambiente correspondiente

#### **Punto 4. NO PROCESAR MÁS DE UNA VEZ EL MISMO ARCHIVO**

Se considera que dos archivos son el mismo si poseen el mismo filename

Como se debe evitar procesar más de una vez el mismo archivo, chequear antes si en el subdirectorio del día<sup>7</sup> ya hay otro del mismo nombre. Si hay, entonces se debe suponer que el archivo ya fue procesado, rechazarlo y seguir con el siguiente

- grabar en el log que el archivo fue rechazado por encontrarse **duplicado**

<sup>6</sup> Un log es un registro oficial de eventos durante un periodo de tiempo en particular.

Es usado para registrar información sobre cuándo, quién, dónde, qué y por qué un evento ocurre para una aplicación, proceso o dispositivo.

A estos 5 valores se los llama estándar W5, por su origen en ingles: when, who, where, what and why.

<sup>7</sup> Cuando se procesa un archivo se este comando lo mueve a un subdirectorio del directorio de procesados llamado aaaammdd (un subdirectorio por día)



## **Punto 5. INTERPRETAR CADA REGISTRO DE INPUT**

### **A. Determinar Separadores**

Se deben recuperar de la tabla T1 los valores de separador de campo y separador decimal para el país y sistema correspondiente.

La tabla de Separadores o Tabla T1 nos dice cuál es el separador de campo y cuál es el separador decimal que usa el archivo de input según el país y el sistema de origen.

### **B. Determinar Campos**

Se debe recuperar de la tabla de Campos o Tabla T2 la estructura del registro de input y el formato de los campos que nos interesan leer según el país y el sistema de origen.

La tabla de Campos o Tabla T2 nos dice que campos, en qué orden de aparición vamos a encontrar esos campos en los archivos de préstamos personales recibidos y con qué formato viene informado.

#### El formato de los campos de fecha:

- El formato de las fechas puede ser: ddmmyyy<longitud>. O yyymmdd<longitud>.

Donde dd corresponde a dos caracteres para el día, mm corresponde a dos caracteres para el mes y yy corresponde a cuatro caracteres para el año. La interpretación de la longitud es la siguiente: Si la longitud es 8, no posee separador entre día, mes y año, si la longitud es 10, posee separador entre día, mes y año. El separador de fecha lo desconocemos. Solo podemos asegurar que nunca es igual al separador de campos ni al separador decimal.

#### El formato de los campos alfanuméricos

- El formato de los alfanuméricos es: \$<longitud>.

El \$ nos indica que el formato es alfanumérico y <longitud> indica la longitud máxima del campo.

#### El formato de los campos numéricos

- El formato de los numéricos es: commax<longitud parte entera>.<longitud parte decimal>

El commax nos indica que el formato es numérico, <longitud parte entera> nos indica la longitud máxima de la parte entera, <longitud parte decimal > nos indica la longitud máxima de la parte decimal.

### **C. Leer el registro**

Se debe leer el registro de input interpretando la estructura de acuerdo con los separadores obtenidos en A y los campos determinados en B

## **Punto 6. ARMAR EL REGISTRO DE SALIDA**

Construir el registro de préstamo de acuerdo con la estructura indicada para los archivos

### **Préstamos Personales Por País**

Interpretar la fecha CTB\_FE y determinar día mes y año a grabar

Interpretar los montos y llevarlos al formato estándar para grabarlos.

Todos los campos de montos deben ser grabados en cero si no están informados

#### Cálculo de monto restante

El campo de monto restante debe ser calculado como la suma de los montos PRES, IMPAGO, INDE, INNODE y deducir el monto debitado DEB

$$MT\_REST = MT\_PRES + MT\_IMPAGO + MT\_INDE + MT\_INNODE - MT\_DEB$$

## **Punto 7. GRABAR LA SALIDA**

Según sea el país de que se trate, grabar el archivo en el directorio procesados con el nombre **PRESTAMOS.<país>**

Si el Archivo ya existe, agregarle los nuevos registros. Si no existe, crearlo.

Este archivo debe usar:



- como separador de campos siempre el punto y coma
- como separador decimal siempre la coma
- como formato de fecha siempre dd/mm/aaaa

este mismo estándar es el que tiene el archivo PPI.mae

### **Punto 8. CONSERVAR EL ARCHIVO PROCESADO**

Cada día se crea un nuevo subdirectorio en el directorio de procesados indicado en la variable de ambiente correspondiente en donde se depositan las novedades procesadas. Cuando cambia el día, se crea otro subdirectorio

Al mover las novedades a procesados, respete la política de manejo de duplicados, en el siguiente punto se aclara lo referente a movimiento de archivos.

Escriba en las hipótesis cual es el formato de fecha que se usara para crear estos directorios

*Ejemplo: /mipc/miusuario/mitp/procesados/12042018/A-6-2017-06*

### **Punto 9. MOVIMIENTO DE ARCHIVOS**

En líneas generales, para todo el TP, nada se borra, se lo mueve de un lugar a otro: A aceptados, a rechazados, a procesados.

Siempre que movemos archivos, lo hacemos con el mismo nombre con el que vino.

¿Pero qué sucede si al moverlo ya existe en ese destino otro igual?

Cuando se intenta mover un archivo y ya existe en ese lugar otro del mismo nombre, no se debe sobrescribir ni debe fracasar la operación, se deben poder conservar ambos.

No se exige ningún método específico, adopte el de su preferencia, aclarándolo debidamente en las hipótesis generales del sistema.

A continuación, un ejemplo que puede usar para lograr esto:

En la primera colisión, crear un subdirectorio /dup para depositar los duplicados, y mover allí el archivo, pero con el nombre modificado: modificar el nombre del archivo agregándole una secuencia numérica conservada, es decir, una secuencia que no se destruye al cerrar la sesión, sino que se conserva durante toda la existencia del sistema

De esta forma, todos los archivos con nombre repetidos se pueden conservar.

El comando DetectO también debe mover de un lugar a otro, el sistema debe mantener coherencia en la política de manejo de duplicados

### **Producto: Consultas y Reportes: ReportO**

Este comando Debe ser desarrollado en Perl

Prever los siguientes escenarios:

- El sistema nunca fue inicializado (terminar)
- El sistema está bien inicializado (ejecutar)
- Consultas
  - Comparativa: usa préstamos y PPI.mae, graba “comparado”
  - Divergencias en tanto %: usa “comparado”
  - Divergencias en tantos \$: usa “comparado”
- Opciones
  - -a (ayuda)
  - -g (grabar)
- Parámetros



- País (OBLIGATORIO) (código de país)
- Sistema (código del sistema)
- Rango de periodos (desde-hasta) se incluyen ambos extremos

### Requisitos:

- Se deben emplear estructuras Hash en la resolución (requisito indispensable)
- Debe permitir al usuario efectuar N consultas, cambiando los filtros opcionales
- Debe mostrar los resultados por pantalla indicando en el encabezado los datos de la consulta
- Siempre las salidas se graban en el directorio de reportes indicado por el usuario

### **Punto 1. RECOMENDACIÓN**

- A. Filtrar el maestro según los parámetros ingresados y calcular el monto restante del maestro
- El monto restante del maestro debe ser calculado como la suma de los montos PRES, IMPAGO, INDE, INNODE y deducir el monto debitado DEB
- B. Comparar el monto restante del maestro contra el monto restante del préstamos.país

El registro de prestamos.país contra el cual se efectúa la comparación es aquel donde:

Código Préstamo del maestro	= Código Préstamo de prestamos.país
Año contable del maestro	= Año contable de prestamos.país
Mes contable del maestro	= Mes contable de prestamos.país

Si no se halla registro en prestamos.país, no se hace la comparación

Si hay más de un registro contra el cual comparar, tomar el de mayor día contable en prestamos.país

SI AÚN ASI hay más de un registro contra el cual comparar, tomar el de fecha de grabación más reciente

- C. Determinar el valor de la “recomendación” que puede ser “RECAL” o “NO-RECAL”. La recomendación se determina de la siguiente manera
- Si estado contable del maestro = SMOR y estado contable prestamos.país distinto a SMOR, recomendación = RECAL
  - Si monto restante del maestro **es menor** al monto restante de prestamos.país, recomendación = RECAL

- D. Mostrar un listado de los registros con recomendación con título en cada columna

Si fue invocado con la opción -g, entonces además grabar el resultado en un archivo llamado “comparado.país” si el archivo existe, agregarle los registros.

Campo	Variable
Código de País	PAIS
Código de Sistema	SISID
Código Préstamo	PRESID
RECOMENDACIÓN (calculado en este proceso)	RECO
Estado contable maestro	M.ESTADO
Estado contable préstamo	P.ESTADO
Monto Restante Maestro (calculado en este proceso)	M.REST
Monto Restante préstamo	P.REST
Diferencia en \$ (Monto Restante Maestro- Monto Restante préstamo)	DIF
Año contable	ANIO
Mes contable	MES
Día contable maestro	M.DIA
Día contable préstamo	P.DIA





E. Preguntar si se desea hacer otra comparativa

**Punto 2. LISTAR DIVERGENCIAS EN TANTO %**

- A. Para esta consulta se debe solicitar ADEMÁS que se ingrese un porcentaje de referencia X  
B. Esta consulta responde a la pregunta: En qué casos la diferencia (en valor absoluto) entre el monto restante del maestro y el monto restante del país es mayor al X %.  
C. Lee el archivo “comparado” correspondiente, y si la diferencia en porcentaje (en valor absoluto) es MAYOR al porcentaje de referencia, se lista.

Ejemplo: X= 7%

Renglón	maestro	préstamo	diferencia en \$	diferencia en %	acción
1	1000	700	300	30%	se lista
2	1000	750	250	25%	se lista
3	1000	800	200	20%	se lista
4	1000	850	150	15%	se lista
5	1000	900	100	10%	se lista
6	1000	950	50	5%	no se lista
7	1000	1000	0	0%	no se lista
8	1000	1050	-50	-5%	no se lista
9	1000	1100	-100	-10%	se lista
10	1000	1150	-150	-15%	se lista
11	1000	1200	-200	-20%	se lista
12	1000	1250	-250	-25%	se lista
13	1000	1300	-300	-30%	se lista

Columnas
Código de País
Código de Sistema
Código Préstamo
RECOMENDACIÓN
Monto Restante Maestro
Monto Restante préstamo
Diferencia en \$ (en valor absoluto)
Diferencia en % (en valor absoluto)

D. Cuando se ingresa la opción – g graba la salida en un archivo que debe tener un número de secuencia para asegurar que no se sobrescribe ningún informe previo.

E. Preguntar si se desea hacer otra consulta

**Punto 3. LISTAR DIVERGENCIAS EN TANTOS \$**

- A. Idem anterior pero aquí X es un valor (un importe) y no un porcentaje.  
B. Esta consulta responde a la pregunta: En qué casos la diferencia (en valor absoluto) entre el monto restante del maestro y el monto restante del país es mayor a X \$





## Archivos y Estructuras

### Punto 1. ARCHIVO MAESTRO DE PAÍSES Y SISTEMAS “P-S.MAE”

Campo	Tipo	Variable
Código de País	carácter	PAIS_ID
Nombre del País	caracteres	PAIS_DESC
Código de Sistema	numérico	SIS_ID
Nombre del Sistema	caracteres	SIS_DESC

Separador de campos: - guion

Ejemplo DATOS P-S.mae			
País	Nombre del País	Sistema	Nombre del Sistema
A	Argentina	6	Oracle Financial

### Punto 2. TABLA DE SEPARADORES: T1.TAB

Campo	Tipo	Variable
Código de País	carácter	PAIS_ID
Código de Sistema	numérico	SIS_ID
Carácter Separador de Campo	carácter	SEP_CAMP
Carácter Separador de decimales	carácter	SEP_DEC

Separador de campos: - guion

Ejemplo DATOS T1.tab			
País	Sistema	Separador de Campo	Separador Decimal
A	6	;	,

### Punto 3. TABLA DE CAMPOS: T2.TAB

Campo	Tipo	Variable
Código de País	carácter	PAIS_ID
Código de Sistema	numérico	SIS_ID
Nombre del Campo	caracteres	NOM_CAMP
Orden de aparición del campo	numérico	ORD_CAMP
Formato del campo	caracteres	FMT_CAMP

Separador de campos: - guion

Ejemplo DATOS T2.tab				
País	Sistema	Nombre del Campo	Nro. de Orden	Formato
A	6	MT_IMPAGO	1	commmax16.2
A	6	MT_PRES	2	commmax16.2
A	6	MT_DEB	3	commmax16.2
A	6	MT_INDE	4	commmax16.2
A	6	PRES_CLI	5	\$44.
A	6	CTB_ESTADO	6	\$4.
A	6	PRES_ID	7	\$12.
A	6	PRES_CLI_ID	8	\$6.
A	6	MT_INNODE	9	commmax16.2
A	6	CTB_FE	10	ddmmyy10.



**Punto 4. ARCHIVO MAESTRO CONTABLE DE PRÉSTAMOS PERSONALES IMPAGOS: PPI.MAE**

Campo	Tipo	Variable
Código de País	carácter	PAIS_ID
Código de Sistema	numérico	SIS_ID
Año contable	numérico	CTB_ANIO
Mes contable	numérico	CTB_MES
Día contable	numérico	CTB_DIA
Estado contable	caracteres	CTB_ESTADO
Fecha Préstamo	Fecha	PRES_FE
Código Préstamo	caracteres	PRES_ID
Tipo Préstamo	caracteres	PRES_TI
Monto Préstamo	numérico	MT_PRES
Monto Impago	numérico	MT_IMPAGO
Monto Interés Devengado	numérico	MT_INDE
Monto Interés No Devengado	numérico	MT_INNODE
Monto Debitado	numérico	MT_DEB

Separador de campos: ; punto y coma

Ejemplo DATOS PPI.mae													
PAIS_ID	SIS_ID	CTB_ANIO	CTB_MES	CTB_DIA	CTB_ESTADO	PRES_FE	PRES_ID	PRES_TI	MT_PRES	MT_IMPAGO	MT_INDE	MT_INNODE	MT_DEB
A	6	2017	10	1	MOR1	12/12/2009	RETA1366267	ALOA	11221,36	93,63	138,54	415,61	0
A	6	2017	10	1	MOR1	11/11/2011	RETA1366268	ALOA	11775,50	15,51	138,54	415,61	0
A	6	2017	10	1	MOR2	10/10/2010	RETA1373910	ALOA	10551,99	233,52	124,14	372,43	0
A	6	2017	10	1	SMOR	09/09/2009	RETA1384934	ACB	0	6,06	0	0	0
A	6	2017	10	1	SMOR	08/08/2008	RETA1384932	ACB	0	181,35	0	0	0
A	6	2017	11	1	MOR1	11/11/2011	RETA1369516	ALOA	9929,48	37,67	116,82	350,46	197,34
A	6	2017	12	1	SMOR	07/07/2007	RETA1385029	ALOA	0	191,43	0	0,01	0
A	6	2017	12	1	SMOR	06/06/2006	RETA1385022	ACB	0	0,07	0	0,01	0

**Punto 5. ARCHIVOS DE PRÉSTAMOS PERSONALES (NOVEDADES)**

Sin estructura definida

**Punto 6. ARCHIVOS DE PRÉSTAMOS PERSONALES POR PAÍS: PRESTAMOS.<PAÍS>**

Campo	Tipo	Variable	Fuente
Código de Sistema	numérico	SIS_ID	Del nombre del archivo de input
Año contable	numérico	CTB_ANIO	Interpretar la fecha CTB_FE y determinar año a grabar Input.CTB_FE[anio]
Mes contable	numérico	CTB_MES	Interpretar la fecha CTB_FE y determinar mes a grabar Input.CTB_FE[mes]
Día contable	numérico	CTB_DIA	Interpretar la fecha CTB_FE y determinar día a grabar Input.CTB_FE[dia]
Estado contable	caracteres	CTB_ESTADO	Input.CTB_ESTADO
Código Préstamo	caracteres	PRES_ID	Input.PRES_ID
Monto Préstamo	numérico	MT_PRES	Input.MT_PRES



Monto Impago	numérico	MT_IMPAGO	Input.MT_IMPAGO
Monto Intereses Devengados	numérico	MT_INDE	Input.MT_INDE
Monto Intereses No Devengados	numérico	MT_INNODE	Input.MT_INNODE
Monto Debitado	numérico	MT_DEB	Input.MT_DEB
Monto Restante	numérico	MT_REST	$MT\_REST = MT\_PRES + MT\_IMPAGO + MT\_INDE + MT\_INNODE - MT\_DEB$
Código del Cliente	caracteres	PRES_CLI_ID	Input.PRES_CLI_ID
Nombre del Cliente	caracteres	PRES_CLI	Input.PRES_CLI
Fecha de Grabación	Fecha	No Aplica	Fecha corriente en el formato estándar dd/mm/aaaa
Usuario de Grabación	caracteres	No Aplica	Usuario corriente

Separador de campos: ; punto y coma

<b>Ejemplo: algunos prestamos de argentina</b>															
SIS_ID	CTB_ANIO	CTB_MES	CTB_DIA	CTB_ESTADO	PRES_ID	MT_PRES	MT_IMPAGO	MT_INDE	MT_INNODE	MT_DEB	MT_REST	PRES_CLI_ID	PRES_CLI	INS_FE	INS_USER
6	2017	10	1	MOR1	RETA1366267	11221,36	93,63	138,5	415,6	0	11869,14	783920	ARGENTINA POINT CLAIM MANAGEMENT SRL	09/04/2017	Prueba01
6	2017	10	2	MOR1	RETA1366268	11775,5	15,51	138,5	415,6	0	12345,16	783920	ARGENTINA POINT CLAIM MANAGEMENT SRL	09/04/2017	Prueba01
6	2017	10	3	MOR2	RETA1373910	10551,99	233,5	124,1	372,4	0	11282,08	722973	ARGENTINA COACHWORKS	09/04/2017	Prueba01
6	2017	10	4	MOR1	RETA1384934	0	6,06	0	0	0	6,06	741508	ARGENTINA CARS SA	09/04/2017	Prueba01
6	2017	10	5	MOR1	RETA1384932	0	181,4	0	0	0	181,35	741508	ARGENTINA CARS SA	09/04/2017	Prueba01
6	2017	11	6	MOR1	RETA1369516	9929,48	37,67	116,8	350,5	197,3	10237,09	722973	ARGENTINA DYNE MOTORS SA	09/04/2017	Prueba01
6	2017	12	7	MOR1	RETA1385029	0,01	191,4	0	0,01	0	191,45	541165	AUTO ARGENTINA LANGUAGE SA	09/04/2017	Prueba01
6	2017	12	8	MOR1	RETA1385022	0,01	0,07	0	0,01	0	0,09	541165	AUTO ARGENTINA LANGUAGE SA	09/04/2017	Prueba01

**FIN**