



Tema	Grupo	Ayudante	Evaluación Grupal del TP			
<b>J</b>			Nota			
			Fecha			
			Firma			
<b>Integrantes</b>						
Padrón	Apellido	Nombre	E-mail	Asistencias		Evaluación Individual
<b>Evaluación Grupal de la Entrega</b>						
	<b>Aprobada</b>	<b>Se Debe Rehacer</b>	<b>No Entrega</b>	<b>Observaciones</b>		
Presentación						
Hipótesis Planteadas						
Problemas Relevantes						
Funcionamiento General						
<b>Comandos</b>						
INSTALAR						
INICIAR						
DETECTAR						
INTERPRETE						
REPORTE						
<b>Funciones de Apoyo y Comandos Auxiliares</b>						
Mover						
GLog						
VLog						
<b>Archivos Auxiliares</b>						



# Índice

<b>Condiciones de Resolución y Corrección.....</b>	<b>3</b>
Introducción .....	3
Documentación a Presentar .....	3
Observaciones a tener en cuenta en el desarrollo.....	3
<b>Especificación de Funciones.....</b>	<b>3</b>
Nombre de la función: Mover .....	3
Nombre de la función: Glog.....	3
Nombre de la función: Vlog .....	3
<b>Especificación de Comandos.....</b>	<b>3</b>
Comando de Instalación.....	3
Comando de Inicialización de Ambiente .....	3
Comando de Detección de Archivos .....	3
Comando de Interpretación de Archivos .....	3
Comando de Impresión .....	3
<b>Estructuras de Archivos .....</b>	<b>3</b>
Configuración \$grupo/conf/practico.conf.....	3
Log \$grupo/logdir/<nombre del comando>.log.....	3
Países y Sistemas \$grupo/conf/p-s.tab.....	3
Separadores \$grupo/conf/T1.tab.....	3
Campos \$grupo/conf/T2.tab.....	3
Archivo de Contratos de Préstamos Personales \$grupo/datadir/new/CONTRAT.<pais> .....	3
Archivo Maestro Contable de Préstamos Personales Impagos \$grupo/datadir/mae/PPI.mae .....	3
Archivo de Modificaciones de Contratos de Préstamos Personales \$grupo/datadir/new/MODIF.<pais> .....	3
Listados .....	3
Datos .....	3



## Condiciones de Resolución y Corrección

### Introducción

1. Se deberá tener en cuenta para la resolución TODAS las condiciones que se enuncian.
2. Se deben respetar los formatos de archivos especificados
3. Se debe respetar la estructura de directorios planteada
4. Cada comando deberá ser resuelto utilizando el lenguaje indicado.
5. El día de vencimiento del TP, cada ayudante convocará a los integrantes de un grupo e iniciará la corrección mediante una entrevista grupal. Es imprescindible la presencia de todos los integrantes del grupo el día de la corrección. El objetivo de esto es comprender la dinámica de trabajo del equipo y los roles desempeñados por cada uno.
6. Posterior a la entrega se podrá acordar entre el Ayudante y el Grupo el intercambio de correspondencia a través de la cuenta [so7508@gmail.com](mailto:so7508@gmail.com). Por este medio el ayudante podrá solicitar a los alumnos correcciones, mejoras en las instrucciones de ejecución, nuevos juego de datos, etc. Cabe aclarar que todos los mensajes a esta casilla deberán tener como asunto "Grupox (xx es el número asignado al grupo) para poder redireccionar el mail al ayudante correspondiente.
7. Dentro de los ítems a chequear el ayudante evaluará **aspectos formales** (como ser la forma de presentación de la carpeta), **aspectos funcionales** (como ser que el TP funcione) y **aspectos operativos**: que el TP resuelva el problema planteado.

### Documentación a Presentar

Se deberá hacer entrega de:

- Una Carpeta de Trabajo Práctico con el contenido solicitado mas adelante. En el pie de cada hoja se debe incluir: Número de Grupo y Tema (en el margen izquierdo) y Número de Hoja (en el margen derecho). TODAS las hojas deberán estar numeradas y enganchadas a una carpeta. Las hojas sueltas no se considerarán como parte de la misma.
- Un archivo con el formato y contenido solicitado mas adelante

### Contenido de la Carpeta

1. **Carátula**: La entregada en este mismo documento con los datos completos en 2 COPIAS.
2. **Índice del Contenido de la Carpeta**.
3. **Hipótesis y Aclaraciones**: todas las hipótesis que han sido consideradas en la resolución del TP y cualquier otra aclaración que se considere necesaria.
4. **Problemas relevantes**: Enumeración de los problemas relevantes que se les hayan presentado durante el desarrollo y prueba de los comandos y como los solucionaron.
5. **Archivo README: Instructivo de Instalación**
6. **Comandos Desarrollados**: para cada comando desarrollado incluir cuando corresponda:
  - a. **Nombre del Comando**
  - b. **Tipo de comando**: Solicitado o Auxiliar (los comandos auxiliares son aquellos que no forman parte del set de comandos solicitados.)
  - c. **Justificación de su uso** (solo para los comandos auxiliares)
  - d. **Archivos de Input**
  - e. **Archivos de Output**
  - f. **Parámetros**, si corresponde
  - g. **Opciones**, si corresponde
  - h. **Ejemplos de invocación** con distintos parámetros y valores, indicando en que oportunidad es usado y cuál es el resultado previsto.
  - i. **Listado del código** del comando: éste deberá incluir comentarios útiles para la comprensión del código.
7. **Archivos**
  - a. **Archivos del enunciado**: Por cada archivo del enunciado indicar:
    - i. Nombre del Archivo y Estructura.
  - b. **Archivos auxiliares**: Por cada archivo auxiliar que se utilice indicar:
    - i. Nombre del Archivo y Estructura
    - ii. Justificación de su uso
8. **Apéndice A**
  - a. **Enunciado COMPLETO**: es el presente documento.



## Formato y contenido del Archivo

El archivo de Trabajo Práctico deberá ser enviado a la cuenta [so7508@gmail.com](mailto:so7508@gmail.com) dentro de las 24hs posteriores a la corrección. Deberá ser un único archivo instalable en formato ".tgz" con todos los archivos y directorios empaquetados en un archivo "tar" y luego comprimido con "gzip".

**MUY IMPORTANTE:** cuando se efectúa el "tar" se debe usar la opción de "directorio relativo" para evitar problemas cuando se proceda a la instalación en la facultad para su corrección.

El instalable deberá contener:

- Los scripts desarrollados (según se detallan mas adelante)
- Juego de datos de ejemplo y de prueba lo suficientemente variado como para contemplar todas las variantes posibles. Los archivos de prueba deben respetar las estructuras y directorios indicados en el enunciado ya que es uso y costumbre intercambiar datos entre los distintos grupos para enriquecer los testeos.
- Un archivo README con las instrucciones de instalación y cualquier otra indicación u observación que se quiera incorporar para que el ayudante pueda realizar los testeos
- Cualquier otro archivo que el grupo considere necesario (ej: resultado de las pruebas por Uds. Efectuadas, logs, etc.)

## **Observaciones a tener en cuenta en el desarrollo**

### **1. Código de Retorno:**

1.1. 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. Siempre al finalizar el comando se debe indicar si finalizó correctamente o con errores.

### **2. Pasos Sugeridos:**

2.1. En la descripción de los comandos se sugieren los pasos de ejecución. Estos deben considerarse indicativos, ya que el programador puede alterar este orden según su conveniencia siempre que no afecte el resultado final esperado. Se dan a los efectos de ordenar la explicación. Si Ud. considera conveniente modificar algún paso, tanto sea en el orden de ejecución o en la forma de resolverlo, no hay inconveniente, siempre que lo aclare debidamente en las hipótesis.

### **3. Errores y Mensajes:**

3.1. Cuando se describen los comandos se hace referencia a errores y mensajes informativos, estas referencias son sólo una guía para el desarrollador, pueden existir errores no descriptos en el enunciado que deban ser manejados por el programador.

3.2. Siempre se debe explicar el error con un mensaje amigable.

3.3. Es conveniente para el seguimiento y corrección de los comandos que a medida que se avanza en la ejecución del mismo se vayan mostrando por consola mensajes descriptivos indicando el punto de avance y resultados intermedios.

### **4. Archivos de Log:**

4.1. Todos los comandos deben grabar un archivo de log llamado: <nombre del comando>.log

4.2. En el Log se debe registrar toda la interacción entre el comando y el operador: ej: todos los mensajes que se muestran por la consola del operador (3.3) y los mensajes de los errores que se produzcan (3.2), las opciones o parámetros que ingresa el operador.

4.3. La escritura en el archivo de Log debe ser homogénea para todos los comandos que generan registros de log, por lo tanto debe estar centralizada a través del comando **Glog**.

4.4. Cuando graba una línea de log siempre se debe indicar: fecha y hora, tipo de mensaje y mensaje.

4.5. Existen cuatro tipos de mensajes:

- I = INFORMATIVO
  - En el Archivo de Log se registran eventualmente mensajes informativos sobre el curso de ejecución del comando. Ej: Inicio de Ejecución
- W = WARNING (de alerta)
  - En el Archivo de Log se registran eventualmente mensajes de alerta sobre el curso de ejecución del comando. Ej: Archivo ya procesado
- E = ERROR
  - En el Archivo de Log se registran SIEMPRE mensajes de error Ej: Archivo Inexistente.
- SE = ERROR SEVERO



- En el Archivo de Log se registran SIEMPRE mensajes severos de error Ej: Comando Inexistente.

## **5. Visualizacion de Logs:**

5.1. para una mayor compresión del contenido del archivo de log se debe generar un visualizador llamado **Vlog**.

## **6. Movimiento de Archivos:**

6.1. En líneas generales no se borra ningún archivo, excepto los archivos temporales por Uds, definidos. Debido a esto muchos comandos mueven archivos de un directorio a otro.

6.2. El movimiento de archivos debe ser homogéneo para todos los comandos, por lo tanto debe estar centralizado a través del comando **Mover**



## Especificación de Funciones

### Nombre de la función: **Mover**

#### Input

- Archivo definido en el parámetro 1

#### Output

- Archivo definido en el parámetro 2
- Archivo de Log del comando que la invoca (si corresponde)

#### Opciones y Parámetros

- Parámetro 1 (obligatorio): origen
- Parámetro 2 (obligatorio): destino
- Parámetro 3 (opcional): comando que la invoca
- Si son necesarios mas, especificar por el desarrollador

#### Descripción

Esta función tiene por objeto mover un archivo de un directorio a otro contemplando la posibilidad de archivos duplicados.

Puede ser invocada desde la línea de comando o bien desde otro comando.

En caso de archivos duplicados se debe gestionar un numero de secuencia nnn.

Este número de secuencia puede ser centralizado (una única secuencia para todo el sistema) o descentralizado (diferentes secuencias).

#### Pasos Sugeridos

1. Antes de efectuar el movimiento, debe:
  - 1.1. Verificar si el origen y el destino son iguales, no hacer nada
  - 1.2. Verificar si es un archivo duplicado, es decir, si en el destino ya existe un archivo con ese mismo nombre. En este caso debe:
    - 1.2.1. Ver si existe dentro del directorio destino un subdirectorio /dup
      - 1.2.1.1. si no existe, crearlo
      - 1.2.1.2. Mover el archivo a ese subdirectorio con el siguiente nombre: <nombre del archivo original>.nnn dónde nnn es un número de secuencial que evita nombres duplicados.
    - 1.3. si no es un archivo duplicado, entonces hacer el move al destino indicado.
  2. Si esta función es invocada por un comando que graba en el archivo de log, actualizarlo indicando el resultado de la operación
    - 2.1. Ejemplo: Si el move no tuvo complicaciones Grabar en el Log:  
<fecha y hora>-I-Mover exitoso desde: <parámetro 1> hacia <parámetro 2>
  3. Debe devolver un código de retorno cero (0) si fue exitoso el movimiento o distinto de cero si tuvo errores.



## **Nombre de la función: *Glog***

### Input

- 

### Output

- Archivo de Log

### Opciones y Parámetros

- Parámetro 1 (obligatorio): comando
- Parámetro 2 (opcional): tipo de mensaje
- Parámetro 3 (obligatorio): mensaje
- Si son necesarios mas, especificar por el desarrollador

### Descripción

Esta función tiene por objeto grabar en el log correspondiente los mensajes pasados como parámetro contemplando la creación del archivo de log cuando este no existiera.

Puede ser invocada desde la línea de comando o bien desde otro comando.

El nombre del **Archivo de Log** siempre tiene el siguiente formato: <nombre del comando>.log

El directorio donde se graba el log se indica en la variable de ambiente \$LOGDIR (excepto para el log de la instalación que se graba en un directorio fijo ".../log")

El tamaño máximo que puede alcanzar un archivo de log se indica en la variable de ambiente \$maxlog. Este valor se solicita en la instalación y debe ser empleado para evitar el crecimiento indiscriminado de los archivos de log.

Este tamaño máximo es un valor de referencia, en la práctica es posible encontrar archivos de log que superan el valor \$maxlog dado que depende del momento en que se efectúa este control.

La idea es que cuando un archivo de log supera ese tamaño, se debe truncar el archivo de log dejando por ejemplo las últimas 50 líneas (valor arbitrariamente establecido). Esto debe ser aplicado a todos los comandos excepto al log de la instalación, este log nunca debe ser recortado.

Existen cuatro tipos de mensajes:

- I = INFORMATIVO
- W = WARNING (de alerta)
- E = ERROR
- SE = ERROR SEVERO

Cada línea de log debe contener Fecha y hora-Tipo de Mensaje-Mensaje. Todo en una sola línea.

El separador de registros debe ser new line.



**Nombre de la función: *Vlog***

**Input**

- Archivo de Log

**Output**

- 

**Opciones y Parámetros**

- Parámetro 1 (obligatorio): comando
- Parámetro 2 (deseable): algún tipo de filtro
- Si son necesarios mas, especificar por el desarrollador

**Descripción**

Esta función tiene por objeto generar una visualización amigable del contenido del archivo de log asociado al comando pasado como parámetro.

Debe ser una función que se invoca desde línea.

Es deseable poder filtrar la visualización usando por ejemplo

- Numero de línea
- Fecha
- Filtro
- Algún otro concepto de búsqueda definido por el desarrollador

Debido al alto grado de libertad que se les permite en el desarrollo de este comando, documentarlo detalladamente.





## Especificación de Comandos

### **Comando de Instalación**

Nombre del Comando: **INSTALAR**

#### Input

- Archivos de la instalación

#### Output

- Comandos
- Archivo de Configuración
- Datos de Prueba
- Tablas de Parámetros
- Archivo de Log
- Otros a especificar por el desarrollador

#### Opciones y Parámetros

- A especificar por el desarrollador

#### Descripción

Este es el script para la instalación del paquete. Debe grabar un log exhaustivo de su ejecución.

Cuando este script interactúa con el usuario, el valor por omisión propuesto (o valor default) se debe mostrar entre paréntesis ().

Siempre que este proceso cancele, mostrar el siguiente mensaje:

`"Proceso de instalación cancelado"`

Los usuarios disponibles son solo tres: prueba01 (pw test01), prueba02 (pw test02), prueba03 (pw test03)

Debido a esto cada grupo deberá crear un subdirectorio llamado grupox (donde x es el nro de grupo). Si desean agregar subdirectorios a grupox pueden hacerlo.

Antes de iniciar la ejecución de INSTALAR es necesario copiar desde un medio externo el instalable.

Estos pasos deberán ser explicados dentro del instructivo de instalación (archivo README.txt), a modo de ejemplo podemos enumerar los siguientes pasos:

1. Crear en el directorio corriente un subdirectorio grupox
2. Copiar el archivo \*.tgz en ese directorio
3. Descomprimir el \*.tgz de manera de generar un \*.tar
4. Extraer los archivos del tar.

Estos pasos deben generar AL MENOS la siguiente estructura de directorio:

directorio corriente/grupox/inst con los archivos relativos al paquete de instalación

Todo el camino (path) directorio corriente/grupox lo llamaremos "**\$grupo**" para simplificar el enunciado.



## Pasos sugeridos

### 1. Inicializar archivo de log

Este comando graba un archivo de Log cuyo nombre es instalar.log en el directorio \$grupo/log

En el archivo de log se deben registrar cada uno de los pasos de la instalación con todos los mensajes que se muestran por consola y todas las respuestas dadas por el instalador.

Como resultado de esto se debe obtener una bitácora exhaustiva de la instalación. Cualquiera que desee podrá consultar allí la forma en que el paquete fue instalado.

Grabar en el Log: "Inicio de Ejecución"

### 2. Mostrar (y grabar en el log) donde se graba el log de la instalación

"Log de la instalación: (\$grupo/log/instalar.log)"

### 3. Mostrar (y grabar en el log) el nombre del directorio de instalación

"Nombre del directorio de instalación: (\$grupo/inst)"

### 4. Mostrar (y grabar en el log) el nombre del directorio de parámetros de configuración

"Nombre del directorio de configuración: (\$grupo/conf)"

### 5. Detectar si alguno de los componentes del paquete ya está instalado

Este script de instalación se debe preparar de manera tal que detecte si el práctico ya se encuentra instalado. Una forma de hacer esto es verificando la existencia del archivo \$grupo/conf/practico.conf (que se genera a partir de la instalación como verán mas adelante)

- Si \$grupo/conf/practico.conf no existe, asumir que el paquete no fue instalado. El proceso sigue en el punto 6.
- Si \$grupo/conf/practico.conf existe, asumir que el paquete ya fue instalado pero se debe verificar que componentes están instalados. Para ello se puede leer el archivo de configuración \$grupo/conf/practico.conf y por cada comando que figure allí verificar en el directorio ejecutable correspondiente si realmente existe.
  - Mostrar los componentes instalados indicando fecha de instalación y usuario de instalación.  
Ej:  
Comando INICIAR instalado el <fecins> por <usins>
  - Si alguno de sus componentes no esta instalado:
    - indicar cuales faltan
    - indicar claramente que debe hacer el usuario para completar la instalación
  - Si todos los componentes del practico están instalados, mostrar un mensaje explicativo y terminar la instalación

### 6. Aceptación de términos y condiciones

Mostrar en consola el siguiente mensaje:

```
*****
*      Proceso de Instalación de "Practico"      *
*      Copyright TPSistemasOp (c)2009            *
*****
A T E N C I O N: Al instalar Practico UD. expresa aceptar los términos y
Condiciones del "ACUERDO DE LICENCIA DE SOFTWARE" incluido en este paquete.
```

Acepta? Si -No

Si el usuario no acepta, cancelar el proceso



Si el usuario acepta, continuar  
Cualquier otra cosa, seguir en este punto

### **7. Chequear que Perl esté instalado**

Para la instalación de Practico es necesario chequear que Perl versión 5 o superior este instalado.  
Si está instalado mostrar (y grabar en el log) un mensaje con la versión de perl detectada.  
Almacenar este valor en la variable PERLVER

ACLARACION: Toda vez que se solicita "Almacenar tal valor en tal variable" es para facilitar la explicación de los pasos subsiguientes.

Si al chequear esto da error, entonces mostrar (y grabar en el log) el siguiente mensaje:

"ATENCION! Esta instalación solo puede ser ejecutada si Perl 5 o superior esta instalado.  
Efectúe la instalación de Perl y reinicie el proceso de instalación."

Cancelar el proceso.

### **8. Definir el directorio de ejecutables**

Se debe solicitar al usuario el directorio donde desea dejar los ejecutables, para ello mostrar en pantalla el siguiente mensaje:

"Nombre del directorio de ejecutables: (\$grupo/bin)"

Si el usuario acepta el nombre propuesto, continuar

Si el usuario lo desea cambiar, debe ingresar un nombre de directorio (p.e. "ejecutables")

Almacenar este valor en la variable BINDIR.

### **9. Definir el directorio de datos**

Se debe solicitar al usuario el directorio donde desea recibir los archivos de input del práctico, para ello mostrar en pantalla el siguiente mensaje:

"Nombre del directorio de datos: (\$grupo/datos)"

Si el usuario acepta el nombre propuesto, continuar

Si el usuario lo desea cambiar, debe ingresar un nombre de directorio

Almacenar el nombre en la variable DATADIR

### **10. Mostrar (y grabar en el log) el espacio disponible**

"Espacio disponible en \$grupo/\$DATADIR: <cantidad> <unidad de medida>"

Almacenar <cantidad> en la variable DATASIZE.

Almacenar <unidad de medida> en la variable USIZE.

"Desea continuar? Si -No

Si el usuario no acepta, cancelar el proceso

Si el usuario acepta, continuar

Cualquier otra cosa, seguir en este punto

### **11. Definir el directorio de logs para los comandos**

Se debe solicitar al usuario el directorio donde desea dejar los archivos de log del práctico, para ello mostrar en pantalla el siguiente mensaje:

"Nombre del directorio de log: (\$grupo/log)"

Si el usuario acepta el nombre propuesto, continuar

Si el usuario lo desea cambiar, debe ingresar un nombre de directorio

Almacenar el nombre en la variable LOGDIR.

### **12. Definir el espacio máximo para logs**



Calcular \$MAXLOG = 1% del DATASIZE

Se debe solicitar al usuario el espacio máximo que puede ocupar el log de los comandos del práctico, para ello mostrar en pantalla el siguiente mensaje:

"Espacio máximo para archivos de log? (\$MAXLOG \$USIZE)

Si el usuario ingresa un valor menor al 1% del DATASIZE, aceptar ese valor y continuar

Si el usuario ingresa un valor mayor al 1% del DATASIZE, rechazar ese valor y Mostrar un mensaje indicando cual es el máximo permitido para los archivos de log (MAXLOG=DATASIZE/100) y volver a preguntar.

Almacenar este valor en la variable MAXLOG

### 13. Mostrar la estructura de directorios resultante

Limpiar la pantalla y Mostrar en consola el siguiente mensaje:

```
"PARÁMETROS DE INSTALACION DE PRACTICO"
"Versión de Perl:  $PERLVER"
"Nombre del directorio de instalación:  $grupo/inst"
"Nombre del directorio de configuración: $grupo/conf"
"Nombre del directorio de ejecutables:  $grupo/$BINDIR"
"Nombre del directorio de datos:        $grupo/$DATADIR"
"Nombre del directorio de logs:         $grupo/$LOGDIR"
"Espacio máximo para archivos de log:   $MAXLOG $USIZE"
```

### 14. Confirmar Inicio de Instalación

Mostrar en consola el siguiente mensaje:

Continuar instalación? Si - no

Si el usuario acepta, continuar

Si el usuario no acepta limpiar la pantalla y volver a empezar por ejemplo a partir del punto "Chequear que Perl esté instalado"

En este caso, cuando se reinicia la instalación, los valores default propuestos deben ser los previamente ingresados por el usuario, es decir, los contenidos en las variables: BINDIR, DATADIR, DATASIZE, LOGDIR, MAXLOG, etc

Cualquier otra cosa, seguir en este punto

### 15. Crear la estructura de Directorio definida.

Crear la estructura correspondiente y Mostrar en consola el siguiente mensaje:

```
/$BINDIR
/$DATADIR
/$LOGDIR
```

"Creando Estructura de Directorio....."

### 16. Mover archivos

Mostrar en consola el siguiente mensaje:

"Moviendo Archivos ....."

Mover las tablas de configuración a CONFDIR

Mover las funciones de apoyo y los comandos a BINDIR

Mover todo otro archivo donde se considere apropiado. A medida que se avanza en los move, ir mostrando los mensajes correspondientes.

No es necesario usar el comando mover para hacer esta operación, pero si el diseñador contempla esta posibilidad adaptar el comando move para estos fines e Indicar en las hipótesis las adaptaciones al diseño.



### 17. Guardar la información de Instalación

Almacenar la siguiente información en el archivo \$grupo/conf/practico.conf en el directorio CONFDIR.

Línea 01: INSTDIR=/inst  
Línea 02: FILECONF=/conf/practico.conf  
Línea 03: LOGFILE=/log/instalar.log  
Línea 04: DATADIR=\$DATADIR  
Línea 05: PERLVER=\$PERLVER  
Línea 06: MAXLOG=\$MAXLOG  
Línea 07: USIZE=\$USIZE  
Línea 08:  
Línea 09:

(Líneas 8 y 9 reservadas para futuro uso)

A partir de la línea 10, en bloques de 5 líneas, se deben grabar los datos de cada comando principal (Ej DETECTAR, INICIAR, INTERPRETE, etc).

Se debe grabar: nombre del comando, directorio de instalación, usuario de instalación, fecha completa de instalación, directorio de log para este comando y una línea en blanco

Ejemplo: para iniciar

Línea 10: COMAND = INICIAR  
Línea 11: BINDIR = \$BINDIR  
Línea 12: USERID = Usuario de la instalación  
Línea 13: FECINS = fecha de instalación  
Línea 14: LOGDIR = \$LOGDIR  
Línea 15:

Línea 40 y siguientes: disponibles para su uso por parte del grupox

### 18. Borrar archivos temporarios, si los hubiese generado

### 19. Mostrar mensajes indicando que fue lo que se instaló.

### 20. Cerrar el archivo de log

Grabar en el archivo de log: "Fin de Ejecución"

Cerrar el archivo de Log



## **Comando de Inicialización de Ambiente**

Nombre del Comando: **INICIAR**

Input

- A especificar por el desarrollador

Output

- A especificar por el desarrollador

Opciones y Parámetros

- A especificar por el desarrollador

Descripción

Este comando debe ser el primero en orden de ejecución. Se dispara manualmente.

El propósito de este comando es preparar el entorno de ejecución del TP (ambiente). Debe efectuar el seteo inicial de las variables de ambiente y debe ofrecer arrancar automáticamente al comando DETECTAR siempre que DETECTAR no se esté ejecutando.

Pasos Sugeridos:

### **1. Inicializar el Log**

### **2. Ver si INICIAR ya fue ejecutado**

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

Si el ambiente ya fue inicializado, entonces INICIAR solo mostrar las variables de ambiente y sus valores y el process id de DETECTAR si aun sigue corriendo.

### **3. Ver si PRACTICO esta completo**

INICIAR debe verificar si la instalación está completa, si detecta algún problema en la instalación, debe explicar la situación y terminar su ejecución

Este control debe incluir la verificación de tablas indispensables para ejecutar el sistema.

### **4. Solicitar variables de configuración de DETECTAR**

Se debe solicitar al usuario la cantidad de ciclos de DETECTAR (variable CANLOOP) y el tiempo de espera entre cada ciclo de DETECTAR (variable TESPERA), para ello mostrar en pantalla el siguiente mensaje:

"Cantidad de Ciclos de DETECTAR? (100 ciclos)"

Almacenar este valor en CANLOOP

Validar que CANLOOP sea siempre positivo

"Tiempo de espera entre ciclos? (1 minuto)"

Almacenar este valor en TESPERA

Validar que TESPERA sea siempre mayor o igual a un minuto

Si el usuario ingresa un valor incorrecto volver a preguntar.

### **5. Setear Variables de ambiente**

Setear las variables de ambiente siempre y cuando sea la primera vez que se ejecuta INICIAR en esta sesión de usuario.

Dentro del seteo de variables se incluyen:



- La variable PATH, para permitir la correcta ejecución de los scripts por parte de usuarios que no son root.
- La variable \$grupo para evitar el hardcode de directorios
- Las variables TESPORA y CANLOOP
- Otras variables que se consideren necesarias: MAXLOG – BINDIR – LOGDIR –, etc

## **6. Ver si arranca DETECTAR**

INICIAR debe ofrecer la posibilidad de arrancar el demonio Detectar en forma automática. Está en condiciones de arrancar siempre que:

- Se encuentren instalados todos los componentes del sistema
- Detectar no se esté ejecutando
- 

Si está bajo estas condiciones, Ej: Mostrar el siguiente mensaje:

“Desea efectuar la activación de DETECTAR?” Si – No

Si arranca el demonio, mostrar mensaje indicando que esta activado y bajo que numero de proceso.  
Demonio corriendo bajo el no.: <Process Id de GEMONI>

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

Ej: Mostrar el siguiente mensaje:

“Para realizar la activación del comando DETECTAR debe ... <describir como invocar el comando>”

## **7. Cerrar el Log**



## **Comando de Detección de Archivos**

Nombre del Comando: **DETECTAR**

### Input

- Archivos (cualesquiera) que arriban al directorio \$DATADIR
- Tabla de Países y Sistemas \$grupo/conf/**p-s.tab**

### Output

- Archivos de Practico Recibidos \$grupo/datadir/**ok**/**<pais>-<система>-<año>-<mes>**
- Archivos (cualesquiera) Rechazados \$grupo/datadir/**nok**/**<nombre del archivo>**
- Log \$LOGDIR/detectar.log

### Opciones y Parámetros

- A especificar por el desarrollador

### Descripción

Este comando puede ser disparado manualmente o a través del comando INICIAR.

No se puede ejecutar este comando si la inicialización de ambiente no fue realizada.

El propósito de este comando es recibir los archivos que arriban al directorio DATADIR. Si los archivos son validos, entonces ponerlos a disposición del siguiente paso. Si los archivos no son válidos, rechazarlos.

Para determinar si un archivo es válido o no se debe verificar su nombre.

Los archivos válidos poseen nombres con este formato: **<pais>-<система>-<año>-<mes>**

- Para verificar el país usar la tabla p-s.tab
- Para verificar el sistema usar la tabla p-s.tab
- Para verificar el año usar los valores 2000 hasta el año corriente
- Para verificar el mes usar los valores 1 a 12. Para el año corriente, verificar que el mes sea menor o igual al mes corriente.

Es decir que al directorio datadir pueden arribar archivos provenientes de varios paises, de varios sistemas y de diferentes períodos. Ejemplo

A-6-2009-6

B-7-2009-8

A-6-2008-12

A-7-2009-6

Este comando es un proceso del tipo "Demonio"

- Un demonio es un proceso que corre en forma permanente, en background.
- Suele armarse como un conjunto de sentencias dentro de un ciclo eterno.
- Se utiliza generalmente para:
  - monitorear otros procesos
  - ejecutar procesos rutinarios

### Pasos sugeridos:

1. Grabar en el Log el nro de ciclo: Ej: "Ciclo Nro 1"





2. Chequear la existencia de archivos en el directorio \$grupo/datadir
3. Si existen archivos, por cada archivo que se detecta
  - 3.1. Verificar si el nombre del archivo es valido
    - 3.1.1. Si el archivo es válido mover el archivo a \$grupo/datadir/ok y grabar en el log "RECIBIDO: <nombre del archivo>"
    - 3.1.2. Si el archivo no es válido mover el archivo a \$grupo/datadir/nok y grabar en el archivo de log "RECHAZADO: <nombre del archivo>"
4. Chequear la existencia de archivos en el directorio \$grupo/datadir/ok (ya sean del ciclo actual o de ciclos anteriores)
5. Si existen archivos en \$grupo/datadir/ok
  - 5.1. Invocar al Comando INTERPRETAR

Se debe invocar al comando **INTERPRETE** siempre que se haya algún archivo en \$grupo/datadir/ok e **INTERPRETE** no se esté ejecutando.

Si arranca correctamente se debe mostrar por pantalla el process id de interpretar

Si da algún tipo de error se debe mostrar por pantalla el mensaje explicativo
6. Si DETECTAR aún no alcanzó la cantidad de ciclos definida en la variable de ambiente en CANLOOP, entonces
  - 6.1. Dormir un tiempo TESPORA (variable de ambiente)
  - 6.2. Volver al punto 1
7. Si se alcanzó la cantidad de ciclos definida en la variable de ambiente CANLOOP, terminar DETECTAR.

### p-s.tab

Se adelanta la estructura de la tabla p-s.tab dado que se emplea en este comando para validar el nombre de los archivos.

En detalle encontramos: país, nombre del país, sistema, nombre del sistema, nombre del archivo maestro de transacciones. El separador de campo es el guion.

Nombre campo	Descripción	Formato
PAIS_ID	Código de país	Char(3)
PAIS_DESC	Nombre del Pais	varchar
SIS_ID	Código de sistema	Number(2)
SIS_DESC	Nombre del Sistema	varchar
MAE_ID	Código de Archivo Maestro	Char(2)
MAE_DESC	Nombre del Archivo Maestro	varchar



## **Comando de Interpretación de Archivos**

Nombre del Comando: **INTERPRETE**

### Input

- Archivos de Practico Recibidos \$grupo/datadir/**ok**/**<pais>-<sistema>-<año>-<mes>**
- Tabla de Separadores \$grupo/conf/**T1.tab**
- Tabla de Campos \$grupo/conf/**T2.tab**

### Output

- Archivo de Contratos de Préstamos Personales \$grupo/datadir/**new/CONTRAT.<pais>**
- Archivos (duplicados) Rechazados \$grupo/datadir/**nok**/**<nombre del archivo>**
- Archivos de Practico Procesados \$grupo/datadir/**old**/**<pais>-<sistema>-<año>-<mes>**
- Log \$grupo/logdir/**interprete.log**

### Opciones y Parámetros

- A especificar por el desarrollador

### Descripción

Este comando puede ser disparado manualmente o a través del comando DETECTAR.

No se puede ejecutar este comando si la inicialización de ambiente no fue realizada.

El propósito de este comando es leer los archivos que se encuentran en el directorio \$grupo/datadir/ok y grabar sus registros formateados en los archivos que se encuentran en \$grupo/datadir/new

Los archivos del directorio \$grupo/datadir/ok llamados <pais>.<sistema>.<año>.<mes> contienen información de contratos de préstamos personales impagos generada por los sistemas legacy que operan en distintos países. Cada sistema reúne la información por periodo (año-mes) y la remite a nuestro sistema.

El comando previo a éste (el comando DETECTAR) filtró los archivos por nombres y dejó en el directorio \$grupo/datadir/ok sólo los archivos correspondientes a este proceso.

La característica principal de estos archivos es que debido a su mas variado origen, no poseen una estructura estándar.

El objetivo es desarrollar un intérprete para que la lectura de los archivos de input sea dinámica y se adapte a cualquier archivo recibido independientemente del país o el sistema de origen.

A partir de la interpretación de esta información se debe construir el registro de salida formateando los campos para llevarlos al formato estándar.

El formato estándar (alineado al formato del archivo maestro) es:

Separador Decimal: siempre coma

Fecha: dd/mm/yyyy

Longitud numérica entera: hasta 10 posiciones enteras

Longitud numérica decimal: hasta 2 posiciones decimales

Una vez obtenido el registro grabarlo si corresponde en el archivo de salida según sea el país al que pertenece llamado CONTRAT.<pais> (este Archivo de Contrato esta centralizado por país)

### Pasos sugeridos

#### **1. Ver si existe otro INTERPRETE corriendo**

Este INTERPRETE puede ejecutar siempre que no exista otro en ejecución.

Si se detecta otro en ejecución mostrar un mensaje explicativo y terminar la ejecución.



## 2. Ver si se inicializó el ambiente

Este INTERPRETE puede ejecutar siempre que el ambiente haya sido inicializado.

Si detecta algún problema mostrar un mensaje explicativo y terminar la ejecución.

## 3. Inicializar el Log

Inicializar el log grabando: Inicio de Intérprete y Cantidad de Archivos.

## 4. Por cada archivo de input

Los archivos de input se encuentran en \$grupo/datadir/**ok**

### 4.1 Verificar que no sea un archivo duplicado

Como los archivos ya interpretados se movieron oportunamente (y tal cual fueron recibidos) a \$grupo/datadir/**old**, es posible detectar tempranamente si el archivo vino duplicado analizando este directorio.

Si se detecta que el archivo vino duplicado, mover el archivo a \$grupo/datadir/nok y grabar en el archivo de log "DUPLICADO: <nombre del archivo>"

### 4.2 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 cual es el separador de campo y cual es el separador decimal que usa el archivo de input según el país y el sistema de origen.

En detalle en esta tabla encontramos: país, sistema, carácter separador de campo, carácter separador decimal.

### 4.3 Determinar Campos

Se debe recuperar de la tabla T2 la estructura del registro de input y el formato de los campos de interés (aquellos cuyo flag de utilización es 1) según el país y el sistema de origen.

La tabla de Campos o Tabla T2 nos dice que campos y en que orden de aparición vamos a encontrar en el Archivo de Practico Recibido.

En detalle en esta tabla encontramos: país, sistema, nombre del campo, orden de aparición, formato del campo, flag de utilización.

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

- El formato de las fechas puede ser: ddmmyy<longitud>. O yymmdd<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 desconecemos. 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.



## **5. Por cada Registro del Archivo a Interpretar**

### **5.1 Leer el registro**

Se debe leer el registro de input interpretando la estructura de acuerdo a los separadores obtenidos en 4.2 y los campos determinados en 4.3

### **5.2 Construir el registro de contrato**

#### Formateo de Fechas

El formato de las fechas debe ser llevado al formato estándar.

#### Formateo del Estado Contable

Los contratos con estado contable CTX se descartan.

Los estados contables 'SNA' y 'SNIM' deben ser reasignados como 'Sanos'.

Los estados contables 'DTCA' y 'DTXA' deben ser reasignados como 'Dudosos'.

(no existen "otros" estados contables que no sean 'CTX' 'SNA' 'SNIM' 'DTCA' y 'DTXA')

#### Formateo del Montos

Todos los campos de montos deben ser informados a cero si no están informados.

El formato de los montos debe ser llevado al formato estándar.

#### Cálculo de monto restante

El campo de monto restante debe ser calculado como la suma de los montos CRD, IMPAGO, INDE y deducir las otras sumas del cliente.

$$MT\_RESTANTE = MT\_CRD + MT\_IMPAGO + MT\_INDE - MT\_OTRSUMDC$$

Los contratos con saldo restante menor a cero se descartan.

### **5.3 Grabar, si corresponde, el registro de salida**

Grabar el contrato en el archivo \$grupo/datadir/**new/CONTRACT.<pais>** correspondiente

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

Repetir el paso 5 hasta que se termine el archivo.

## **6. Fin de Archivo**

Mover el archivo procesado a: \$grupo/datadir/**old**

Grabar en el Log la cantidad de Registros de Input y la cantidad de Registros de Output

Repetir desde el paso 4 hasta que se terminen todos los archivos.

## **7. Cerrar el Log**

Cerrar el log grabando "Fin de Intérprete" .



## **Comando de Impresión**

Nombre del Comando: **REPORTE**

### Input

- Archivo de Contratos de Préstamos Personales \$grupo/datadir/new/CONTRAT.<pais>
- Archivo Maestro Préstamos Personales Impagos \$grupo/datadir/mae/PPI.mae
- Tabla de Países y Sistemas \$grupo/conf/p-s.tab

### Output

- Log \$grupo/logdir/reporte.log
- Listados \$grupo/datadir/list/<nombre listado>.<id>
- Archivo de Modificaciones de Contratos \$grupo/datadir/new/MODIF.<pais>

### Descripción

Este comando se dispara manualmente.

No se puede ejecutar este comando si la inicialización de ambiente no fue realizada.

Si al invocarlo ya existe otro comando en ejecución, terminar la ejecución del presente comando.

El propósito de este comando es resolver, mostrar y, eventualmente, grabar consultas vinculadas a la comparación entre dos fuentes: El archivo de contratos (CONTRACT.pais) y el archivo maestro (PPI.mae)

Ambos archivos contienen información de contratos de préstamos personales impagos, una fuente (CONTRACT.pais) es generada por los sistemas legacy que operan en distintos países. La otra fuente (PPI.mae) es generada por el Sistema Contable Central.

El objetivo es desarrollar un comando de consultas para que el usuario pueda controlar y decidir si genera los archivos de modificaciones de contratos que luego cada sistema legacy deberá usar para actualizar sus registros (esta actualización de sistemas legacy no forma parte del alcance del presente trabajo)

Para efectuar la comparación siempre se deberá cargar en memoria el archivo maestro empleando estructura hash (requisito indispensable) en donde el valor debe ser el Monto Restante.

#### Cálculo de monto restante del maestro

El campo de monto restante debe ser calculado como la suma de los montos CRD, IMPAGO, INDE y deducir las otras sumas del cliente.

$$MT\_RESTANTE = MT\_CRD + MT\_IMPAGO + MT\_INDE - MT\_OTRSUMDC$$

La solución debe:

- Presentar un menú amigable y permitir al usuario efectuar N consultas
- Solicitar al usuario
  - Parámetros
    - País (obligatorio)
    - Sistema (opcional)
      - Si el usuario ingresa un sistema que no existe, indicarlo y volver a preguntar
    - Año (opcional)
    - Mes (opcional)
      - Si el usuario no ingresa año, no corresponde ingresar mes
  - Opciones
    - Si desea o no grabar los listados
    - Si desea o no grabar los archivos de modificaciones de contratos
- Resolver las consultas seleccionadas
- Mostrar los resultados por pantalla indicando en el encabezado los datos de la consulta y subtotales y totales por año y sistema
- Grabar, si corresponde, los listados con lo mismo que se muestra por pantalla



- Grabar, si corresponde, las modificaciones con lo mismo que viene en el Maestro
  - Esta posibilidad aplica solo para los registros que en el maestro tienen estado SANO.
  - No aplica para los Contratos Comunes Sanos con Idéntico Monto Restante.

Las consultas/listados posibles son los que describen con el ejemplo.

Datos de Ejemplo	País	Sistema	Año	Mes	Numero de Contrato	Estado	Monto Restante
Archivo Contrato	A	6	2009	3	100	SANO	100
					101	SANO	101
					110	SANO	110
					111	SANO	111
					120	SANO	120
					121	SANO	121
					130	SANO	130
					131	SANO	131
					140	DUDOSO	140
					150	DUDOSO	150
					160	DUDOSO	160
					170	DUDOSO	170
Archivo Maestro	A	6	2009	3	100	SANO	100
					101	SANO	101
					110	SANO	100
					111	SANO	100
					120	DUDOSO	120
					121	DUDOSO	121
					130	DUDOSO	100
					131	DUDOSO	100
					140	DUDOSO	140
					150	DUDOSO	100
					160	SANO	160
					170	SANO	100



Consultas	Listado Ejemplo					
	Parámetros	Cantidad de Contratos	Estado Contrato	Estado Maestro	Monto Restante Contrato	Monto Restante Maestro
a = Contratos Comunes Sanos con Idéntico Monto Restante	A 6 2009 3	2	SANO	SANO	201	201
b = Contratos Comunes Dudosos con Idéntico Monto Restante	A 6 2009 3	1	DUDOSO	DUDOSO	140	140
c = Contratos Comunes Sanos con Diferente Monto Restante	A 6 2009 3	2	SANO	SANO	221	200
d = Contratos Comunes Dudosos con Diferente Monto Restante	A 6 2009 3	1	DUDOSO	DUDOSO	150	100
e = Contratos Comunes con Diferente Estado e Idéntico Monto Restante	A 6 2009 3	2	SANO	DUDOSO	241	241
	A 6 2009 3	1	DUDOSO	SANO	160	160
f = Contratos Comunes con Diferente Estado y Diferente Monto Restante	A 6 2009 3	2	SANO	DUDOSO	261	200
	A 6 2009 3	1	DUDOSO	SANO	170	100

Ejemplo para el Archivo de Modificaciones						
La estructura se detalla mas abajo						
Pais	Sistema	Año	Mes	Numero de Contrato	Estado	Monto Restante
A	6	2009	3	110	SANO	100
A	6	2009	3	111	SANO	100
A	6	2009	3	160	SANO	160
A	6	2009	3	170	SANO	100



## Estructuras de Archivos

### Configuración \$grupo/conf/practico.conf

Nombre campo	Descripción
CD_LIN	Línea

### Log \$grupo/logdir/<nombre del comando>.log

Nombre campo	Descripción
DT_REGIS	Fecha y Hora de Registro, formato a establecer por el desarrollador
CD_TIP_MEN	Tipo de Mensaje
CD_MEN	Mensaje
Separador de campos – (guión)	

### Países y Sistemas \$grupo/conf/p-s.tab

Nombre campo	Descripción	Formato
PAIS_ID	Código de país	Char(3)
PAIS_DESC	Nombre del País	varchar
SIS_ID	Código de sistema	Number(2)
SIS_DESC	Nombre del Sistema	varchar
MAE_ID	Código de Archivo Maestro	Char(2)
MAE_DESC	Nombre del Archivo Maestro	varchar
Separador de campos – (guión)		

### Separadores \$grupo/conf/T1.tab

Nombre campo	Descripción	Formato
PAIS_ID	Código de país	Char(3)
SIS_ID	Código de sistema	Number(2)
SEP_CAMP	Separador de campo	Char(1)
SEP_DEC	Separador de decimales	Char(1)
Separador de campos – (guión)		

### Campos \$grupo/conf/T2.tab

Nombre campo	Descripción	Formato
PAIS_ID	Código de país	Char(3)
SIS_ID	Código de sistema	Number(2)
NOM_CAMP	Nombre del campo	VarChar(15)
ORD_CAMP	Orden de aparición del campo	Number(2)
FMT_CAMP	Formato del campo	VarChar(15)
FLAG_CAMP	Flag de utilización del campo	Number(1)
Separador de campos – (guión)		





**Archivo de Contratos de Préstamos Personales**  
**\$grupo/datadir/new/CONTRAT.<pais>**

Nombre campo	Descripción
SIS_ID	Código de sistema
AÑO_ID	Año de proceso
MES_ID	Mes de proceso
NO_CONTRAT	Numero de Contrato
DT_FLUX	Fecha de Contrato
CD_STATCTB	Estado Contable
MT_CRD	Monto del Crédito
MT_IMPAGO	Monto Impago
MT_INDE	Monto del Interés Devengado
MT_INNODE	Monto de Interés No Devengado
MT_OTRUMDC	Otras Sumas del Cliente
MT_RESTANTE	Monto Restante
DT_INSERT	Fecha de Grabación
US_ID	Usuario de Grabación
Separador de Campos: - Guión	

Generar archivos de prueba para: Argentina y Chile sistemas 6 y 7, años 2008 y 2009, meses todos.

**Archivo Maestro Contable de Préstamos Personales Impagos**  
**\$grupo/datadir/mae/PPI.mae**

Nombre campo	Descripción
PAIS_ID	Código de Pais
SIS_ID	Código de Sistema
AÑO_ID	Año de proceso
MES_ID	Mes de proceso
CD_STATORI	Estado Original
CD_COD_PROD	Código de Producto
CD_STATCTB	Estado Contable
NO_CONTRAT	Numero de Contrato
DT_CONTRAT	Fecha de Contrato
CD_CAT_CLI	Categoría del Cliente
MT_CRD	Monto del Crédito
MT_IMPAGO	Monto Impago
MT_INNODE	Monto Interés Devengado
MT_INDE	Monto Interés no Devengado
MT_OTRUMDC	Otras Sumas del Cliente

Separador de Campos: - Guión



**Archivo de Modificaciones de Contratos de Préstamos Personales**  
**\$grupo/datadir/new/MODIF.<pais>**

Nombre campo	Descripción	Fuente
SIS_ID	Código de sistema	Archivo Maestro
AÑO_ID	Año de proceso	Archivo Maestro
MES_ID	Mes de proceso	Archivo Maestro
NO_CONTRAT	Numero de Contrato	Archivo Maestro
DT_FLUX	Fecha de Contrato	Archivo Maestro
CD_STATCTB	Estado Contable	Archivo Maestro
MT_CRD	Monto del Crédito	Archivo Maestro
MT_IMPAGO	Monto Impago	Archivo Maestro
MT_INDE	Monto del Interés Devengado	Archivo Maestro
MT_INNODE	Monto de Interés No Devengado	Archivo Maestro
MT_OTRUMDC	Otras Sumas del Cliente	Archivo Maestro
MT_RESTANTE	Monto Restante	Cálculo correspondiente
DT_INSERT	Fecha de Grabación	Fecha de Grabación
US_ID	Usuario de Grabación	Usuario de Grabación

Separador de Campos: - Guión

**Listados**

\$grupo/datadir/list/listA.<id>  
\$grupo/datadir/list/listB.<id>  
\$grupo/datadir/list/listC.<id>  
\$grupo/datadir/list/listD.<id>  
\$grupo/datadir/list/listE.<id>  
\$grupo/datadir/list/listF.<id>

Nombre campo	Descripción
CD_LIN	Línea

**Datos**

Los Datos de tablas y maestro se encuentran en DatosPracticoTemaJ.xls