HOJA CARATULA

Primera hoja de la carpeta, complete con grupo e integrantes

Tema	**Grupo**	**Ayudante**	**Correcciones**			
			Fecha	**Hora Inicio**	**Hora Fin**	**Resultado**
Parte I		**Sandra o Guido**	**Entrega parte I 27/09**			
			Entrega parte II 18/10			

INTEGRANTES

	Padrón	**Apellido y Nombre**	**Asistencia a Entrega I**	**Asistencia a Entrega II**	**Evaluación Individual Final**
1					
2					
3					
4					
5					
6					

Incorpore a continuación el índice de la carpeta y luego todos los ítems solicitados en la documentación con hojas numeradas en el margen derecho y numero de grupo en el margen izquierdo

Incorpore como apéndice al final de la carpeta el enunciado Primera Parte del TP

Sujete las hojas de manera tal que se pueda hojear la carpeta sin que las hojas se caigan ni mezclen (no dentro de folios que impiden su lectura, carpeta con ganchos o broches tipo nepaco es lo adecuado) Las hojas sueltas no se considerarán como parte de la entrega.

El día de la entrega debe traer: carpeta, planilla de evaluación impresa para entregar al docente, maquina con SO instalado adecuado para ejecutar el TP, pen drive con paquete de instalación y datos entregados por la cátedra.

27 de Septiembre o 29 de Septiembre según el turno asignado	*Vencimiento Primera Parte del TP – Asistencia obligatoria de los responsables de la resolución de la primera parte del TP. Asistencia opcional del resto del grupo.*
18 de Octubre o 20 de Octubre según el turno asignado	*Vencimiento Segunda y ultima Parte del TP – Asistencia obligatoria de los responsables de la resolución. Asistencia opcional del resto del grupo.*

Se evaluará el trabajo grupal y a cada integrante en forma individual. El objetivo de esto es comprender la dinámica de trabajo del equipo y los roles que ha desempeñado cada integrante del grupo.

Para que el alumno apruebe el trabajo práctico debe estar aprobado en los dos aspectos: grupal e individual.

Dentro de los ítems a chequear el ayudante evaluará aspectos formales (como ser la forma de presentación de la carpeta), aspectos funcionales: que se resuelva el problema planteado y aspectos operativos: que el TP funcione integrado.

Entrega final, una vez evaluado satisfactoriamente el TP, se debe remitir el paquete de instalación vía correo electrónico a so7508@gmail.com. En el asunto del correo indicar Nro. de Grupo y Ayudante asignado. En el cuerpo del correo se debe indicar la versión de Sistema Operativo usada para la ejecución.

Facultad de Ingeniería - UBA

Sistemas Operativos 75.08

Trabajo Práctico: Primera Entrega

EPLAM

Integrantes:	*Mails:*
Agustina Barbetta *96528*	agustina.barbetta@gmail.com
Ana Czarnitzki *96812*	czarniana@gmail.com
Francisco Ordoñez *96478*	ordonez.f93@gmail.com
Manuel Porto *96587*	manu.porto94@gmail.com
Santiago Aguilera *95795*	marquito.santi@gmail.com

Grupo 6

30 de noviembre de 2016

Índice

1. Aclaraciones Globales e Hipótesis por comando

Para el correcto funcionamiento del aplicativo se deben cumplir los siguientes requisitos:

1. Bash v4 o superior
2. Perl v5.10.0 o superior
3. Sistema Operativo Linux

2. Problemas relevantes

Como primer y principal problema podemos destacar la dificultad de escribir código bash común para los diferentes sistemas operativos. En nuestro caso programamos tanto en Mac OS como Linux y dado que no pudimos encontrar un método que funcione en todas las computadoras (en particular para el comando du -b, que pudimos utilizar en la segunda y no así en la primera distribución) decidimos utilizar la versión para Linux ya que este es el sistema operativo con el que generalmente estuvimos trabajando en materias anteriores.
En el caso de Windows, se pudo correr el `EPLAM` satisfactoriamente, pero solo para la distribución `Windows 10` y haciendo uso del *Windows Subsystem for Linux*.
Otro problema que tuvimos, al momento de realizar la primera entrega de este trabajo, fue el hecho de que el mismo no tiene todos los comandos programados. De esta manera, hay ciertas funciones que nos hubieran sido más sencillas de probar si contáramos con una solución global. Específicamente, en el `Installep` nos hubiera servido tener hecho el `Movep`.
Al momento de realizar la segunda entrega subsanamos este primer problema al ya tener todo el sistema como un conjunto, facilitándo la integración entre las distintas partes.

3. README

Adjunto por separado

4. Listado de Nuevas Funciones y/o Comandos Auxiliares

4.1. Installep

El comando `Installep.sh` utiliza varias funciones, cada una representa uno de los puntos del enunciado, de modo que las mismas son:

- input_directories.
- system_already_installed.
- set_news_size.
- show_values.
- instalation_confirm.
- installation.
- create_conf_archive.
- end_process.

Cada una de ellas se encuentra documentada en el código fuente.
A eso se le suma un main, el `Installep` propiamente dicho, que ejecuta a cada una de las anteriores de forma secuencial, pidiendo las validaciones correspondientes (que devuelvan 0 o 1 para el caso de system_already_installed y aquellas que esperan confirmaciones por parte del usuario) y volviendo atrás en el caso de ser necesario.
Existen algunas funciones más pero que podrían considerarse auxiliares llamadas `input_directory` y `directory_already_exists`. La primera espera del usuario un nombre para un directorio y sobre-escribe

una variable pasado por parámetro en el caso de que la cadena escrita sea distinta de `config` dado que el mismo es un valor reservado. Por otro lado, la ultima mencionada verifica que el directorio propuesto por el usuario no exista previamente.

4.2. Initep

El comando `Initep.sh` se implementa con una función auxiliar por paso requerido en el enunciado del trabajo. Cada función realiza una única tarea, devolviendo 0 en caso de éxito o 1 en caso contrario. Al final del *script*, se invoca a cada función y se evalúa su retorno. En caso de falla, se destruye el ambiente creado (`unset` de cada variable exportada) y se retorna un código de salida específico.
Para verificar una inicialización previa de las variables de entorno, se crea una variable adicional `ENV`, a la cual se le asigna el valor 1 al momento de la inicialización. De esta forma, en cada ejecución del comando, se verifica si `ENV` es equivalente a 1. En este caso, se detiene `Initep` notificando al usuario que las variables ya fueron inicializadas y exportadas.
Las funciones auxiliares se encuentran documentadas en el código fuente.

4.3. Demonep

El daemon es el encargado de estar corriendo siempre en segundo plano (para ello se lo necesita ejecutar agregando al final un `&`). Sus tareas consisten en:

- Verificar si hay nuevos archivos dentro del directorio de novedades `$DIRREC`
- Validar cada archivo si es apto para ser procesado, moviendolo de directorios de acuerdo al estado de su validacion
- Invocar a la función `Procep.sh` si hay archivos validos a ser procesados.

El *script* corre en ciclos con un tiempo de espera de 15 segundos. Esto quiere decir que el mismo cada 15 segundos realizara "pasadas" a la carpeta de novedades y validara los archivos internos.
La forma de validación es la siguiente:

1. Ver que el archivo comience con `ejecutado_` y tenga la extensión correcta `.csv`.
2. Ver que el archivo tenga el año presupuestario corriente. En este caso será 2016. El mismo se adapta al año corriente (no es un numero mágico)
3. Ver que el archivo tenga un código de provincia valido. Los códigos se reciben del archivo `$DIRMAE`/provincias.csv. Si el código de la provincia no se encuentra dentro del csv, no sera correcto
4. Ver que la fecha sea mayor a la del año presupuestario pero menor a la actual. Para ello se validara de la siguiente forma:
   - El año debe ser el corriente
   - Si es el mismo mes, el día no debe ser mayor al actual
   - Si es otro mes, el mes no debe ser mayor al actual
   - Validar que la fecha este bien formada y con el formato pedido (YYYYmmdd)

Las validaciones se hacen en el orden descrito. Si alguna de ellas falla, la única que se reportará en el *log* será la primera ocurrencia.
Un ejemplo de archivo correcto seria:
`ejecutuado_2016_4_20160809.csv`
Si el archivo es correcto, el mismo se moverá al directorio `$DIROK`, caso contrario se encontrara en `$DIRNOK`. Para ello se hará uso del *script* `Movep.sh`.
Finalmente, siempre y cuando haya archivos dentro de `$DIROK`, si el script `Procep.sh` no se encuentra en ejecucion, se ejecutara el mismo para que procese los archivos validos. Si ya se encuentra en ejecucion, se postergara para el proximo ciclo del daemon.
Cabe notar que el daemon guardará en un *log* todos los movimientos de archivos que realice y acciones (incluyendo información especifica de que fue lo que fallo en caso de haber errores). Para eso hará uso del *script* `Logep.sh`

4.4. Logep

El comando `Logep` escribe la informacion indicada en distintos archivos de log, dependiendo del comando que este logeando la informacion. El ejecutable provee sus instrucciones de uso:

```
Uso: Logep.sh -c comando -m 'Mensaje' -t tipo de mensaje
Ejemplo: Logep.sh -c movep -m 'Se movio archivo foo' -t INF
  -h                  Muestra este mensaje de ayuda.
  -c comando          Escribir el mensaje en comando.log.
  -m 'Mensaje'        Mensaje a escribir.
  -t tipo de mensaje  Especifica el tipo de mensaje. Puede ser INF (tipo default),
      WAR, ERR. (Parametro opcional)
```

Si bien para la implementación del comando, `Logep.sh` no se programaron comandos auxiliares, si se utilizaron comandos provistos por el sistema operativo. Estos son:

1. `du` para obtener el espacio disponible en disco.
2. `tail` para obtener las ultimas n lineas y conservarlas en el archivo de log cuando el mismo es reducido.
3. `getopts` para poder parsear los argumentos por linea de comandos.

Además de esto se implementaron funciones con el objetivo de encapsular código repetido y simplificar el código fuente del comando.

1. `show_help`
2. `max_log_size_reached`
3. `reduce_log_file`
4. `write_log_file`

4.5. Movep

Movep es el encargado de mover archivos de un directorio a otro. El mismo puede recibir por parametros a la hora de ser invocado los siguientes argumentos

- c: Comando que lo invoco
- o: Origen
- d: Destino

`Movep` se encargara de mover el archivo del origen al destino, y loggear el resultado de la operación. Si se provee de un comando a la hora de invocarlo, el mismo loggeara bajo el nombre de ese comando. Caso contrario se creara bajo el *log* de `Movep`.

El script valida ciertos casos de uso que no pueden darse. Los mismos son:

1. Origen no existente (o vacío)
2. Destino no existente (o vacío)
3. Origen y destino iguales

Si ocurre alguno de los mismos, el proceso se detendrá y se creara una entrada en el *log* del error ocurrido.

A su vez, si el archivo ya se encuentra en el directorio destino, el mismo se creara bajo un subdirectorio `dpl`, usando un sufijo de incremento para poder diferenciarlos. Un ejemplo sería:

```
# El archivo.csv se mueve correctamente a carpeta/
> bash Movep.sh -o archivo.csv -d carpeta/
# El archivo.csv ya existe en carpeta/, se crea dpl/ y se guarda archivo.csv alli
> bash Movep.sh -o archivo.csv -d carpeta/
# El archivo.csv ya existe en carpeta/ y en dpl/, se guarda dentro de
dpl archivo.1.csv
> bash Movep.sh -o archivo.csv -d carpeta/
# El archivo.csv se mueve correctamente a carpeta/
> bash Movep.sh -o archivo.csv -d carpeta/
# El archivo.csv ya existe en carpeta/, se crea dpl/ y se guarda archivo.csv alli
> bash Movep.sh -o archivo.csv -d carpeta/
# El archivo.csv ya existe en carpeta/ y en dpl/, se guarda dentro de dpl archivo.1.
    csv
> bash Movep.sh -o archivo.csv -d carpeta/
```

4.6. Procep

El propósito de este proceso es reunir en un solo archivo todos los registros de presupuesto ejecutado validos

- Es el tercero en orden de ejecución y es invocado por Demonep.
- Graba los registros validos en un archivo de salida.
- Graba los registros rechazados en un archivo de rechazos indicando su motivo.

Luego de verificar que los archivos procesados por `Demonep` tengan la estructura correcta, `Procep` carga los archivos maestros en distintas tablas de hash para su fácil acceso en el momento de la verificación, luego procesa cada registro de cada archivo verificando que sus campos sean correctos.

Cada campo tiene una función que lo verifica, si el valor del campo `no` se encuentra en su tabla de hash se concatena el error a una variable que se usa en todas las verificaciones. Al finalizar el checkeo, si esa variable está vacia, el registro es correcto y se lo graba en el archivo de aceptados, caso contrario, al archivo de rechazados.

Al finalizar todos los registros de un archivo, se escribe en el log cuántos se validaron, cuántos son correctos y cuántos erróneos.

Las descripciones de los errores se encuentran en una tabla de hash.

4.7. Listep

El propósito de este proceso es generar diferentes listados de control entre el presupuesto sancionado y el presupuesto ejecutado. Tambien permite generar listados de control en base a distintos archivos de cotizaciones y años presupuestarios.

- Es el cuarto en orden de ejecución y se dispara manualmente.
- Es el unico proceso que no graba en el archivo de log.

El mismo dependiendo de la opción seleccionada permite aplicarle filtros para el orden en el que las entries serán mostradas, o filtrar por distintos campos a procesar (dependiendo la opción seleccionada se puede filtrar por trimestres o por centros, entre otros).

El mismo permite una opción para poder guardar la salida en algún archivo. El formato a ser guardado sera en csv, con ";"de separador. Siempre va a estar también entregando la salida por `stdout`.

5. Listado de Nuevos Archivos

El programa `Listep` crea diferentes archivos no temporales dependiendo de los flags con los que se lo ejecute.

1. `perl Listep.pl --sanc [sanc_file]` Crea un archivo `sancionado-yyyy.csv`.
2. `perl Listep.pl --ejec [ejec_file]` Crea un archivo sobre el presupuesto ejecutado (`ejecutado-yyyy.csv`).

3. `perl Listep.pl --ctrl [ejec_file] [sanc_file]` Crea un reporte sobre el presupuesto en su conjunto.

En todos los casos los archivos son creados en el path donde fue ejecutado `Listep` y no son utlizados por otros comandos.

Por su parte, el comando `Procep` crea dos archivos distintos, `ejecutado-yyyy.csv` donde se almacenan todos los registros validos y `rechazado-yyyy.csv` que contiene registros rechazados. El primer archivo es utilizado por `Listep`, mientras que el segundo no es utilizado por ningún comando. Ambos archivos son guardados en `$DIRPROC`.

6. Listado de DATOS

Los archivos de datos utilizados por el aplicativo son los siguientes:
Archivos maestros:

actividades.csv con campos COD_ACT, OBJ_ACT, PGM_ACT, NOM_ACT y 33 entradas.

centros.csv con campos COD_CEN, NOM_CEN y 17 entradas.

provincias.csv con campos COD_PROV, NOM_PROV, REG_PROV y 24 entradas.

sancionado-2016.csv con campos COD_CEN_SAN, NOM_TRI_SAN, F11_FIN_SAN, F21_FIN_SAN y 68 entradas.

sancionado-2015.csv con campos COD_CEN_SAN, NOM_TRI_SAN, F11_FIN_SAN, F21_FIN_SAN y 68 entradas.

tabla-AxC.csv con campos ACT_AXC, CEN_AXC y 157 entradas.

trimestres.csv con campos ANO_TRI, NOM_TRI, FDESDE_TRI, FHASTA_TRI y 10 entradas.

Archivos novedades:

ejecutado_2016_*.csv con campos ID_EJE, FECHA_EJE, COD_CEN_EJE, NOM_ACT_EJE, NOM_TRI_EJE, GASTO_EJE y distintas cantidades de entradas.

7. Apéndice

7.1. Enunciado: Primera parte

Apéndice I

Enunciado primera parte – Tema M

Índice del enunciado del TP

Introducción Narrativa – ejecución planificación presupuestaria

Al sistema lo llamaremos EPLAM y el mismo estará compuesto por:

- La documentación del sistema EPLAM
- Un comando Shell Instalep para la instalación de sistema
- Un comando Shell Initep para la configuración del entorno de ejecución
- Un comando shell Demonep para la recepción de los archivos de novedades
- Un comando Shell Procep para determinación de imputación de gastos a cada partida presupuestaria
- Un comando PERL Listep para la generación de listados y reportes de gestión
- Una Función (en Shell o en Perl) denominada Logep que se emplea para grabar los archivos de log
- Una Función (en Shell o en Perl) denominada Movep que se emplea para mover archivos

Se requiere que estos comandos trabajen en forma integrada, no deben ser comandos independientes ya que la naturaleza del TP es que desarrollen UN SISTEMA.

Documentación Solicitada

1. Contenido de la Carpeta – Primera Entrega

1.1. Carátula (primer hoja de este documento)
1.2. Nuevo Índice del Contenido de la Carpeta.

OBLIGATORIO contar con un índice con número de página en cada ítem, el número de página puede ser incorporado manualmente.

1.3. Aclaraciones Globales e Hipótesis por comando

Documente cualquier aclaración que considere necesaria para asegurar el éxito de la corrección

Documente las hipótesis que ha considerado para la resolución en cada comando (opcional)

Si no hay aclaraciones ni hipótesis, igualmente incorpore esta hoja con solo su titulo

1.4. Problemas relevantes

Describa los problemas relevantes que se hayan presentado durante el desarrollo, la integración y/o la prueba del sistema. Explique cómo fueron solucionados.

1.5. Archivo README

OBLIGATORIO que en la carpeta se incluya la impresión del README, el docente seguirá estos pasos para instalar y ejecutar el TP

1.6. Listado de Nuevas Funciones y/o Comandos Auxiliares

Si agrega funciones o comandos auxiliares, Indique: Nombre de la función, quienes la usan, para que la usan.

1.7. Listado de Nuevos Archivos

Si agrega archivos NO TEMPORALES, Indique: Nombre del archivo, donde lo almacenan, quienes lo usan, para que lo usan.

1.8. Listado de DATOS

Imprima un listado con todos los nombres de los archivos de datos entregados por la cátedra y su cantidad de registros

1.9. Apéndice, enunciado del TP

2. Contenido de la Carpeta – Segunda Entrega

2.1. Carátula (heredada de la primera entrega)
2.2. Índice de la Segunda Parte
2.3. Aclaraciones Globales e Hipótesis por comando
2.4. Problemas relevantes
2.5. Archivo README
2.6. Listado de Nuevas Funciones y/o Comandos Auxiliares
2.7. Listado de Nuevos Archivos
2.8. Casos de Prueba

Imprima un listado con todos los casos de prueba: nombre del archivo y que lo caracteriza (que es lo que se prueba con ese archivo)

2.9. Apéndice A, todo el Contenido de la Carpeta – Primera Entrega
2.10. Apéndice 2, el enunciado del TP segunda entrega

Especificación de comandos y Funciones

Recomendaciones para el equipo de desarrollo

1. Se deberá tener en cuenta para la resolución TODAS las condiciones que se enuncian.

Se deben respetar los formatos de archivos especificados y la estructura de directorios planteada

Los pasos de ejecución sugeridos son solo a los efectos de ordenar la explicación, por lo cual deben considerarse meramente indicativos.

Si el equipo de desarrollo lo considera pertinente, puede modificarlos tanto sea en el orden de ejecución como en la forma de resolverlo, siempre y cuando esto no afecte el resultado final esperado. También pueden:

- Crear nuevos scripts
- Aumentar la funcionalidad de los comandos solicitados

Estos cambios deben estar documentados en la carpeta que entrega el día de vencimiento del tp.

2. Archivos Auxiliares

Se debe evitar el uso de archivos auxiliares permanentes, los archivos auxiliares temporales, se deben eliminar ANTES de finalizar la ejecución del comando.

3. Movimiento de Archivos

En líneas generales no se borra ningún archivo de datos, se los mueve de un lugar a otro para asegurar la integridad de la información original. Se solicita una función de librería Movep para el movimiento de archivos

4. Manejo de errores, logueo

Toda invocación desde un comando a otro debe devolver un código de retorno cero (0) si fue exitoso o distinto de cero si tuvo errores.

Todo evento que genera algún tipo de error debe ser grabado en el log y mostrado por pantalla

EVITE retrasar el proceso de evaluación/corrección del TP debido a la falta de rastreo de eventos. Es por ello que se recomienda dejar en los scripts las pistas de rastreo que crea convenientes condicionadas a un flag que se enciende si es necesario. Esto evitara que en la ejecución estándar se inunde de mensajes sin interés para el usuario.

La escritura de los archivos de log de los comandos debe ser homogénea, es por ello que se centraliza en la función Logep

5. Archivo de Configuración

La instalación deja un archivo de configuración con varios registros con el contenido de variables usadas en el sistema. Los desarrolladores pueden agregar más registros si lo consideran necesario.

Indicaciones para el equipo de integración y testing

Siempre realizar la integración antes de cualquier entrega, para poder subsanar los errores de comunicación que surjan entre los comandos encadenados.

La cátedra provee los archivos de prueba

- El primer registro de cada archivo contiene los nombres de cada campo, pueden remover este registro y dejar solo los datos o bien contemplar el no uso del primer registro.

Indique en el punto hipótesis y aclaraciones globales de la carpeta que política emplea en todo el sistema, la misma política debe aplicarse a todos los comandos.

- Si los archivos remitidos por la cátedra contienen caracteres incompatibles con su configuración (por ejemplo, el carácter de fin de registro o fin de archivo), puede realizar la conversión que necesite siempre que sea homogénea (igual para todos los archivos) y sin modificar los datos

Si lo hace, indíquelo en el punto hipótesis y aclaraciones globales de la carpeta

Se proveen archivos de prueba (novedades) con un alto porcentaje de información libre de error, es responsabilidad del equipo de testing generar otros archivos de prueba con casos lo suficientemente heterogéneos como para contemplar todas las variantes de ejecución, en particular las de rechazo o error

Comandos Solicitados

1. Contenido de la Primera Entrega

1.1. La documentación del sistema EPLAM, primera entrega
1.2. Un comando Shell Instalep para la instalación de sistema
1.3. Un comando Shell Initep para la configuración del entorno de ejecución
1.4. Un comando shell Demonep que pueda ser invocado por Initep, duerma un tiempo X (largo) y vuelva a empezar. Eventualmente pueden hacer mas desarrollo dentro de este script, pero no es necesario que este completo para la primera entrega.
1.5. Una Función (en Shell o en Perl) denominada Logep que se emplea para grabar los archivos de log

Se requiere que funcione el instalador, el configurador de entorno grabando en el log con la función Logep y llamando al comando Demonep

2. Contenido de la Segunda Entrega

2.1. La documentación del sistema EPLAM, segunda entrega
2.2. Todo el TP completo y funcionando

Instalación: Instalep

Descripción

El propósito de este comando es efectuar la instalación del sistema EPLAM.

Punto 1. PAQUETE

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 instalable deberá contener:

- La documentación, los scripts desarrollados y los archivos con datos (maestros y novedades) entregados por la cátedra

La extracción del paquete debe generar en $GRUPO un subdirectorio denominado /dirconf

Resguarde siempre el paquete original incluidos los datos de prueba en algún lugar de $grupo

Brinde indicaciones precisas en el README para realizar la descarga del paquete

- Una explicación de cómo descargar el paquete
- Una explicación de cómo descomprimir, crear directorio del grupo, etc
- Una explicación de lo que sea crea a partir de la descompresión

Punto 2. INSTALADOR

Brinde indicaciones precisas en el README para realizar la instalación del paquete

- Una explicación sobre que se requiere para poder instalar y/o ejecutar el sistema
- Instrucciones de instalación del sistema EPLAM
- 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 que considere necesaria

Punto 3. DIRECTORIO DE TRABAJO

Para realizar la instalación el directorio de trabajo debe ser Grupoxx, donde xx es su número de grupo

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

Toda la instalación debe depender del directorio de trabajo $GRUPO (creado en la descompresión del paquete)

Ningún comando puede leer o grabar información del sistema fuera de esta rama

Punto 4. INTERACCIÓN CON EL USUARIO

Se Interactúa con el usuario para solicitarle que defina subdirectorios, espacio requerido, etc.

Siempre que se interactúa con el usuario proponer el valor por default mostrándolo entre paréntesis. Ejemplo:

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

Si durante el mismo hilo (o instancia) de ejecución de Instalep se debe reiniciar desde un paso previo (volver a un punto anterior), los valores default propuestos inicialmente por el script deben ser reemplazados por los valores recientemente ingresados por el usuario, es decir, que el script debe tener "memoria" de los valores definidos por el usuario durante el mismo hilo de ejecución.

Punto 5. PREVENIR USO DE DIRCONF

El directorio $GRUPO/dirconf (ya fue 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.

Si Uds necesitan ampliar la lista de nombres reservados aclárenlo en las hipótesis y programen la validación en este script.

Punto 6. LOG

El log de la instalación Instalep.log debe grabarse en el directorio de configuración $GRUPO/dirconf

Grabar en el log todos los mensajes que se muestran por pantalla, los valores ingresados por el usuario. Si se ejecuta muchas veces el script, se agregan las nuevas líneas.

Grabar en el log el siguiente mensaje

```
Inicio del proceso. Usuario Fecha y Hora
```

Punto 7. DETECTAR SISTEMA YA INSTALADO

Detectar si el paquete EPLAM ya está instalado
Este script de instalación se debe preparar de manera tal que detecte si el paquete ya se encuentra instalado

- Si Instalep.conf no existe, asumir que el paquete no fue instalado. Ir a definir los nombres de los directorios
- Si Instalep.conf existe, asumir que el paquete ya fue instalado. Mostrar y grabar en el log el siguiente mensaje

```
Directorio de Configuración: $GRUPO/dirconf (mostrar path y listar archivos)
Directorio de Ejecutables: ($GRUPO/DIRBIN mostrar path y listar archivos)
Directorio de Maestros y Tablas: ($GRUPO/DIRMAE mostrar path y listar archivos)
Directorio de Recepción de Novedades: ($GRUPO/DIRREC mostrar path)
Directorio de Archivos Aceptados: ($GRUPO/DIROK mostrar path)
Directorio de Archivos Procesados: ($GRUPO/DIRPROC mostrar path)
Directorio de Archivos de Reportes: ($GRUPO/DIRINFO mostrar path)
Directorio de Archivos de Log: ($GRUPO/DIRLOG mostrar path)
Directorio de Archivos Rechazados: ($GRUPO/DIRNOK mostrar path)
```

Ir a FIN

Punto 8. DEFINIR LOS NOMBRES DE LOS DIRECTORIOS

1. **Ejecutables**
   Mostrar y grabar en el log el siguiente mensaje con su respuesta:

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

   Proponer /bin y si el usuario lo desea cambiar, permitírselo.

   El usuario puede ingresar un nombre simple como “bin” o un subdirectorio como /so7508/tp/binarios

   Reservar este path en la variable DIRBIN

2. **Maestros y tablas**
   Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Defina el directorio de Maestros y Tablas ($grupo/mae):
```

   Proponer /mae y si el usuario lo desea cambiar, permitírselo.

El usuario puede ingresar un nombre simple o un subdirectorio

Reservar este path en la variable DIRMAE

3. **Recepción de novedades**
   Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Defina el directorio de recepción de novedades ($grupo/nov):
```

Proponer /nov y si el usuario lo desea cambiar, permitírselo.

El usuario puede ingresar un nombre simple o un subdirectorio

Reservar este path en la variable DIRREC

4. **Aceptados por el demonio**
   Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Defina el directorio de Archivos Aceptados ($grupo/ok):
```

Proponer /ok y si el usuario lo desea cambiar, permitírselo.

El usuario puede ingresar un nombre simple o un subdirectorio

Reservar este path en la variable DIROK

5. **Procesados por el proceso de imputación de gastos**
   Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Defina el directorio de Archivos Procesados ($grupo/imp):
```

Proponer /imp y si el usuario lo desea cambiar, permitírselo.

El usuario puede ingresar un nombre simple como “imp” o un subdirectorio como /so7508/tp/presupuesto/imputacion

Reservar este path en la variable DIRPROC

6. **Reportes**
   Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Defina el directorio de Reportes ($grupo/rep):
```

Proponer /rep y si el usuario lo desea cambiar, permitírselo.

El usuario puede ingresar un nombre simple o un subdirectorio

Reservar este path en la variable DIRINFO

7. **Log**
   Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Defina el directorio de log ($grupo/log):
```

Proponer /log y si el usuario lo desea cambiar, permitírselo.

El usuario puede ingresar un nombre simple o un subdirectorio

Reservar este path en la variable DIRLOG

8. **Rechazados**
   Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Defina el directorio de rechazados ($grupo/nok):
```

Proponer /nok y si el usuario lo desea cambiar, permitírselo.

El usuario puede ingresar un nombre simple o un subdirectorio

Reservar este path en la variable DIRNOK

Punto 9. **VERIFICAR ESPACIO EN DISCO.**

Se debe solicitar al usuario que indique cual es el espacio mínimo libre requerido en el directorio para la recepción de novedades

Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Defina espacio mínimo libre para la recepción de archivos en Mbytes (100):
```

Proponer 100 Mb, Si el usuario lo desea cambiar, debe ingresar una cantidad que se interpreta como Mb. Reservar este valor en la variable datasize

Chequear si hay disponible el espacio requerido. Si no hay, mostrar y grabar en el log

```
Insuficiente espacio en disco.
Espacio disponible: xx Mb.
Espacio requerido $DATASIZE Mb
Inténtelo nuevamente.
```

Volver a verificar espacio en disco.

Si hay suficiente espacio, mostrar y grabar en el log

```
Suficiente espacio en disco.
Espacio disponible: xx Mb.
Espacio requerido $DATASIZE Mb
De enter para continuar.
```

Punto 10. MOSTRAR VALORES

Limpiar la pantalla y Mostrar los valores de los parámetros configurados

Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Directorio de Configuración: $GRUPO/dirconf (mostrar path y listar archivos)
Directorio de Ejecutables: ($GRUPO/DIRBIN mostrar path y listar archivos)
Directorio de Maestros y Tablas: ($GRUPO/DIRMAE mostrar path y listar archivos)
Directorio de Recepción de Novedades: ($GRUPO/DIRREC mostrar path)
Directorio de Archivos Aceptados: ($GRUPO/DIROK mostrar path)
Directorio de Archivos Procesados: ($GRUPO/DIRPROC mostrar path)
Directorio de Archivos de Reportes: ($GRUPO/DIRINFO mostrar path)
Directorio de Archivos de Log: ($GRUPO/DIRLOG mostrar path)
Directorio de Archivos Rechazados: ($GRUPO/DIRNOK mostrar path)
Estado de la instalación: LISTA
Desea continuar con la instalación? (Si – No) _
```

Si el usuario indica No, volver atrás

Si el usuario indica Si confirmar inicio de instalación

Punto 11. VOLVER ATRAS

Si el usuario indica No, Limpiar la pantalla y realizar nuevamente la Validación 6: Definir los nombres de los directorios pero esta vez, los valores default propuestos deben cambiarse por los recientemente ingresados por el instalador.

Por ejemplo si para los ejecutables el usuario indicó “mis_comandos_del_tp” entonces ahora el default no debe ser bin sino mis_comandos_del_tp porque es lo último que ingreso el usario para ese directorio

Punto 12. CONFIRMACIÓN DE LA INSTALACIÓN

Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Iniciando Instalación. Esta Ud. seguro? (Si-No)
```

Si el usuario indica Si, Continuar en el paso: “Instalación”

Si el usuario indica No, ir a **FIN**

Punto 13. INSTALACIÓN

1. crear las estructuras de directorio requeridas
Mostrar y grabar en el log el siguiente mensaje

```
Creando Estructuras de directorio. . . .
```

Como resultado de la instalación la estructura de $GRUPO debe ser la siguiente:

- $GRUPO/DIRBIN en donde se depositarán los scripts ejecutables
- $GRUPO/DIRMAE en donde depositarán los archivos maestros y tablas
- $GRUPO/DIRREC sin archivos, lo usaremos para simular la recepción de archivos de novedades
- $GRUPO/DIROK sin archivos, lo usaremos para depositar los archivos aceptados por el demonio
- $GRUPO/DIRPROC/proc sin archivos, lo usaremos para depositar los archivos procesados por el proceso de imputación de gastos
- $GRUPO/DIRINFO sin archivos, lo usaremos para depositar los Reportes que genera el comando Perl luego de la imputación de gastos
- $GRUPO/DIRLOG sin archivos, lo usaremos para depositar los Archivos de Log
- $GRUPO/DIRNOK para depositar los Archivos Rechazados

2. Mover los ejecutables y funciones
Mostrar y grabar en el log el siguiente mensaje

```
Instalando Programas y Funciones
```

3. Mover los archivos maestros y tablas
Mostrar y grabar en el log el siguiente mensaje

```
Instalando Archivos Maestros y Tablas
```

Punto 14. ARCHIVO DE CONFIGURACIÓN

Se debe almacenar la información de configuración del sistema en el archivo Instalep.conf en $GRUPO/dirconf grabando un registro para cada una de las variables indicadas durante este proceso.

Mostrar y grabar en el log el siguiente mensaje

```
Actualizando la configuración del sistema
Instalación CONCLUIDA.
```

Punto 15. FIN

Borrar archivos temporarios, si los hubiese generado

Grabar en el log el siguiente mensaje

```
Fin del proceso. Usuario Fecha y Hora
```

Cerrar el archivo Instalep.log y Terminar el proceso

Initep

Input

- Archivo de Configuración $GRUPO/dirconf/EPLAM.conf

Output

- Log del Comando $GRUPO/DIRLOG/Initep.log

Opciones y Parámetros

- A especificar por el desarrollador

Descripción

El propósito de este comando es preparar el ambiente del sistema y dejarlo listo para su ejecución.

- Es el primero en orden de ejecución
- Se dispara manualmente
- Graba en el archivo de Log a través del Logep
- Invoca, si corresponde, el siguiente proceso: Demonep

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

Continúa con la asignación de valor a un conjunto de variables de ambiente que van a ser usadas por el resto del sistema y luego ofrece arrancar automáticamente el comando Demonep

No se puede ejecutar ningún comando si la inicialización de ambiente no fue realizada, es por ello que los comandos JAMAS deben acceder al archivo de configuración para conocer directorios o el contenido de las variables del sistema, sino que deben usar las variables de ambiente que define este proceso.

Pasos Sugeridos

Punto 1. **VERIFICAR SI EL AMBIENTE YA HA SIDO INICIALIZADO**

Initep debe setear las variables de ambiente una sola vez por cada sesión de usuario.

Si se intenta ejecutar más de una vez en la misma sesión de usuario, no permitirlo y explicar la situación mostrando y grabando en el log el siguiente mensaje:

```
Ambiente ya inicializado, para reiniciar termine la sesión e ingrese nuevamente
```

y terminar la ejecución.

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

Setear la variable PATH y cualquier otra variable de ambiente que considere necesarias

GRUPO DIRBIN DIRMAE DIRREC DIROK DIRPROC DIRINFO DIRLOG

Si no se puede efectuar esta tarea, mostrar mensaje explicativo y Terminar la ejecución

Punto 3. **VERIFICAR LOS PERMISOS**

Si se detecta que algún archivo no tiene los permisos adecuados explicar la situación con un mensaje, configurar los permisos de los archivos correctamente.

Si no se puede efectuar la corrección, mostrar mensaje explicativo y Terminar la ejecución

Cuando todos los archivos tienen los permisos adecuados, Mostrar y Grabar en el log usando la función Logep el siguiente mensaje

```
Estado del Sistema: INICIALIZADO
```

Punto 4. **VER SI SE ARRANCA EL DEMONIO**

Initep debe ofrecer la posibilidad de arrancar el demonio Demonep

Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
¿Desea efectuar la activación de Demonep? Si – No
```

Punto 5. **NO ARRANCAR**

Si el usuario no desea arrancar el demonio Demonep, entonces explicar cómo hacerlo manualmente

Punto 6. **SI ARRANCAR**

Si el usuario desea arrancar el demonio Demonep, activarlo y explicar cómo detenerlo manualmente

Mostrar mensaje y grabar en el log

```
Demonep corriendo bajo el no.: <Process Id de Demonep>
```

Punto 7. **LOG**

Cerrar el archivo de log- Terminar el proceso

Punto 8. **NUEVA SESIÓN, NUEVO INICIO**

Al cerrar la sesión de usuario todos los valores de las variables de ambiente deben perderse, para poder ejecutar los comandos será necesario volver a inicializar el sistema

Demonep

Input

- Maestros y Tablas $GRUPO/DIRMAE
- Novedades $GRUPO/DIRREC/<nombre del archivo>

Output

- Archivos Aceptados $GRUPO/DIROK/<nombre del archivo>
- Archivos Rechazados $GRUPO/DIRNOK/<nombre del archivo>
- Log del Comando $GRUPO/DIRLOG/Demonep.log

Descripción

El propósito de este comando es detectar la llegada de archivos al directorio $GRUPO/DIRREC y aceptar o rechazar estos archivos según corresponda

- Es el segundo en orden de ejecución
- Es un proceso del tipo “Demonio” :
- Se dispara con Initep o manualmente
- Se detiene manualmente
- Mueve los archivos a través del Movep
- Graba en el archivo de Log a través del Logep
- Invoca, si corresponde, el siguiente proceso: Procep

El Proceso se inicia con la detección de la presencia de archivos en el directorio $GRUPO/DIRREC

Si el nombre del archivo (filename) cumple con el formato de nombre esperado y el archivo es de texto, el archivo se acepta, de lo contrario se lo rechaza.

También verifica si hay archivos ya aceptados para arrancar automáticamente el proceso Procep

Luego duerme un tiempo x y vuelve a a empezar, es decir, que a menos que se detenga, este proceso no tiene condición de fin.

A este tipo de programas se los denomina demonio, daemon o dæmon (de sus siglas en inglés Disk And Execution Monitor).

Otra característica de los procesos del tipo demonio, es que se ejecutan en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo).

Pasos Sugeridos:

Punto 1. OBLIGACIÓN DE INICIAR ANTES DE EJECUTAR CUALQUIER COMANDO

Si el ambiente no fue inicializado, no ejecutar

Mostrar mensaje adecuado y terminar el programa

Punto 2. MANTENER UN CONTADOR DE CICLOS DE EJECUCIÓN DE DEMONEP.

Se debe mantener un contador de ciclos de ejecución del demonio, grabar en el log el número de ciclo.

```
Demonep ciclo nro. 1
```

Punto 3. CHEQUEAR SI HAY ARCHIVOS EN EL DIRECTORIO $GRUPO/DIRREC

Si existen archivos en el directorio $GRUPO/DIRREC

Grabar en el log el nombre del archivo detectado

```
Archivo detectado: < nombre del archivo >
```

Punto 4. VERIFICAR QUE EL ARCHIVO SEA UN ARCHIVO COMÚN, DE TEXTO.

Los archivos que no sean de texto se rechazan, si este fuera el caso, grabar en el log

```
Archivo rechazado, motivo: no es un archivo de texto
```

Punto 5. VERIFICAR QUE EL ARCHIVO NO ESTÉ VACIO.

Los archivos vacios se rechazan, si este fuera el caso, grabar en el log

```
Archivo rechazado, motivo: archivo vacio
```

Punto 6. VERIFICAR QUE EL FORMATO DEL NOMBRE DEL ARCHIVO SEA CORRECTO

FORMATO CORRECTO: ejecutado_<año_presupuestario>_<cod_provincia>_<aniomesdia>.csv

Los archivos con nombres que no se correspondan con el formato de nombre esperado, se rechazan. Si este fuera el caso, grabar en el log

```
Archivo rechazado, motivo: formato de nombre incorrecto.
```

Punto 7. VERIFICAR AÑO

- Año presupuestario debe ser el año corriente
- Los archivos con error en el año, se rechazan. Si este fuera el caso, grabar en el log

```
Archivo rechazado, motivo: año xxxx incorrecto.
```

Punto 8. VERIFICAR PROVINCIA

- Cod_provincia debe existir en el maestro de provincias
- Los archivos con error en la provincia, se rechazan. Si este fuera el caso, grabar en el log

```
Archivo rechazado, motivo: provincia xx incorrecta.
```

Punto 9. VERIFICAR FECHA

- ANIOMESDIA debe ser una fecha válida
- ANIOMESDIA debe ser menor o igual a la fecha del día pero mayor o igual a la fecha de inicio del año presupuestario.
- Los archivos con error en la fecha, se rechazan. Si este fuera el caso, grabar en el log

```
Archivo rechazado, motivo: fecha aaaammdd incorrecta.
```

NOTA: si detecta otro tipo de error clasificable, puede incluir su validación.

Punto 10. **ACEPTAR LOS ARCHIVOS CON NOMBRE VÁLIDO.**

Si el nombre del archivo es válido Grabar en el log el mensaje de archivo aceptado

```
Archivo aceptado
```

Invocar la función Movep para mover el archivo aceptado al directorio DIROK

Punto 11. **RECHAZAR LOS ARCHIVOS INVÁLIDOS**

Invocar la función Movep para mover el archivo rechazado al directorio DIRNOK

Punto 12. **VER SI SE ARRANCA EL** Procep

Si en el directorio DIROK existen archivos aceptados (del presente ciclo o de ciclos anteriores), Demonep debe arrancar el Procep siempre que éste no esté corriendo.

Si arranca, grabar en el log

```
Procep corriendo bajo el no.: <Process Id de Procep>
```

Punto 13. **NO ARRANCAR**

Si correspondía arrancar Procep pero se debe posponer la ejecución de Procep porque ya hay un Procep corriendo, grabar en el log:

```
Invocación de Procep pospuesta para el siguiente ciclo
```

Punto 14. **DORMIR UN TIEMPO X Y EMPEZAR UN NUEVO CICLO**

El tiempo X para la primera entrega puede ser un tiempo largo, de varios minutos, el tiempo x para la segunda entrega debe ser un tiempo corto, no mayor al minuto.

Función Logep

Opciones y Parámetros

- Parámetro 1 (obligatorio): comando
- Parámetro 2 (obligatorio): mensaje
- Parámetro 3 (opcional): tipo de mensaje
- Otros parámetros u opciones a especificar por el desarrollador

Descripción

¿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. Es empleado por los profesionales de IT, auditoria y seguridad informática.

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

- WHO: ¿Quién?

  - Usuario, es el login del usuario
- WHEN: ¿Cuándo?
  - Fecha y Hora, en el formato que deseen y calculada justo antes de la grabación.
- WHERE: ¿Dónde?
  - Comando (parámetro 1), nombre del comando o función que genera el mensaje.
  - Se apreciará la utilidad de este parámetro por ejemplo cuando la función Movep deba generar mensajes en el log del comando llamador
- WHAT: ¿Qué?
  - Tipo de Mensaje (Parámetro 3) valores posibles:
    - INFO = INFORMATIVO: mensajes explicativos sobre el curso de ejecución del comando. Ej: Inicio de Ejecución
    - WAR = WARNING: mensajes de advertencia pero que no afectan la continuidad de ejecución del comando. Ej: Archivo duplicado
    - ERR = ERROR: mensajes de error Ej: Archivo Inexistente.
- WHY: ¿Por qué?
  - Mensaje (Parámetro 2)

Requerimientos

- El directorio de log está determinado por la variable de ambiente LOGDIR
- El nombre del archivo de log es igual al nombre del comando y extensión .log
- Grabar un archivo distinto para cada comando, en el lugar indicado y con el nombre adecuado.
- La escritura de archivos de log debe ser homogénea para todos los comandos
- Cada registro de log debe responder al estándar W5.
- Si el archivo de log no existe, se debe crear. Si existe se le deben agregar los nuevos registros **siempre**
- Se debe controlar el crecimiento del archivo de log
  - En todo sistema, es importante evitar el crecimiento INDISCRIMINADO de los archivos de Log. Es por ello que esta función debe preveer un mecanismo para controlarlo y evitarlo
  - En el log deben permanecer siempre los mensajes mas recientes
  - Cada vez que se hace la reducción del tamaño del archivo, señalizar la situación con el mensaje “Log Excedido” en el propio log
  - aclare en **Hipótesis y Aclaraciones Globales** cual fue el mecanismo que adoptó.
    - **Ejemplo sencillo**: cuando un archivo de log supera un tamaño X, el archivo se trunca dejando las últimas xx líneas

Función Movep

Opciones y Parámetros

- Parámetro 1 (obligatorio): origen
- Parámetro 2 (obligatorio): destino
- Parámetro 3 (opcional): comando que la invoca
- Otros parámetros u opciones a especificar por el desarrollador

Descripción

Esta función tiene por objeto centralizar el movimiento de archivos que deben realizar la mayor parte de los comandos de este sistema para evitar diferentes políticas en el tratamiento de archivos duplicados.

En líneas generales el sistema no borra ningún archivo, los mueve de un lugar a otro, en el contexto de este TP se deben CONSERVAR todos los archivos aun cuando:

- sea un archivo improcesable, roto, dañado, vacio;
- sea un archivo con un nombre incorrecto, con espacios, mal formado;
- ya haya sido procesado
- al moverlo nos encontremos que ya existe otro archivo del mismo nombre en ese lugar.

Requerimientos

- Mover el archivo solicitado al directorio indicado sin alterar su contenido
- Si en el destino ya existe otro archivo con el mismo nombre (nombre de archivo duplicado), no debe fracasar la operación, la función debe poder conservar ambos.
    - Crear un subdirectorio /dpl para depositar el archivo duplicado y moverlo allí
        - Si también en /dpl ya existe otro archivo con el mismo nombre, emplear un numero secuencial (desde 1 hasta n) para modificar el nombre del archivo y ponerlo como complemento final de la extensión: Ejemplo: <nombre del archivo original>.nnn dónde nnn es el número de secuencia
- Si esta función es invocada por un comando que graba en un archivo de log, registrar el resultado de su uso en el log del comando

Premisas:

- No se puede definir un archivo auxiliar solo para registrar el número de secuencia, usar el archivo de configuración del sistema para estos propósitos
- Si el origen y el destino son iguales, no mover y registrar en el log el error
- Si el origen no existe, no mover y registrar en el log el error
- Si el destino no existe, no mover y registrar en el log el error
- Indicar en las Hipótesis Globales cuantos números de secuencia emplea: uno por directorio o uno para toda la instalación

Procep

Falta especificar

Listep

Falta especificar

Estructuras y Archivos

Maestro de Centros Presupuestarios: DIRMAE/centros.csv

Centro de Presupuesto	Nombre
0.0.0.1	Unidad Ministro
0.0.0.1-1	Secretaria de Coordinacion Ministerial
0.0.0.1-1-1	SubSubsecretaria de Administracion General
0.0.0.1-1-1.1	Direccion General Alfa

0.0.0.1-1-1.2	Direccion General Prima
0.0.0.1-1-2	SubSubsecretaria de Infraestructura Ministerial
0.0.0.1-2	Secretaria de Trabajo
0.0.0.1-2-1	SubSecretaria de Fiscalizacion
0.0.0.1-2-1.1	Direccion General Baires
0.0.0.1-2-1.2	Direccion General Sancor
0.0.0.1-2-1.3	Direccion General Norte
0.0.0.1-2-1.4	Direccion General Sur
0.0.0.1-3	Secretaria de Empleo
0.0.0.1-3-1	Subsecretaria D
0.0.0.1-3-1.1	Direccion General Delta
0.0.0.1-3-2	Subsecretaria E
0.0.0.1-3-2.1	Direccion General Epsilon

Maestro de Provincias: DIRMAE/provincias.csv

Codigo de provincia	Provincia	Region
1	CABA	Baires
2	Buenos Aires	Baires
3	Catamarca	Norte
4	Chaco	Norte
5	Chubut	Sur
6	Cordoba	Sancor
7	Corrientes	Norte
8	Entre Rios	Norte
9	Formosa	Norte
10	Jujuy	Norte
11	La Pampa	Sur
12	La Rioja	Norte
13	Mendoza	Sur
14	Misiones	Norte
15	Neuquen	Sur
16	Rio Negro	Sur
17	Salta	Norte
18	San Juan	Sur
19	San Luis	Sur
20	Santa Cruz	Sur
21	Santa Fe	Sancor
22	Santiago del Estero	Norte
23	Tierra del Fuego	Sur
24	Tucuman	Norte

Tabla de Trimestres: DIRMAE/trimestres.csv

Año Presupuestario	Trimestre	Desde	Hasta
2016	Primer Trimestre 2016	10/12/2015	9/3/2016
2016	Segundo Trimestre 2016	10/3/2016	9/6/2016
2016	Tercer Trimestre 2016	10/6/2016	9/9/2016

2016	Cuarto Trimestre 2016	10/9/2016	9/12/2016
2016	Anual 2016	10/12/2015	9/12/2016
2017	Primer Trimestre 2017	10/12/2016	9/3/2017
2017	Segundo Trimestre 2017	10/3/2017	9/6/2017
2017	Tercer Trimestre 2017	10/6/2017	9/9/2017
2017	Cuarto Trimestre 2017	10/9/2017	9/12/2017
2017	Anual 2017	10/12/2016	9/12/2017

Archivos de Log: DIRLOG/<nombre del comando>.log

Campo	Descripción /Fuente/Valor	
Quien	Caracteres	Es el login del usuario que graba el registro
Cuando	Fecha y hora	Formato a Elección Es la fecha y hora en el momento de grabación del registro.
Donde	Caracteres	Nombre del Comando, función o rutina en donde se produce el evento que se registra en el log
Que	Caracteres	Lo determina el programador.
Porque	Caracteres	Lo determina el programador.

Separador de campos: - guion

Ejemplo: 20160905 19:53:22-Sandra-Demonep-WAR-No se pudo mover el archivo

En la variable de ambiente LOGSIZE se tiene el tamaño máximo en Kbytes que puede alcanzar un archivo de log

Archivo de Configuración /dirconf/Instalep.conf

Campo	**Descripción/Fuente/Valor**	
Variable	Caracteres	Valores posibles: GRUPO, DIRBIN, DIRMAE, etc
Valor	Caracteres	Contenido de la variable al momento de la grabación.
Usuario	Caracteres	Es el login del usuario que graba el registro
Fecha	Fecha y hora	Formato a Elección. Es la fecha y hora en el momento de grabación del registro.

Separador de campos: = igual

Ejemplo: GRUPO=/usr/alumnos/temp/grupo01=alumnos=09/04/2015 10:03 p.m

Se debe grabar un registro por cada variable. Luego de los registros requeridos, puede agregar todos los registros que desee

NOTA: más archivos en la próxima entrega

7.2. Enunciado: Segunda parte

Apéndice II – Temas pares

Enunciado segunda parte

Procep

Input

- Maestro de centros — DIRMAE/centros.csv
- Maestro de actividades — DIRMAE/actividades.csv
- Tabla de trimestres — DIRMAE/trimestres.csv
- Maestro de provincias — DIRMAE/provincias.csv
- Archivos provinciales con presupuesto ejecutado — DIROK/ejecutado_<año presupuestario>_<cod provincia>_<aniomesdia>.csv

Output

- Archivos Procesados — DIRPROC/proc/<nombre del archivo>
- Archivo anual con presupuesto ejecutado — DIRPROC/ejecutado-<año presupuestario>
- Archivo anual con registros rechazados — DIRPROC/rechazado-<año presupuestario>
- Archivos Rechazados — DIRNOK/<nombre del archivo>
- Log del Comando — DIRLOG/Procep.log

Descripción

El propósito de este proceso es reunir en un solo archivo todos los registros de presupuesto ejecutado validos

- Es el tercero en orden de ejecución
- Es invocado por Demonep
- Graba los registros validos en un archivo de salida
- Graba los registros rechazados en un archivo de rechazos indicando su motivo
- No debe procesar dos veces un mismo archivo
- Mueve los archivos a través del Movep
- Graba en el archivo de Log a través del Logep

Los archivos de novedades contienen el presupuesto ejecutado, Pueden existir varios registros de gastos para la misma fecha-centro-actividad-trimestre. Ejemplo

id	Fecha **(formato aaaammdd)**	Centro de Presupuesto	Actividad	Trimestre	Gasto
1	20151210	0.0.0.1	Capacitacion Continua	Primer Trimestre 2016	22
2	20151210	0.0.0.1-2-1.1	Desarrollo de Procesos	Primer Trimestre 2016	92,5
3	20151210	0.0.0.1-2-1.1	Fiscalizacion Buenos Aires	Primer Trimestre 2016	111
4	20151210	0.0.0.1-2-1.1	Fiscalizacion Buenos Aires	Primer Trimestre 2016	111
5	20151210	0.0.0.1-2-1.1	Fiscalizacion Buenos Aires	Primer Trimestre 2016	111
6	20151210	0.0.0.1-2-1.1	Fiscalizacion Buenos Aires	Primer Trimestre 2016	118,75
7	20151210	0.0.0.1-2-1.1	Desarrollo de Procesos	Primer Trimestre 2016	157,5
8	20151210	0.0.0.1-2-1.1	Fiscalizacion Buenos Aires	Primer Trimestre 2016	189
9	20151210	0.0.0.1-2-1.1	Fiscalizacion Buenos Aires	Primer Trimestre 2016	189
10	20151210	0.0.0.1-2-1.1	Fiscalizacion Buenos Aires	Primer Trimestre 2016	189

Pasos Sugeridos:

Punto 1. OBLIGACIÓN DE INICIAR ANTES DE EJECUTAR CUALQUIER COMANDO

Si el ambiente no fue inicializado, no ejecutar

Mostrar mensaje adecuado y terminar el programa

Punto 2. PROCESAR TODOS LOS ARCHIVOS

Grabar en log

```
Cantidad de archivos a procesar:<cantidad>
```

Punto 3. ANTES DE PROCESAR UN ARCHIVO, VERIFICAR QUE NO ESTA DUPLICADO

Cada vez que se procesa un archivo, se lo mueve tal cual fue recibido y con el mismo nombre a DIRPROC/proc/<nombre del archivo>

Es por ello que es posible detectar antes de intentar procesar un archivo si ya fue procesado solo inspeccionando el contenido de ese directorio. Si ya fue procesado, rechazar el archivo completo y grabar en el log un mensaje aclaratorio, como ser:

```
Archivo Duplicado. Se rechaza el archivo <nombre del archivo>
```

El archivo duplicado se lo mueve a DIRNOK empleando la función Movep.

Punto 4. ANTES DE PROCESAR UN ARCHIVO, VERIFICAR EL FORMATO

A efectos de este TP, bastara con verificar el primer registro para determinar si el formato es correcto.

Si la cantidad de campos del primer registro no se corresponde con el formato establecido, asumir que el archivo está dañado, rechazar el archivo completo y grabar en el log un mensaje aclaratorio, como ser:

```
Estructura inesperada. Se rechaza el archivo <nombre del archivo>.
```

El archivo rechazado se lo mueve a DIRNOK empleando la función Movep.

Punto 5. VALIDAR LOS CAMPOS DE LA SIGUIENTE FORMA

Grabar en log

```
Archivo a procesar <nombre del archivo>.
```

Validación Centro de Presupuesto: El centro de presupuesto debe existir en el maestro de centros

Validación de Actividad: La actividad debe existir en el maestro de actividades

Validación Trimestre: El trimestre debe existir en la tabla de trimestres y el trimestre debe ser del año presupuestario corriente

Validación de Fecha: la fecha indicada en el registro debe ser una fecha valida; menor o igual a la fecha del nombre del archivo y estar comprendida entre la fecha de inicio del trimestre (FDESDE_TRI) y la fecha de fin del trimestre (FHASTA_TRI)

Validación de Gasto: el importe de gasto debe ser mayor a cero.

NOTA: si detecta otro tipo de error clasificable, puede incluir su validación.

Punto 6. GRABAR REGISTRO VALIDADO

Si el registro supera estas validaciones se debe grabar en el archivo anual con presupuesto ejecutado (DIRPROC/ejecutado-<año presupuestario>) un registro con la estructura indicada.

Incrementar en uno el contador de registros validados ok.

Al finalizar el proceso grabar en el log la cantidad de registros validados ok.

Punto 7. GRABAR REGISTRO RECHAZADO

Si el registro NO supera estas validaciones se debe rechazar solo ese registro y grabar en el archivo anual con registros rechazados (DIRPROC/rechazado-<año presupuestario>) un registro con la estructura indicada.

Nota: al menos considerar describir los siguientes motivos de rechazo:

- Centro inexistente
- Actividad inexistente

- Trimestre invalido
- Fecha invalida
- La fecha no se corresponde con el trimestre indicado
- Importe invalido
- Cualquier otro motivo que considere adecuado

Incrementar en uno el contador de registros rechazados.

Al finalizar el proceso grabar en el log la cantidad de registros rechazados.

Punto 8. FIN ARCHIVO

- MOVER ARCHIVO PROCESADO

  Cuando se termina de procesar el archivo, para evitar su reprocesamiento, se debe mover el archivo procesado a DIRPROC/proc/<nombre del archivo> empleando la función Movep.

- GRABAR TOTALES POR ARCHIVO EN LOG

  Cuando se termina de procesar el archivo, registrar en el log la cantidad de registros leidos, cuantos fueron rechazados y cuantos validaron bien

- Continuar con el siguiente archivo

Listep

Input

- Maestros y Tablas — DIRMAE
- Tabla de actividades por centro — DIRMAE/tabla-AxC.csv
- Presupuesto sancionado — DIRMAE/sancionado-<año presupuestario>.csv
- Archivo anual con presupuesto ejecutado — DIRPROC/ejecutado-<año presupuestario>

Output

- Listados — DIRINFO/<nombre archivo>

Requisitos

- Debe estar programado en PERL
- Debe emplear estructuras hash
- Debe presentar un menú amigable y una opción de ayuda del comando
- Debe presentar una opción de grabación del resultado en archivo de salida
- Debe mostrar siempre el resultado por pantalla, independientemente si graba o no archivo
- Incluir todas las opciones y parámetros indicadas en la descripción de la forma que considere mas adecuada
- queda a criterio del desarrollador que campos mostrar en cada caso, los ejemplos son solo ilustrativos
- queda a criterio del desarrollador los nombres de los archivos de salida
- No sobreescribir archivos de salida

Descripción

El propósito de este proceso es generar diferentes listados de control entre el presupuesto sancionado y el presupuesto ejecutado.

- Es el cuarto en orden de ejecución
- Se dispara manualmente
- No graba en el archivo de log

Las partidas presupuestarias están determinadas por el presupuesto sancionado del año presupuestario corriente.

El archivo de presupuesto sancionado tiene la siguiente estructura y ejemplo:

Centro de Presupuesto	Trimestre	Financiacion Fuente 11	Financiacion Fuente 21
0.0.0.1	Primer Trimestre 2016	1600,00	400,00
0.0.0.1-1	Primer Trimestre 2016	0,00	32,24
0.0.0.1-1-1	Primer Trimestre 2016	0,00	24,40
0.0.0.1-1-1.1	Primer Trimestre 2016	121,00	0,00
0.0.0.1-1-1.2	Primer Trimestre 2016	123,00	0,00
0.0.0.1-1-2	Primer Trimestre 2016	0,00	54,00

El presupuesto sancionado para un centro se establece por periodos trimestrales

Las dos fuentes de financiación (fuente 11 = Tesoro, fuente 21=fondos propios) serán tomadas indistintamente para realizar la imputación de gastos. Es decir que a efectos de este TP, se deben sumar ambas fuentes para conocer cuál es el **presupuesto sancionado** para un centro en un trimestre.

Listados de presupuesto sancionado

Listar el presupuesto sancionado para un año presupuestario pasado como parámetro

Opción ct) listar los registros ordenados por código de centro y luego por trimestre (primer trimestre, segundo trimestre, tercer trimestre, cuarto trimestre) mostrando el total por centro (F11_FIN_SAN + F21_FIN_SAN)

Ejemplo:

Año presupuestario 2016	Total Sancionado
Primer Trimestre 2016	2000,00
Segundo Trimestre 2016	2000,00
Tercer Trimestre 2016	2000,00
Cuarto Trimestre 2016	2000,00
Total Unidad Ministro	8000,00

…

Total Anual	41033,34

Opción tc) idem anterior pero ordenado primero por trimestre y luego por código de centro

Ejemplo:

Año presupuestario 2016	Total Sancionado
Unidad Ministro	2000,00
Secretaria de Coordinacion Ministerial	32,24
SubSubsecretaria de Administracion General	24,40
Direccion General Alfa	121,00
Direccion General Prima	123,00
SubSubsecretaria de Infraestructura Ministerial	54,00
Secretaria de Trabajo	836,62
SubSecretaria de Fiscalizacion	727,50
Direccion General Baires	2100,00
Direccion General Sancor	1450,00
Direccion General Norte	500,00
Direccion General Sur	300,00
Secretaria de Empleo	173,55
Subsecretaria D	30,02
Direccion General Delta	300,20
Subsecretaria E	99,25
Direccion General Epsilon	397,00
Total Primer Trimestre 2016	9268,78

…

Total Anual	41033,34

Listados de presupuesto ejecutado

Listar el presupuesto ejecutado para un año presupuestario pasado como parámetro

Que filtros se deben permitir?

Permitir filtrar los registros por actividad

- Una actividad, pe Desarrollo de Aplicativos
- Varias actividades
- Todas las actividades

Cuando un gasto esta fuera de la planificación?

- Cada centro de presupuesto gasta su partida asignada ejecutando distintas actividades.
- Existe una tabla de Actividades por Centro que nos permite verificar si la actividad a la cual se imputa el gasto es una actividad planificada.
- Si la combinación de centro y actividad ejecutada no existe en la tabla AxC, el centro realizó un **"gasto fuera de planificación"**
- **Indicar en el listado donde se presenta esta situación**

Mostrar siempre el total de gastos

Ejemplo: actividad Credito Fiscal todos los centros, trimestres

Fecha	Centro	Nom Cen	cod Act	Actividad	Trimestre	Gasto	provincia	control
20160107	0.0.0.1	Unidad Ministro	e.m.f3	Credito Fiscal	Primer Trimestre 2016	66,00	Cordoba	
20160122	0.0.0.1	Unidad Ministro	e.m.f3	Credito Fiscal	Primer Trimestre 2016	138,75	Formosa	
20160610	0.0.0.1	Unidad Ministro	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	88,00	Buenos Aires	
20160611	0.0.0.1	Unidad Ministro	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	99,00	Buenos Aires	
20160627	0.0.0.1	Unidad Ministro	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	66,00	CABA	
20160711	0.0.0.1	Unidad Ministro	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	83,25	Chubut	
20160814	0.0.0.1-1-1.1	Direccion General Alfa	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	161,00	Santa Cruz	gasto fuera de la planificación
20160903	0.0.0.1-2	Secretaria de Trabajo	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	378,00	Santa Cruz	gasto fuera de la planificación
20160727	0.0.0.1-2-1	SubSecretaria de Fiscalizacion	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	174,83	Santiago del Estero	gasto fuera de la planificación
20160821	0.0.0.1-3-2	Subsecretaria E	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	100,00	Rio Negro	
20160507	0.0.0.1-3-2.1	Direccion General Epsilon	e.m.f3	Credito Fiscal	Segundo Trimestre 2016	100,00	Rio Negro	
20160610	0.0.0.1-3-2.1	Direccion General Epsilon	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	100,00	Buenos Aires	
20160628	0.0.0.1-3-2.1	Direccion General Epsilon	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	100,00	CABA	
20160801	0.0.0.1-3-2.1	Direccion General Epsilon	e.m.f3	Credito Fiscal	Tercer Trimestre 2016	100,00	Entre Rios	
					Total Credito Fiscal	**1754,83**		

Listados de control del presupuesto ejecutado

Listar los saldos por centro/trimestre del presupuesto ejecutado

El primer movimiento de un trimestre siempre debe ser el correspondiente al presupuesto sancionado, para ello es útil considerar la fecha del movimiento = fecha de inicio del trimestre (FDESDE_TRI)

Imputar los gastos en orden cronológico, del mas antiguo al mas reciente

El monto correspondiente al presupuesto sancionado debe considerarse con signo contrario a los gastos para poder realizar el cálculo del saldo (sancionado "suma" y los gastos "restan")

Cuando un centro excede al presupuesto?

Si dentro de un trimestre hay gastos del centro que superan el saldo disponible de presupuesto sancionado, hay **"presupuesto excedido"**

Indicar en el listado donde se presenta esta situación

Que saldos puedo mostrar?

- Saldo por trimestre (ver ejemplo abajo)
- Saldo acumulado (acarrea el saldo del trimestre anterior)

Que filtros se deben permitir?

Permitir filtrar los registros por trimestre

- Un trimestre (cuando se pide un solo trimestre no es posible calcular el indicador de presupuesto compensado)
- Varios trimestres
- todos los trimestres

Permitir filtrar los registros por centro

- Un centro: p. e 0.0.0.1-3-2.1
- Varios centros
- Un rango de centros, por ejemplo 0.0.0.1-3* (incluye los centros 0.0.0.1-3, 0.0.0.1-3-1, 0.0.0.1-3-1.1, 0.0.0.1-3-2, 0.0.0.1-3-2.1)
- Todos los centros

Ejemplo: centro 0.0.0.1-2-1 todos los trimestres ejecutados

ID	FECHA MOV	CENTRO	ACTIVIDAD	TRIMESTRE	IMPORTE	SALDO por TRIMESTRE	CONTROL	SALDO ACUMULADO
(++)	20151210	0.0.0.1-2-1	0	Primer Trimestre 2016	727,50	727,50		727,50
209	20160223	0.0.0.1-2-1	Capacitacion Continua	Primer Trimestre 2016	174,83	552,68		552,68
207	20160224	0.0.0.1-2-1	Capacitacion Continua	Primer Trimestre 2016	102,68	450,00		450,00
131	20160121	0.0.0.1-2-1	Desarrollo de Procesos	Primer Trimestre 2016	138,75	311,25		311,25
280	20160301	0.0.0.1-2-1	Desarrollo de Procesos	Primer Trimestre 2016	127,26	183,99		183,99
275	20160226	0.0.0.1-2-1	Fiscalizacion Santiago del Estero	Primer Trimestre 2016	138,75	45,24		45,24
278	20160302	0.0.0.1-2-1	Laboratorios de Emprendimiento	Primer Trimestre 2016	74,74	-29,50	gasto fuera de planificación presupuesto excedido	-29,50
(++)	20160310	0.0.0.1-2-1	0	Segundo Trimestre 2016	727,50	727,50		698,00

18	20160310	0.0.0.1-2-1	Capacitacion Continua	Segundo Trimestre 2016	138,75	588,75		559,25
61	20160406	0.0.0.1-2-1	Capacitacion Continua	Segundo Trimestre 2016	139,86	448,89		419,39
53	20160408	0.0.0.1-2-1	Capacitacion Continua	Segundo Trimestre 2016	82,14	366,75		337,25
221	20160525	0.0.0.1-2-1	Desarrollo de Aplicativos	Segundo Trimestre 2016	82,14	284,61	gasto fuera de planificación	255,11
26	20160315	0.0.0.1-2-1	Desarrollo de Procesos	Segundo Trimestre 2016	166,50	118,11		88,61
222	20160524	0.0.0.1-2-1	Desarrollo de Procesos	Segundo Trimestre 2016	139,86	-21,75	presupuesto excedido	-51,25
227	20160527	0.0.0.1-2-1	Fiscalizacion Santa Cruz	Segundo Trimestre 2016	166,50	-188,25	presupuesto excedido	-217,75
251	20160531	0.0.0.1-2-1	Fiscalizacion Santa Fe	Segundo Trimestre 2016	166,50	-354,75	presupuesto excedido	-384,25
(++)	20160610	0.0.0.1-2-1	0	Tercer Trimestre 2016	727,50	727,50		343,25
195	20160822	0.0.0.1-2-1	Capacitacion Continua	Tercer Trimestre 2016	138,75	588,75		204,50
230	20160902	0.0.0.1-2-1	Capacitacion Continua	Tercer Trimestre 2016	174,83	413,93		29,68
229	20160904	0.0.0.1-2-1	Capacitacion Continua	Tercer Trimestre 2016	102,68	311,25		-73,00
269	20160727	0.0.0.1-2-1	Credito Fiscal	Tercer Trimestre 2016	174,83	136,43	gasto fuera de planificación	-247,83
93	20160713	0.0.0.1-2-1	Desarrollo de Procesos	Tercer Trimestre 2016	174,83	-38,40	presupuesto excedido	-422,65
92	20160714	0.0.0.1-2-1	Desarrollo de Procesos	Tercer Trimestre 2016	102,68	-141,08	presupuesto excedido	-525,33
267	20160728	0.0.0.1-2-1	Desarrollo de Procesos	Tercer Trimestre 2016	102,68	-243,75	presupuesto excedido	**-628,00 (*)**

(*) saldo al final de los tres trimestres

(++) movimientos provenientes del presupuesto sancionado

Estructuras y Archivos

Maestro de Centros Presupuestarios: DIRMAE/centros.csv

Separador de campos: ; punto y coma

Data Set publicado en datos.zip

Campo	Descripción /Fuente/Valor
COD_CEN	Código del Centro, N caracteres
NOM_CEN	Nombre del Centro, N caracteres

Maestro de Provincias: DIRMAE/provincias.csv

Separador de campos: ; punto y coma

Data Set publicado en datos.zip

Campo	Descripción /Fuente/Valor
COD_PROV	Código de la Provincia, Numérico
NOM_PROV	Nombre de la Provincia, N caracteres
REG_PROV	Región a la cual pertenece la provincia, N caracteres

Tabla de Trimestres: DIRMAE/trimestres.csv

Separador de campos: ; punto y coma

Data Set publicado en datos.zip

Campo	Descripción /Fuente/Valor
ANO_TRI	Año presupuestario al cual pertenece el trimestre, Numérico
NOM_TRI	Nombre del Trimestre, N caracteres
FDESDE_TRI	Fecha de inicio del trimestre, dd/mm/aaaa
FHASTA_TRI	Fecha de fin del trimestre, dd/mm/aaaa

Maestro de Actividades: DIRMAE/actividades.csv

Separador de campos: ; punto y coma

Data Set publicado en datos.zip

Campo	Descripción /Fuente/Valor
COD_ACT	Código de la Actividad, N caracteres
OBJ_ACT	Objetivo estratégico al cual pertenece la actividad, N caracteres
PGM_ACT	Programa en el cual se encuadra la actividad, N caracteres
NOM_ACT	Nombre de la actividad, N caracteres

Tabla de Actividades por Centros: DIRMAE/tabla-AxC.csv

Separador de campos: ; punto y coma

Data Set publicado en datos.zip

Campo	Descripción /Fuente/Valor
ACT_AXC	Código de la Actividad, N caracteres
CEN_AXC	Código del Centro, N caracteres

Presupuesto Sancionado: DIRMAE/sancionado-<año_presupuestario>.csv

Separador de campos: ; punto y coma

Data Set publicado en datos.zip sancionado-2016.csv

Data Set publicado en Lote2.zip sancionado-2015.csv

Campo	Descripción /Fuente/Valor
COD_CEN_SAN	Código del centro al cual pertenece esta línea de presupuesto sancionado, N caracteres
NOM_TRI_SAN	Nombre del trimestre al cual pertenece esta línea de presupuesto sancionado, N caracteres
F11_FIN_SAN	Monto de financiamiento a través de Fuente 11 (tesoro) , Importe positivo
F21_FIN_SAN	Monto de financiamiento a través de Fuente 21 (recursos propios) , Importe positivo

Archivos de novedades: DIRREC/<nombre del archivo>

Archivos rechazados: DIRNOK/<nombre del archivo>

Archivos provinciales con presupuesto ejecutado: DIROK/ejecutado_<año_presupuestario>_<cod_provincia>_<aniomesdia>.csv

Separador de campos: ; punto y coma

Data Sets publicados en Lote2.zip

ejecutado_2016_16_20160824.csv
ejecutado_2016_20_20160907.csv
ejecutado_2016_22_20160731.csv

Data Sets publicados en datos.zip:

ejecutado_2016_10_20160811.csv
ejecutado_2016_18_20160911.csv
ejecutado_2016_1_20151222.csv
ejecutado_2016_1_20160413.csv
ejecutado_2016_1_20160722.csv
ejecutado_2016_21_20160909.csv
ejecutado_2016_2_20151210.csv
ejecutado_2016_2_20160310.csv
ejecutado_2016_2_20160402.csv
ejecutado_2016_2_20160621.csv
ejecutado_2016_3_20160713.csv
ejecutado_2016_4_20160709.csv
ejecutado_2016_5_20160722.csv
ejecutado_2016_6_20160209.csv
ejecutado_2016_6_20160505.csv
ejecutado_2016_6_20160808.csv
ejecutado_2016_7_20160803.csv
ejecutado_2016_8_20160802.csv
ejecutado_2016_9_20160823.csv

Campo	Descripción /Fuente/Valor
ID_EJE	Id del registro, Numérico
FECHA_EJE	Fecha de ejecución, formato aaaammdd
COD_CEN_EJE	Código del centro ejecutor, N caracteres
NOM_ACT_EJE	Nombre de la actividad ejecutada, N caracteres
NOM_TRI_EJE	Nombre del trimestre al cual pertenece esta línea de presupuesto ejecutado, N caracteres
GASTO_EJE	Monto del gasto efectuado sin discriminación de la fuente de financiación , Importe positivo

Archivos procesados: DIRPROC/proc/<nombre del archivo>

Archivo anual con presupuesto ejecutado: DIRPROC/ejecutado_<año_presupuestario>

Separador de campos: ; punto y coma

Campo	Descripción /Fuente/Valor
id	Id del registro proveniente del registro original
Fecha	Fecha de ejecución proveniente del registro original, al grabar, puede modificar el formato de esta fecha a su elección
Centro de Presupuesto	Código del centro ejecutor proveniente del registro original
Actividad	Nombre de la actividad ejecutada proveniente del registro original
Trimestre	Nombre del trimestre proveniente del registro original
Gasto	Monto del gasto proveniente del registro original
Archivo Origen	Nombre del archivo de origen del registro
COD_ACT	Código de la Actividad proveniente del maestro de actividades
NOM_PROV	Nombre de la Provincia proveniente del maestro de provincias
NOM_CEN	Nombre del Centro ejecutor proveniente del maestro de centros

Archivo anual con registros rechazados
DIRPROC/rechazado_<año_presupuestario>

Separador de campos: ; punto y coma

Campo	Descripción /Fuente/Valor
Fuente	Nombre del archivo de input
Motivo	Motivo por el cual se rechaza este registro
Registro de Oferta	Registro Original COMPLETO
usuario	Login del usuario que graba el registro
fecha	Fecha y hora de grabación del registro rechazado, en el formato que se desee

FIN