



### CARÁTULA

Tema	Grupo	Ayudante	Correcciones			
			Fecha	Hora Inicio	Hora Fin	Resultado
			Entrega ____ / ____			
			Revisión ____ / ____			

### INTEGRANTES

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

### PLANILLA DE EVALUACIÓN – Evaluación mínima

ITEM	SI	NO	Responsable	Observaciones
CARPETA				
Carátula completa? Dos?				
Hojas sueltas?				
Índice, Pie y nro. de hoja??				
Hipótesis y Aclaraciones Glob?				
Problemas Relevantes?				
README				
Instrucciones de instalación?				
Instrucciones de ejecución?				
DISPOSITIVO - PAQUETE				
Completo y con última versión?				
Genera /conf				
Genera /datos				
Arranca el Installer?				
Logra realizar la instalación?				
Graba .log?				
Graba .conf correctamente?				
Arranca el Initializer?				
Setea variables?				
Dispara el Listener?				
Graba log?				
Hay hoja de ruta camino feliz?				



Identifica archivos útiles para la prueba? Impresos?				
Listener				
Contabiliza los ciclos?				
Acepta listas de precios?				
Dispara Masterlist?				
Acepta listas de compras?				
Dispara Rating?				
Graba Log?				
Masterlist				
Graba bien 1° Lista Maestra?				
Graba bien 2° vez (ALTA)?				
Graba bien 3° vez (reemplazo)?				
Graba Log?				
Mueve los procesados?				
Rating				
Procesa bien 0 coincidencia?				
Procesa bien 1 coincidencia?				
Procesa bien n coincidencias?				
Graba bien presupuestadas?				
Graba Log?				
Mueve los procesados?				
Arranca Reporting?				
Opciones f p m d				
Algún Filtro				
Graba/Muestra bien 1° Informe?				
Graba/Muestra bien 2° Informe?				



---

## Índice

<b>Enunciado – Tema D .....</b>	<b>4</b>
Introducción .....	4
<b>Estructuras y Ejemplos .....</b>	<b>9</b>
Archivos del Sistema .....	9
Archivos Maestros y Tablas .....	10
Archivos de Novedades.....	12
Archivos de Retail .....	13
<b>Especificación de Comandos y Funciones .....</b>	<b>14</b>
Arranque: Start .....	14
Detención: Stop .....	14
Movimiento de Archivos: Mover .....	15
Logueo: Logging.....	16
Instalación: Installer.....	18
Inicialización: Initializer .....	25
Recepción de novedades: Listener .....	27
Actualización de Lista Maestra: Masterlist.....	30
Cálculo de Presupuesto: Rating.....	33
Impresión de Informes: Reporting .....	36



## Enunciado – Tema D

### Introducción

Una asociación de consumidores ofrece a sus asociados un servicio para presupuestar su lista de compras.

Para ello usan el sistema **RETAILD** que les permite crear una lista maestra de precios a partir de las listas de precios de supermercados y grandes tiendas que luego se emplea para presupuestar las listas de compras.

### Instalación del Sistema

El sistema desarrollado debe tener un comando para su instalación.

Esta actividad debe llevarse a cabo con el comando **Installer**

### Inicialización del entorno de ejecución del Sistema

El Proceso se inicia con el aseguramiento de la disponibilidad de la información para llevar adelante el proceso total.

Continúa con la asignación de valor a un conjunto de variables de ambiente y luego ofrece arrancar automáticamente el comando de recepción de novedades (novedades de precios o listas de compras).

Todas estas actividades deben llevarse a cabo con el comando **Initializer**.

### Recepción de novedades

Luego tenemos el proceso de recepción de novedades.

Pueden llegar dos tipos novedades: listas de compras que envían los asociados, listas de precios que envían los colaboradores de la Asociación.

Si el nombre del archivo (filename) cumple con algún patrón esperado, se acepta el archivo, de lo contrario, se lo rechaza.

Cuando recibe una novedad de precios, dispara la actualización de la lista maestra de precios.

Cuando verifica que hay listas de compras, intenta disparar el cálculo del presupuesto.

Todas estas actividades deben llevarse a cabo con el comando **Listener**.

### Actualización de Lista Maestra de Precios

Los colaboradores de la Asociación, usando técnicas de scrapping, obtienen listas de precios de distintas fuentes. Las listas de precios vienen en los más diversos formatos, es por ello que este comando reúne la información en un formato unificado y lo deja disponible para el cálculo del presupuesto.

Estas actividades deben llevarse a cabo con el comando **Masterlist**

### Cálculo de Presupuesto

Las listas de compras deben ser presupuestadas usando la lista maestra de precios.

Estas listas presupuestadas se graban en una nueva estructura.

Estas actividades deben llevarse a cabo con el comando **Rating**.

### Obtención de Informes

Por último tenemos un proceso que nos permite realizar consultas sobre los presupuestos y emitir informes de recomendación (donde comprar, cuando comprar, etc.).

Todas estas actividades deben llevarse a cabo con un comando PERL denominado **Reporting**.



### Desarrollar el sistema **RETAILD** con todos sus componentes

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

También deben elaborar un comando para instalar todo y varias funciones que centralizan utilidades compartidas por varios comandos.

Los elementos a desarrollar son:

- Shell script denominado **Initializer** que efectúa la inicialización de ambiente para la correcta ejecución del TP
- Shell script denominado **Listener** que detecta las novedades
- Shell script denominado **Masterlist** que crea la lista maestra de precios
- Shell script denominado **Rating** que calcula el presupuesto.
- Programa Perl denominado **Reporting** que genera los informes de recomendación.
- Función (en Shell o en Perl) denominada **Mover** que se emplea para mover archivos
- Función (en Shell o en Perl) denominada **Logging** que se emplea para grabar los archivos de log
- Función en Shell script denominada **Stop** que se emplea para detener procesos y otra complementaria **Start** que permite disparar procesos
- Shell script denominado **Installer** que permite la instalación de todo lo anterior

### Elaborar la documentación del sistema **RETAILD**

Se requiere elaborar la documentación del sistema y armar una carpeta para entregar el día de la corrección al ayudante designado con los siguientes elementos:

#### 1. Carátula

La entregada en este mismo documento con los datos completos en 2 COPIAS una para el grupo y otra para el docente.

#### 2. Planillas de Evaluación

Las entregadas en este mismo documento

#### 3. Índice del Contenido de la Carpeta.

El número de página puede ser incorporado manualmente luego de numerar las hojas

#### 4. Hipótesis y Aclaraciones Globales

Describe las hipótesis que ha considerado para la resolución del TP. Documente cualquier otra aclaración que se considere necesaria. Todas las hipótesis deben presentarse en este punto, puede agruparlas por comando.

#### 5. Problemas relevantes

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

#### 6. Archivo README

Incluya la impresión del README en la carpeta. Ver detalles de su contenido en "Recomendaciones para el equipo de integración y testing" y en la especificación del instalador.

#### 7. Hoja de ruta de prueba "camino feliz"

Instale el tp

Imprima el log de instalación y el contenido del archivo de configuración

Ejecute Initializer y permita que el demonio arranque. Imprima el log de Initializer

Tome un par de archivos de listas de compras que tengan nombres aceptables y deposítelos en NOVEDIR.



Imprima el log de [Listener](#) y el contenido de las listas de compras aceptadas.

Imprima el log de [Masterlist](#) y un contenido parcial de la lista maestra de precios (pequeño)

Imprima el log de [Rating](#) y el contenido de los archivos presupuestados.

Ejecute [Reporting](#) con todas las variantes que considere adecuadas, imprima las invocaciones y los distintos resultados obtenidos.

#### 8. Listado de Nuevas Funciones y/o Comandos Auxiliares

Brinde un listado de las nuevas funciones y/o comandos auxiliares creados por Ustedes, es decir, que no figuran en el enunciado original del TP.

Indique: Nombre de la función, quienes la usan, para que la usan.

Si no crea ninguna, indique: NINGUNA

#### 9. Listado de DATOS

Imprima un listado con algunas listas de compras usadas en la prueba "camino feliz"

#### 10. Listado de Nuevos Archivos

Brinde un listado de los nuevos archivos creados por Ustedes, es decir, que no figuran en el enunciado original del TP.

Indique: Nombre del archivo, si es temporal o permanente, donde lo almacenan, quienes lo usan, para que lo usan. Si no crea ninguno, indique: NINGUNO

#### 11. Apéndice

Incluya este ENUNCIADO completo del TP (sin la caratula ni las planillas de evaluación dado que ya fueron incluidas al principio de la carpeta)

### **Recomendaciones para el equipo de desarrollo**

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. Se debe evitar el uso de archivos auxiliares permanentes
5. Si se emplean archivos auxiliares temporales, se deben eliminar ANTES de finalizar la ejecución del comando
6. Código de Retorno

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 loguear si finalizó correctamente o con errores.

#### 7. 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 Mover para el movimiento de archivos de datos la cual debe ser empleada por todos los comandos que la requieran

#### 8. Manejo de errores, logueo

Todo evento que genera algún tipo de error debe ser logueado. Si el comando no graba en un log específico, muestre el error por pantalla. Evite detener el proceso de evaluación del TP por no poder detectar que está pasando. También puede incorporar al comando una señal que permita dejar pistas de auditoría y se enciende si es necesario.

### **Recomendaciones para el equipo de integración y testing**

1. La carpeta de entrega del TP y el paquete de instalación INCLUYEN la demostración de que llegaron a ejecutar el camino feliz. Debido a esto es importante que efectúen la prueba de integración varios días antes de la entrega del TP dado que usualmente surgen errores de comunicación entre los comandos encadenados.



2. El TP debe correr en las instalaciones de la FIUBA, por lo tanto una integración en sus equipos personales no garantiza que dentro del entorno de la FIUBA la integración muestre los mismos resultados. Ejemplo de esto es el retorno que da el comando ps.
3. Recuerde que dentro del entorno de ejecución de la FIUBA su usuario NUNCA puede ser ROOT. El usuario a emplear es "alumnos"

#### 4. Lotes de Prueba

Se proveerán los archivos de prueba con un alto porcentaje de información libre de error. El grupo debe traer un juego extra de listas de compras con casos lo suficientemente heterogéneos como para contemplar todas las variantes de ejecución, en particular las de rechazo o error

#### 5. Archivo README, responsabilidad del equipo de integración

Como parte de la documentación del sistema se debe proveer un archivo README en donde se brinden indicaciones de instalación y de ejecución de los comandos, como ser:

- a) Una explicación de cómo copiar desde un medio externo el instalable, a modo de ejemplo:
  - Insertar el dispositivo de almacenamiento con el contenido del tp
  - Crear en el directorio corriente un directorio de trabajo
  - Copiar el archivo \*.tgz en ese directorio
  - Descomprimir el \*.tgz de manera de generar un \*.tar
  - Extraer los archivos del tar.
  - ... etc.
- b) Instrucciones de instalación
- c) Que se requiere para poder instalar, Que nos deja la instalación y donde
- d) Cuáles son los primeros pasos para poder correr el paquete una vez instalado
- e) Que comprobaciones se pueden hacer para asegurar que todo está en condiciones para empezar
- f) Como frenar la ejecución de comandos

### Evaluación

El día de vencimiento del TP, cada ayudante convocará a los integrantes de un grupo, solicitará la carpeta y el paquete de instalación e iniciará la corrección mediante una entrevista grupal. Es imprescindible la presencia de todos los integrantes del grupo el día de la corrección

El TP debe correr en las instalaciones de la FIUBA, en los equipos de los Laboratorios C y/o F.

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.



## Carpeta

La documentación debe entregarse en una carpeta con TODAS las hojas numeradas y enganchadas. Las hojas sueltas no se considerarán como parte de la misma.

El pie de página de cada hoja debe tener: Número de Grupo y Tema (en el margen izquierdo) y Número de Hoja (en el margen derecho). **La numeración puede ser manual.**

## Paquete

El paquete de instalación se deberá remitir vía correo electrónico a [so7508@gmail.com](mailto:so7508@gmail.com). En el asunto del correo indicar Nro. de Grupo y Ayudante asignado.

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:

- El archivo README
- Los scripts desarrollados
- Los Archivos de prueba entregados por la cátedra (conservarlos en un directorio aparte para poder reutilizarlos en la evaluación)
- Todos los que se generaron durante la ejecución del camino feliz (incluidos los logs)
- La documentación
- Cualquier otro archivo que se considere necesario

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 los laboratorios de la facultad.

## ¿Dónde instalamos?

Hay un único usuario (alumnos) disponible para la evaluación del TP, debido a esto cada grupo deberá emplear su propio subdirectorio de trabajo llamado grupox (donde xx es el nro. de grupo).

Recuerde que dentro del entorno de ejecución de la FIUBA su usuario NUNCA puede ser ROOT

Antes de comenzar la instalación asegúrese que:

- la última versión del TP está grabada en el dispositivo que trajo
- que la terminal que se usara para la evaluación posea interfaz para la conexión del dispositivo

Una vez que conecte el dispositivo copie su contenido a su subdirectorio de trabajo.

Sus directorios y archivos pueden ubicarse en cualquier lugar a partir del directorio de trabajo grupox. Resguarde el paquete original para poder efectuar reinstalación cuando sea necesario

La extracción del paquete debe generar AL MENOS estos dos subdirectorios:

- grupox/[conf](#), sin archivos. La ubicación del directorio de configuración debe estar fija para que sea fácilmente reconocida por el instalador y el comando de inicialización de ambiente
- grupox/datos en donde depositaran los Archivos de prueba entregados por la cátedra

Para simplificar esta narrativa de aquí en mas a todo el camino (path) que va desde la raíz hasta grupox lo llamaremos genéricamente "\$grupo" y a \$grupo/ <a href="#">conf</a> lo llamaremos CONFDIR.
---





## Estructuras y Ejemplos

### Archivos del Sistema

#### Archivo de Configuración: CONFDIR/Installer.conf

Campo	Descripción/Fuente/Valor	
Variable	Caracteres	Valores posibles: GRUPO, CONFDIR, BINDIR, MAEDIR, NOVEDIR, DATASIZE, ACEPDIR, RECHDIR, INFODIR, LOGDIR, LOGEXT, LOGSIZE
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=30/04/2014 10:03 p.m

Se debe grabar un registro por cada variable. Luego de los registros requeridos, puede agregar todos los registros que desee, es decir que a partir de LOGSIZE, el uso del archivo de configuración es de libre disponibilidad para los desarrolladores.

#### Log de la Instalación: CONFDIR/Installer.log

#### Log de Comandos: LOGDIR/<comando>.LOGEXT

Campo	Descripción /Fuente/Valor	
Cuando	Fecha y hora	Formato a Elección Es la fecha y hora en el momento de grabación del registro.
Quien	Caracteres	Es el login del usuario que graba el 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: 20140509 10:53:22-alumnos-[Initializer](#)-WAR-No se pudo arrancar [Listener](#).



## Archivos Maestros y Tablas

### Maestro de Asociados y Colaboradores: asociados.mae

Campo	Descripción/ Fuente/Valor	
Apellido	Caracteres	Obligatorio - Apellido del Asociado o Colaborador
Nombre	Caracteres	Obligatorio - Nombre del Asociado o Colaborador
Usuario	Caracteres	Obligatorio – usuario del Asociado o Colaborador
Colaborador?	Un carácter	Obligatorio - Valores posibles: 0 y 1 0 (NO es colaborador) - 1 (no es colaborador)
Correo Electrónico	Caracteres	Correo Electrónico del Asociado o Colaborador. Es obligatorio en el caso de colaborador

Separador de campos: ; punto y coma

Ejemplo del maestro de asociados ASOCIADOS.MAE				
Apellido	Nombre	Usuario	Colaborador?	Correo Electrónico
FERRUCCI	DARIO	FERRUCCIDARIO	1	FERRUCCI_DARIO@GMAIL.COM
LEYER	DEMIAN	LEYERDEMIAN	0	LEYER_DEMIAN@GMAIL.COM
FACIONI	DIEGO	FACIONIDIEGO	0	FACIONI_DIEGO@HOTMAIL.COM
ALONSO	EDITH	ALONSOEDITH	0	ALONSO_EDITH@HOTMAIL.COM

### Maestro de Supermercados: super.mae

Campo	Descripción/ Fuente/Valor	
SUPER_ID	Caracteres	Id del supermercado. Es clave univoca.
PROVINCIA	Caracteres	Provincia - obligatorio
NOMBRE_SUPER	Caracteres	Nombre del Supermercado -obligatorio
NRO_DOC	Caracteres	Nro. de Documento - opcional
TIPO_DOC	Caracteres	Tipo de Documento - opcional
DIRECCION	Caracteres	Dirección legal del supermercado

Separador de campos: ; punto y coma

Ejemplo del maestro de supermercados SUPER.MAE					
SUPER_ID	PROVINCIA	NOMBRE_SUPER	NRO_DOC	TIPO_DOC	DIRECCION
1	Buenos Aires	Precios Cuidados	N/A	N/A	N/A
112	Buenos Aires	Coto	30362447534	CUIT	Av. Sáenz 220
134	Buenos Aires	Walmart	33456666367	CUIT	Av. Santa Marina 97
601	Cordoba	Autoservicio Eco	23805112	DNI	Av. Sarmiento 908
1202	La Rioja	Carrefour	30350102311	CUIT	Campos 3854



## Lista Maestra de Precios: precios.mae

Campo	Descripción/ Fuente/Valor	
SUPER_ID	Caracteres	Id del supermercado -obligatorio. Se obtiene del maestro de supermercados.
COLABORADOR	Caracteres	Usuario -obligatorio. Se obtiene del nombre (filename) del archivo de novedad
FECHA ARCHIVO	Fecha	Fecha con formato aaaammdd -obligatorio. Se obtiene del nombre (filename) del archivo de novedad
PRODUCTO	Caracteres	Descripción del Producto –obligatorio. Se obtiene del campo producto del archivo de novedad
PRECIO	Numérico	Precio del Producto – obligatorio. Se obtiene del campo precio del archivo de novedad

Separador de campos: ; punto y coma

### Ejemplo de lista maestra de precios: PRECIOS.MAE

SUPER_ID	COLABORADOR	FECHA ARCHIVO	PRODUCTO	PRECIO
1	ZAIATZOSCAR	20140305	COCINERO ACEITE DE GIRASOL BOTELLA DE PLASTICO - 1500 CM3	12.02
1	ZAIATZOSCAR	20140305	ALCO ARVEJAS EN LATA - 350 G	6.30
1	ZAIATZOSCAR	20140305	CHANGO AZUCAR BLANCA - 1 KG	7.59
1	ZAIATZOSCAR	20140305	LA SERENISIMA DULCE DE LECHE ENTERO CLASICO POTE - 400 G	12.95
1	ZAIATZOSCAR	20140305	BOCA DE DAMA GALLETTITAS DULCES ENVASADAS SECAS SIN RELLENO INDIVIDUAL - 200 G	6.20

## Tabla de Equivalencias de Unidades de Medida: um.tab

Campo	Descripción/ Fuente/Valor	
UM1	Caracteres	Unidad de medida
UM2	Caracteres	Unidad de medida
UM3	Caracteres	Unidad de medida
UM4	Caracteres	Unidad de medida
UM5	Caracteres	Unidad de medida
UM6	Caracteres	Unidad de medida
UM7	Caracteres	Unidad de medida
UM8	Caracteres	Unidad de medida
UM9	Caracteres	Unidad de medida
UM10	Caracteres	Unidad de medida

Separador de campos: ; punto y coma

### Registros de la tabla de Equivalencias: um.tab

UM1	UM2	UM3	UM4	UM5	UM6	UM7	UM8	UM9	UM10
cc	cc.	cm3	cm3.	ml	ml.				
g	g.	gr	gr.	grm	grm.	grms	grms.	grs	grs.
k	k.	kg	kg.	kgm	kgm.	kgr	kgr.	kilos	
l	l.	lt	lt.	ltr	ltr.	ltrs	ltrs.	litros	
u	u.	un	un.	uni	uni.	unidades			

NOTA: traer un archivo um.tab con todos estos registros.



## Archivos de Novedades

### Novedades “Listas de Compras”: usuario.xxx

Campo	Descripción/ Fuente/Valor	
NRO de ITEM	Numérico	Nro. secuencial creciente, empieza en 1
PRODUCTO PEDIDO	Caracteres	Descripción del Producto

Separador de campos: ; punto y coma

#### Ejemplo lista de compras

NRO de ITEM	PRODUCTO PEDIDO
1	Azucar chango 1 kg.
2	Caldo MAGGI 114 grm
3	Fideos soperos semolados MATARAZZO pamperito 3 vegetales 500 gr
4	Aceite Girasol Cocinero 1500 CC

### Novedades “Listas de Precios”: super-fecha.usuario

Campos del registro cabecera	Descripción/Fuente/Valor	
NOMBRE_SUPER	Caracteres	Nombre del Supermercado -obligatorio
PROVINCIA	Caracteres	Provincia - obligatorio
CANTIDAD DE CAMPOS	Numérico	Cantidad de campos en los registros de detalle
UBICACIÓN DEL PRODUCTO	Numérico	Ubicación del campo producto dentro del registro de detalle. Mayor a 0
UBICACIÓN DEL PRECIO	Numérico	Ubicación del campo precio dentro del registro de detalle. Mayor a 0
CORREO DEL COLABORADOR	Caracteres	Correo electrónico del colaborador que remite el archivo.

Separador de campos: ; punto y coma

#### Ejemplo registro cabecera del archivo Lista de Precios: PC-20140305.ZAIATZOSCAR

NOMBRE_SUPER	PROVINCIA	CANTIDAD de CAMPOS	UBICACIÓN del PRODUCTO	UBICACIÓN del PRECIO	CORREO ELECTRÓNICO DEL COLABORADOR
Precios Cuidados	Buenos Aires	5	2	3	ZAIATZ_OSCAR@GMAIL.COM

#### Ejemplo registros de detalle del archivo de Lista de Precios: PC-20140305.ZAIATZOSCAR

ALMACeN	COCINERO ACEITE DE GIRASOL BOTELLA DE PLASTICO - 1500 CM3	12.02	COCINERO	MOLINOS RÍO DE LA PLATA
ALMACeN	ALCO ARVEJAS EN LATA - 350 G	6.30	ALCO	GRUPO CANALE
ALMACeN	CHANGO AZUCAR BLANCA - 1 KG	7.59	CHANGO	TABACAL AGROINDUSTRIA
ALMACeN	LA SERENISIMA DULCE DE LECHE ENTERO CLÁSICO POTE - 400 G	12.95	LA SERENISIMA	MASTELLONE
ALMACeN	BOCA DE DAMA GALLETITAS DULCES ENVASADAS SECAS SIN RELLENO INDIVIDUAL - 200 G	6.20	BOCA DE DAMA	TERRABUSI-KRAFT MONDELEZ



## Archivos de Retail

### Listas Presupuestadas: usuario.xxx

Campo	Descripción/Fuente/Valor	
NRO de ITEM	Numérico	Obligatorio. Se obtiene de la lista de compras
PRODUCTO PEDIDO	Caracteres	Obligatorio. Se obtiene de la lista de compras
SUPER_ID	Numérico	Id del supermercado -opcional. Se obtiene de la lista maestra de precios
PRODUCTO ENCONTRADO	Caracteres	Descripción del Producto – opcional. Se obtiene de la lista maestra de precios
PRECIO	Numérico	Precio del Producto – opcional. Se obtiene de la lista maestra de precios

Separador de campos: ; punto y coma

Ejemplo Lista Presupuestada				
NRO de ITEM	PRODUCTO PEDIDO	SUPER_ID	PRODUCTO ENCONTRADO	PRECIO
1	Azucar chango 1 kg.	1	CHANGO AZUCAR BLANCA - 1 KG	7.59
1	Azucar chango 1 kg.	112	Azucar CHANGO est 1 kgm	12.19
1	Azucar chango 1 kg.	112	Azucar CHANGO organica est 1 kg	19.89
1	Azucar chango 1 kg.	134	Azucar chango 1 kg	7.59
1	Azucar chango 1 kg.	134	Azucar tetra chango 1 kg	10.66
2	Caldo MAGGI 114 grm	1	MAGGI CALDO DE GALLINA - 114 G	7.99
2	Caldo MAGGI 114 grm	112	Caldo MAGGI carne cja 114 grm	9.03
2	Caldo MAGGI 114 grm	112	Caldo MAGGI verdura cja 114 grm	9.03
2	Caldo MAGGI 114 grm	134	Caldo de carne maggi 114 gr	8.20
2	Caldo MAGGI 114 grm	134	Caldo de gallina maggi 114 gr	7.99
2	Caldo MAGGI 114 grm	134	Caldo de verdura maggi 114 gr	8.50
3	Fideos soperos semolados MATARAZZO pamperito 3 vegetales 500 gr			
4	Aceite Girasol Cocinero 1500 CC	1	COCINERO ACEITE DE GIRASOL BOTELLA DE PLASTICO - 1500 CM3	12.02



## Especificación de Comandos y Funciones

A continuación se brinda una descripción de cada función o comando solicitado.

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 y lo aclare debidamente en las hipótesis.

Si el equipo de desarrollo lo considera pertinente puede aumentar la funcionalidad requerida o crear nuevas funciones de librería que crea conveniente.

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

### **Arranque: Start**

#### **Opciones y Parámetros**

- parámetros u opciones a especificar por el desarrollador

#### **Descripción**

Esta función tiene por objeto disparar procesos. Es complementaria a [Stop](#).

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

No se puede arrancar un proceso si éste ya se encuentra corriendo. No se puede arrancar un proceso si la inicialización de ambiente no fue realizada. Indicar en las Hipótesis Globales donde se realizan estos controles

Opción a. en forma centralizada en esta función

Opción b. el control se delega al comando llamador

Si esta función es invocada por un comando que graba en un archivo de log, actualizarlo indicando el resultado de la operación

Explicar claramente su invocación y uso en el README.

### **Detención: Stop**

#### **Opciones y Parámetros**

- parámetros u opciones a especificar por el desarrollador

#### **Descripción**

Esta función tiene por objeto detener procesos. Es complementaria a [Start](#)

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

Si esta función es invocada por un comando que graba en un archivo de log, actualizarlo indicando el resultado de la operación

Explicar claramente su invocación y uso en el README.



## **Movimiento de Archivos: Mover**

### **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 mover un archivo de un directorio a otro contemplando la posibilidad de archivos duplicados.

Cuando se detecta un archivos duplicado, se debe gestionar un numero de secuencia nnn. Este número de secuencia puede ser de cálculo centralizado (una única secuencia para todo el sistema) o descentralizado (diferentes secuencias).

EVITE crear un archivo auxiliar solo para registrar el número de secuencia

Si requiere registrar el número de secuencia en un archivo utilice el archivo de configuración agregando al final un registro con la variable y el valor. Documente esto en las hipótesis generales del sistema.

Si esta función es invocada por un comando que graba en un archivo de log, actualizarlo indicando el resultado de la operación

**Mover** debe devolver un código de retorno cero (0) si fue exitoso el movimiento o distinto de cero si tuvo errores.

### **Pasos Sugeridos**

1. Verificar si el origen y el destino son iguales. Si este fuera el caso, no mover

Si el origen no existe o el destino no existe, es un error. Si este fuera el caso, no mover

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:

- Ver si existe dentro del directorio destino un subdirectorio /dup. si no existe, crearlo
- Mover el archivo a ese subdirectorio con el siguiente nombre: <nombre del archivo original>.nnn dónde nnn es un número de secuencia que evita nombres duplicados, es decir, evita "sobreescribir" archivos.
- Eliminar el archivo del directorio de origen

Ejemplo mover arch.txt de /data1 a /data2

Antes de MOVER	Luego de MOVER
/data1/arch.txt	/data1
/data2/arch.txt	/data2/arch.txt
	/data2/dup/arch.txt.1

3. Si no es un archivo duplicado, entonces hacer el move al destino indicado.

Ejemplo mover arch.txt de /data1 a /data2

Antes de MOVER	Luego de MOVER
/data1/arch.txt	/data1
/data2	/data2/arch.txt



## Logueo: Logging

### 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, auditoría y seguridad informática.

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

#### ¿Cómo deben ser los log en este TP?

Cumplir con el estándar W5

En el contexto de este TP, la escritura de archivos de log debe ser homogénea para todos los comandos y cada registro de log debe responder al estándar W5.

WHEN: ¿Cuándo?

Fecha y Hora, en el formato que deseen y calculada justo antes de la grabación.

WHO: ¿Quién?

Usuario, es el login del usuario

WHERE: ¿Dónde?

Comando (parámetro 1), nombre del comando o función que genera el mensaje.

Se apreciará la utilidad de este parámetro cuando la función Mover deba generar mensajes de log

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.  
Valor default: INFO

WHY: ¿Por qué?

Mensaje (Parámetro 2)

### Evitar el crecimiento INDISCRIMINADO

En todo sistema, es importante evitar el crecimiento INDISCRIMINADO de los archivos de Log. Es por ello que esta función debe proveer un mecanismo para evitarlo.

En este mecanismo debe tener en cuenta la variable de configuración **LOGSIZE** que representa el tamaño máximo que puede alcanzar un archivo de log en nuestro sistema.

Este tamaño máximo es un valor de referencia ya que a los efectos prácticos, todo depende del momento en que se realiza el control.





---

Lo importante es que SIEMPRE adopte un mecanismo para mantener controlado el tamaño de un log. Puede adoptar cualquier mecanismo, aclare en **Hipótesis y Aclaraciones Globales** cual fue el que adoptó.

Cada vez que se reduce el tamaño del log, grabar en el nuevo log el mensaje: Log Excedido para poder controlar que se está realizando este trabajo.

**Ejemplo sencillo:** cuando un archivo de log supera **LOGSIZE**, el archivo se trunca dejando las últimas 50 líneas

**Depositar los log en el lugar correcto y con el nombre indicado.**

El directorio de logs está determinado por la variable de configuración **LOGDIR** (excepto para el log de la instalación)

El nombre del archivo es igual al nombre del comando

La extensión del archivo está determinada por la variable de configuración **LOGEXT** (excepto para el log de la instalación)

Si el archivo de log no existe, se debe crear. Si existe se le deben agregar los nuevos registros.



## **Instalación: Installer**

### **Opciones y Parámetros**

- A especificar por el desarrollador (Explicar claramente en el README)

### **Descripción**

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

Interactúa con el usuario para solicitarle que defina subdirectorios, espacio requerido, extensión de archivos, etc.

El usuario puede elegir cualquier nombre de directorio excepto `/conf` o `/datos`. Estos ya fueron creados en la extracción del paquete.

Siempre que se interactúa con el usuario proponer valores por default mostrándolos entre paréntesis.

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

Si durante el mismo hilo (o instancia) de ejecución **Installer** debe reiniciar desde un paso 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” durante su ejecución.

### **Pasos sugeridos**

#### **1. Inicializar archivo de log**

Este comando graba un archivo de Log cuyo nombre es **Installer.log** en el directorio **CONFDIR**.

Si el archivo de log no existe, crearlo. Si existe, agregarle los nuevos registros.

Un log de instalación debe permitir recrear en forma completa el proceso de instalación, es decir: las condiciones antes de la instalación, las decisiones tomadas durante la instalación, las condiciones finales.

En el contexto de este TP, la escritura de archivos de log debe ser homogénea para todos los comandos incluido este y cada registro de log debe responder al estándar W5 explicado en la función **Logging**

El primer mensaje que se debe grabar es el mensaje de inicio de ejecución (todos los mensajes que se muestran son a modo de ejemplo)

Inicio de Ejecución de **Installer**

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

Log de la instalación: **CONFDIR/Installer.log**

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

Directorio predefinido de Configuración: **CONFDIR**

#### **4. Detectar si el paquete **RETAILD** o alguno de sus componentes ya está instalado**

Este script de instalación se debe preparar de manera tal que detecte si el paquete o alguno de sus componentes ya se encuentran instalados

Una forma de hacer esto es verificando la existencia del archivo **Installer.conf** en el directorio **CONFDIR**

- Si **Installer.conf** no existe, asumir que el paquete no fue instalado. El proceso sigue en el siguiente punto



- Si **Installer.conf** existe, asumir que el paquete ya fue instalado y verificar que la instalación este completa, o indicar los faltantes

4.1. Si esta completo, mostrar y grabar en el log los siguientes mensajes y luego ir a FIN

```
TP SO7508 Primer Cuatrimestre 2014. Tema D Copyright © Grupo xx

Direct. de Configuración: CONFDIR (mostrar path y listar archivos)

Directorio Ejecutables: BINDIR (mostrar path y listar archivos)

Direct Maestros y Tablas: MAEDIR (mostrar path y listar archivos)

Directorio de Novedades: NOVEDIR

Dir. Novedades Aceptadas: ACEPDIR

Dir. Informes de Salida: INFODIR

Dir. Archivos Rechazados: RECHDIR

Dir. de Logs de Comandos: LOGDIR/<comando>.LOGEXT

Estado de la instalación: COMPLETA

Proceso de Instalación Cancelado
```

4.2. Si falta instalar algún componente, listar directorios, archivos existentes y faltantes indicando que la instalación esta incompleta y preguntar si desea completarla (siempre grabando todo en el log)

Si el usuario indica Si, Continuar en el paso: "CONFIRMAR INICIO DE INSTALACIÓN"

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

```
TP SO7508 Primer Cuatrimestre 2014. Tema D Copyright © Grupo xx

Direct. de Configuración: CONFDIR (mostrar path y listar archivos)

Directorio Ejecutables: BINDIR (mostrar path y listar archivos)

Direct Maestros y Tablas: MAEDIR (mostrar path y listar archivos)

Directorio de Novedades: NOVEDIR

Dir. Novedades Aceptadas: ACEPDIR

Dir. Informes de Salida: INFODIR

Dir. Archivos Rechazados: RECHDIR

Dir. de Logs de Comandos: LOGDIR/<comando>.LOGEXT

Componentes faltantes: listado de los componentes faltantes
```



Estado de la instalación: **INCOMPLETA**

Desea completar la instalación? (Si-No)

4.3. Si el paquete no fue instalado, continuar

## 5. Aceptación de términos y condiciones

Mostrar en consola el siguiente mensaje:

```
TP SO7508 Primer Cuatrimestre 2014. Tema D Copyright © Grupo xx
```

```
Al instalar TP SO7508 Primer Cuatrimestre 2014 UD. expresa aceptar los  
términos y condiciones del "ACUERDO DE LICENCIA DE SOFTWARE" incluido en  
este paquete.
```

```
Acepta? Si - No
```

Si el usuario indica No, ir a FIN

Si el usuario indica Si, continuar

Cualquier otra cosa, seguir en este punto

## 6. Chequear que Perl esté instalado

Para la instalación de TP SO7508 **Primer Cuatrimestre 2014** es necesario chequear que Perl versión 5 o superior este instalado.

6.1. Si al chequear esto da error, entonces mostrar y grabar en el log:

```
TP SO7508 Primer Cuatrimestre 2014. Tema D Copyright © Grupo xx
```

```
Para instalar el TP es necesario contar con Perl 5 o superior. Efectúe su  
instalación e inténtelo nuevamente.
```

```
Proceso de Instalación Cancelado
```

Ir a **FIN**

6.2. Si Perl está instalado, mostrar y grabar en el log un mensaje informativo con la versión de perl que se encuentra instalada y continuar.

```
TP SO7508 Primer Cuatrimestre 2014. Tema D Copyright © Grupo xx
```

```
Perl Version: <mostrar la salida de perl -v>
```

## 7. Definir el directorio de instalación de los ejecutables

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

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

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

El usuario puede ingresar un nombre simple como "bin" o un subdirectorio como /tp/sistemas/bin

Reservar este path en la variable **BINDIR**

## 8. Definir el directorio de instalación de los archivos maestros y tablas

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

```
Defina directorio para maestros y tablas ($grupo/mae):
```

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



El usuario puede ingresar un nombre simple como "mae" o un subdirectorio como /data/mae

Reservar este path en la variable **MAEDIR**

### 9. Definir el directorio de input del proceso Listener

Se debe solicitar al usuario que indique cual es el directorio de arribo de archivos de novedades. Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Defina el Directorio de arribo de novedades ($grupo/arribos):
```

Proponer /arribos y si el usuario lo desea cambiar, permitirle.

El usuario puede ingresar un nombre simple como "/arribos" o un subdirectorio como /archivos/externos/arribos

Reservar este path en la variable **NOVEDIR**

### 10. Definir el espacio mínimo libre para el arribo de archivos de novedades

Se debe solicitar al usuario que indique cual es el espacio mínimo libre en el directorio **NOVEDIR**

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

```
Defina espacio mínimo libre para el arribo de novedades 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**.

### 11. Verificar espacio en disco

Chequear si en **NOVEDIR** hay disponibles por lo menos **DATASIZE** Mb. Si esto da error mostrar y grabar en el log el siguiente mensaje:

```
Insuficiente espacio en disco.
```

```
Espacio disponible: xx Mb.
```

```
Espacio requerido $DATASIZE Mb
```

```
Cancele la instalación o inténtelo nuevamente.
```

Volver a verificar espacio en disco.

### 12. Definir el directorio de input del proceso Rating

Se debe solicitar al usuario que indique cual es el directorio que desea usar para mover las Novedades aceptadas. Mostrar y grabar en el log el siguiente mensaje con su respuesta:

```
Defina el directorio de grabación de las Novedades aceptadas ($grupo/aceptadas):
```

Proponer /aceptadas y si el usuario lo desea cambiar, permitirle.

El usuario puede ingresar un nombre simple como /aceptadas o un subdirectorio como /archivos/externos/aceptadas

Reservar este path en la variable **ACEPDIR**

### 13. Definir el directorio de trabajo principal del proceso Reporting.

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

```
Defina el directorio de grabación de los informes de salida ($grupo/informes):
```

Proponer /informes y si el usuario lo desea cambiar, permitirle.



El usuario puede ingresar un nombre simple como `/informes` o un subdirectorio como `archivos/salida/informes`

Reservar este path en la variable **INFODIR**.

#### 14. Definir el repositorio de logs

Se debe solicitar al usuario el directorio donde desea dejar los archivos de log que generan los comandos principales (excepto el log de instalación que queda fijo en **CONFDIR**), para ello mostrar en pantalla el siguiente mensaje:

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

Si el usuario acepta el nombre propuesto, continuar

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

Reservar este path en la variable **LOGDIR**.

#### 15. Definir la extensión para los archivos de log

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

```
"Ingrese la extensión para los archivos de log: (.log)"
```

Proponer `.log` y si el usuario lo desea cambiar, permitirérselo.

Reservar ese valor en la variable **LOGEXT**.

#### 16. Definir el tamaño máximo para los archivos de log

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

```
Defina el tamaño máximo para los archivos $LOGEXT en Kbytes (400):
```

Proponer 400 KB, Si el usuario lo desea cambiar, debe ingresar una cantidad que se interpreta como Kb.

Reservar este valor en la variable **LOGSIZE**.

#### 17. Definir repositorio de archivos rechazados

Se debe solicitar al usuario que indique cual es el directorio que desea usar para mover los archivos rechazados. Mostrar y grabar en el log el siguiente mensaje con su respuesta:

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

Proponer `/rechazados` y si el usuario lo desea cambiar, permitirérselo.

El usuario puede ingresar un nombre simple como `/rechazados` o un subdirectorio como `/archivos/externos/rechazados`

Reservar este path en la variable **RECHDIR**

#### 18. Mostrar estructura de directorios resultante y los valores de los parámetros configurados

Limpiar la pantalla

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

```
TP SO7508 Primer Cuatrimestre 2014. Tema D Copyright © Grupo xx
```

```
Direct. de Configuración: CONFDIR (mostrar path y listar archivos)
```

```
Directorio Ejecutables: BINDIR (mostrar path y listar archivos)
```

```
Direct Maestros y Tablas: MAEDIR (mostrar path y listar archivos)
```



Directorio de Novedades: **NOVEDIR**

Espacio mínimo libre para arribos: **DATASIZE** Mb

Dir. Novedades Aceptadas: **ACEPDIR**

Dir. Informes de Salida: **INFODIR**

Dir. de Logs de Comandos: **LOGDIR**/**<comando>**.**LOGEXT**

Tamaño máximo para los archivos de log del sistema: **LOGSIZE** Kb

Dir. Archivos Rechazados: **RECHDIR**

Estado de la instalación: **LISTA**

Si el usuario indica Si, Continuar en el paso: “**CONFIRMAR INICIO DE INSTALACIÓN**”

Si el usuario indica No, Limpiar la pantalla y continuar después del chequeo de Perl: En este caso, los valores default propuestos deben ser los recientemente ingresados por el instalador, es decir los contenidos en las variables: **BINDIR**, **NOVEDIR**, **DATASIZE**, **INFODIR**, **LOGDIR**, **LOGEXT**, **LOGSIZE**, etc.

## 19. CONFIRMAR INICIO DE 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**

## 20. Instalación

20.1. Crear las estructuras de directorio requeridas mostrando el siguiente mensaje:

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

**\$BINDIR**

**\$MAEDIR**

**\$MAEDIR/listas**

**\$MAEDIR/listas/procesadas**

**\$NOVEDIR**

**\$ACEPDIR**

**\$ACEPDIR/procesadas**

**\$INFODIR**

**\$INFODIR/listas**

**\$LOGDIR**

**\$RECHDIR**

20.2. Mover los archivos maestros y tablas al directorio MAEDIR mostrando el siguiente mensaje

```
Instalando Archivos Maestros y Tablas
```



20.3. Mover los ejecutables y funciones al directorio BINDIR mostrando el siguiente mensaje

Instalando Programas y Funciones

20.4. Actualizar el archivo de configuración mostrando el siguiente mensaje

Actualizando la configuración del sistema

Se debe almacenar la información de configuración del sistema en el archivo [Installer.conf](#) en **CONFDIR**

Si el archivo de configuración no existe, crearlo, si existe actualizar los valores que correspondan.

Se debe grabar un registro para cada una de las siguientes variables:

GRUPO, CONFDIR, BINDIR, [MAEDIR](#), NOVEDIR, DATASIZE, ACEPDIR, INFODIR, RECHDIR, LOGDIR, LOGEXT, LOGSIZE

20.5. Borrar archivos temporarios, si los hubiese generado

20.6. Mostrar mensaje de fin de instalación

Instalación **CONCLUIDA**

## 21. FIN

Cerrar el archivo [Installer.log](#) y Terminar el proceso





## **Inicialización: *Initializer***

### **Descripción**

El propósito de este comando es preparar el entorno de ejecución del TP (ambiente).

- Es el primero en orden de ejecución
- Se dispara manualmente
- Graba en el archivo de Log a través del [Logging](#)
- Invoca, si corresponde, al siguiente paso [Listener](#)

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, archivos maestros y tablas con los permisos adecuados.

Continúa con la asignación de valor a un conjunto de variables de ambiente y luego ofrece arrancar automáticamente el comando de recepción de novedades.

El resto de los comandos JAMAS deben acceder al archivo de configuración para conocer directorios, deben usar las variables de ambiente que define este proceso.

### **Pasos Sugeridos**

1-Inicializar el archivo de log

Comando [Initializer](#) Inicio de Ejecución

2-Verificar que la instalación está completa

Si se detecta algún faltante en la instalación, explicar la situación, indicar los componentes faltantes, dar indicaciones para que el usuario ejecute el instalador, reiniciar el sistema, etc...

Este control también debe contemplar que existan en el directorio **MAEDIR** los archivos maestros y tablas del sistema: um.tab, asociados.mae y super.mae.

Grabar siempre en el log y terminar la ejecución.

3- Verificar si el ambiente ya ha sido inicializado.

[Initializer](#) debe setear las variables de ambiente una sola vez por cada sesión de usuario.

Si se intenta ejecutar nuevamente este comando, explicar la situación, (por ejemplo indicar "ambiente ya inicializado, si quiere reiniciar termine su sesión e ingrese nuevamente" Grabar siempre en el log y terminar la ejecución.

4- Si [Initializer](#) nunca fue ejecutado en esta sesión de usuario

Setear la variable PATH y cualquier otra variable de ambiente que considere necesarias, como ser: **GRUPO, NOVEDIR, RECHDIR, BINDIR, MAEDIR, INFODIR, LOGDIR LOGEXT, etc.**

Si se detecta que algún archivo no tiene los permisos adecuados, setarlos correctamente y continuar en el siguiente paso

5-Ver si se desea arrancar Listener

[Initializer](#) debe ofrecer la posibilidad de arrancar el demonio [Listener](#) por ejemplo mostrando el siguiente mensaje:

"Desea efectuar la activación de [Listener](#)?" Si - No

5.1 Si el usuario no desea arrancar el demonio [Listener](#), entonces explicar cómo hacerlo con el comando [Start](#)

5.2 Si el usuario desea arrancar el demonio [Listener](#), activarlo (SOLO SI NO EXISTE OTRO [Listener](#) CORRIENDO) y explicar cómo detenerlo usando el comando [Stop](#).

**6-FINAL:**



Si el ambiente esta inicializado correctamente, mostrar el contenido de las variables, el process id del demonio (si es que está corriendo) y el estado del sistema, por ejemplo:

```
TP SO7508 Primer Cuatrimestre 2014. Tema D Copyright © Grupo xx
```

```
Direct. de Configuracion: CONFDIR (mostrar path y listar archivos)
```

```
Directorio Ejecutables: BINDIR (mostrar path y listar archivos)
```

```
Direct Maestros y Tablas: MAEDIR (mostrar path y listar archivos)
```

```
Directorio de Novedades: NOVEDIR
```

```
Dir. Novedades Aceptadas: ACEPDIR
```

```
Dir. Informes de Salida: INFODIR
```

```
Dir. de Logs de Comandos: LOGDIR/<comando>.LOGEXT
```

```
(listar todas las otras variables que deseen mostrar)
```

```
Dir. Archivos Rechazados: RECHDIR
```

```
Estado del Sistema: INICIALIZADO
```

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

Cerrar el archivo de log- Terminar el proceso



## **Recepción de novedades: Listener**

### **Input**

- Archivos (cualesquiera) que arriban al directorio \$**NOVEDIR**/**<nombre del archivo>**
- Maestro de Asociados y Colaboradores \$**MAEDIR**/asociados.mae

### **Output**

- Archivos Aceptados \$**ACEPDIR**/**<nombre del archivo>**
- Archivos Rechazados \$**RECHDIR**/**<nombre del archivo>**
- Log \$**LOGDIR**/**Listener**.\$**LOGEXT**

### **Opciones y Parámetros**

- A especificar por el desarrollador

### **Descripción**

El propósito de este comando es detectar la llegada de archivos al directorio \$**NOVEDIR** y aceptar/rechazar estos archivos.

- Es el segundo en orden de ejecución
- Es un proceso del tipo "Demonio" :
- Se dispara automáticamente o a través del **Start**
- Se detiene a través del **Stop**
- Mueve los archivos a través del **Mover**
- Graba en el archivo de Log a través del **Logging**
- Invoca, si corresponde **Masterlist**
- Invoca, si corresponde **Rating**

Los asociados envían archivos de listas de compras y los consultores envían archivos para realizar la actualización de la lista maestra de precios.

Si archivo es enviado por un asociado o consultor registrado, acepta el archivo, de lo contrario, lo rechaza.

Cuando termina con todos los arribos, invoca **Masterlist**, **Rating**, ambos o ninguno según corresponda.

Luego duerme un tiempo x y vuelve a empezar, es decir, que a menos que se detenga con **Stop**, 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).

Queda a consideración de cada grupo el valor que se asigna a la variable de tiempo x, para la prueba del TP se solicitará modificar el valor usando vi.

Mantener un contador de ciclos de ejecución del **Listener**. Para ello puede ser de utilidad el archivo de configuración.

### **Pasos sugeridos:**

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

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

Indicar en las hipótesis como realizan la contabilidad del ciclo.



## 2. Chequear si hay archivos en el directorio \$NOVEDIR.

Si existen archivos, por cada archivo que se detecta

- Verificar que el archivo sea un archivo común, de texto. Los archivos de cualquier otro tipo, se rechazan.
- Verificar que el formato del nombre del archivo sea correcto, los archivos con nombres que no se correspondan con el formato esperado, se rechazan.

## 3. Validación del nombre de los archivos:

En este directorio se reciben dos tipos de archivos:

- Los archivos de listas de compras
- Los archivos de listas de precios

Cuando el archivo corresponde a una lista de compras, el nombre del archivo tiene el siguiente formato: **usuario.xxx**

- El usuario debe existir en el maestro de asociados.
- Luego del usuario viene un punto “.”
- Y finalmente xxx que es cualquier combinación de caracteres sin guiones ni espacios. No realizar ninguna validación de su contenido.

Cuando el archivo corresponde a una lista de precios, el nombre del archivo tiene el siguiente formato: **super.fecha.usuario**. Ejemplo: PC-20140305.ZAIATZOSCAR

- Super es una cadena de caracteres sin espacios. No realizar ninguna validación de su contenido.
- Luego de super viene “-”
- Fecha viene con formato aaaammdd. Se debe verificar que la fecha sea una fecha válida, menor o igual a la fecha del día, mayor a 20140101
- Luego de fecha viene un punto “.”
- El usuario debe existir en el maestro de asociados y debe ser de un colaborador (colaborador = 1)

## 4. Aceptar los archivos con nombre válido.

Si el nombre del archivo es válido y es de lista de compras mover el archivo aceptado a \$ACEPDIR empleando la función Mover y grabar en el log el mensaje de éxito

Si el nombre del archivo es válido y es de lista de precios mover el archivo aceptado a \$MAEDIR/listas empleando la función Mover y grabar en el log el mensaje de éxito

## 5. Rechazar los archivos invalidos

Si se debe rechazar el archivo moverlo a \$RECHDIR empleando la función Mover, grabar en el log el mensaje de rechazo aclarando cual es el motivo:

- Tipo de archivo invalido
- nombre del archivo con formato invalido
- fecha invalida
- asociado inexistente
- colaborador inexistente
- cualquier otro error que considere pertinente indicar.



**6. Una vez que se hayan procesado todos los archivos que existen en \$NOVEDIR**

Se debe chequear la existencia de archivos de listas de precios sin procesar en el directorio \$MAEDIR/listas (ya sean del ciclo actual o de ciclos anteriores).

**7. Si existen archivos de listas de precios sin procesar en \$MAEDIR/listas**

Invocar al Comando Masterlist, siempre que éste o Rating no estén ejecutando.

- Si se pudo Invocar Masterlist, grabar en el log :

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

- Si correspondía invocar pero se debe posponer, grabar en el log:

```
Invocacion de Masterlist pospuesta para el siguiente ciclo
```

- Si arranca correctamente se debe mostrar por pantalla el process id de Masterlist
- Si da algún tipo de error se debe mostrar por pantalla el mensaje explicativo

**8. También se debe chequear la existencia de archivos de listas de compras sin procesar en el directorio \$ACEPDIR**

(ya sean del ciclo actual o de ciclos anteriores).

**9. Si existen archivos de listas de compras sin procesar en \$ACEPDIR**

Invocar al Comando Rating, siempre que éste o Masterlist no estén ejecutando.

- Si se pudo Invocar Rating, grabar en el log :

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

- Si correspondía invocar pero se debe posponer, grabar en el log:

```
Invocacion de Rating pospuesta para el siguiente ciclo
```

- Si arranca correctamente se debe mostrar por pantalla el process id de Rating
- Si da algún tipo de error se debe mostrar por pantalla el mensaje explicativo

**10. Dormir x minutos y Volver al punto 1**



## Actualización de Lista Maestra: Masterlist

### Input

- Archivos de listas de precios sin procesar **\$MAEDIR/listas/super-fecha.usuario**
- Lista Maestra de Precios **\$MAEDIR/precios.mae**
- Maestro de Supermercados **\$MAEDIR/super.mae**

### Output

- Lista Maestra de Precios actualizada **\$MAEDIR/precios.mae**
- Listas de precios procesadas **\$MAEDIR/listas/procesadas/super-fecha.usuario**
- Archivos Rechazados **\$RECHDIR/super-fecha.usuario**
- Log **\$LOGDIR/Masterlist.\$LOGEXT**

### Opciones y Parámetros

- A especificar por el desarrollador

### Descripción

El propósito de este comando es CREAR (la primera vez) o ACTUALIZAR (las veces sucesivas) la lista Maestra de Precios. Las listas de precios vienen en los más diversos formatos, es por ello que este comando reúne la información en un formato unificado y lo deja disponible para el cálculo del presupuesto.

- Puede ser invocado desde la línea de comando o bien desde [Listener](#).
- Graba en el archivo de Log a través del [Logging](#)
- Mueve los archivos a través del [Mover](#)
- No debe procesar dos veces un mismo archivo

### Pasos sugeridos

#### 1. Inicializar el Log

Inicializar el log grabando.

```
Inicio de Masterlist
```

```
Cantidad de Listas de precios a procesar:<cantidad>
```

Los archivos de input se encuentran en **\$MAEDIR**

#### 2. Procesar Un Archivo

Grabar en el log

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

#### 3. Verificar que no sea un archivo duplicado

Cada vez que se procesa un archivo, se lo mueve tal cual fue recibido y con el mismo nombre a **\$MAEDIR/listas/procesadas**

Es por ello que antes de intentar procesar un archivo se debe analizar si ya fue procesado

Si ya fue procesado, rechazar el archivo completo moviéndolo a **\$RECHDIR** empleando la función [Mover](#) y grabar en el log un mensaje aclaratorio:



"Se rechaza el archivo por estar DUPLICADO".

#### 4. Validar el primer registro o registro cabecera

Como la estructura de las listas de precios es variable, el colaborador informa en el primer registro del archivo cual es el **supermercado** y a que **provincia** pertenece, la **cantidad de campos** que tienen los registros de precios, **donde se encuentra el campo producto y el campo precio**, cual es su correo electrónico.

##### Estructura y validaciones a efectuar sobre el registro cabecera

<b>campo1</b>	NOMBRE_SUPER	Caracteres. Combinado con PROVINCIA, debe existir en el Maestro de Supermercados
<b>campo2</b>	PROVINCIA	Caracteres. Combinado con NOMBRE SUPER, debe existir en el Maestro de Supermercados
<b>campo3</b>	CANTIDAD de CAMPOS	Numérico > 1. Representa la cantidad de campos de la lista de precios, o sea, del registro 2 en adelante
<b>Campo4</b>	UBICACIÓN del PRODUCTO	Numérico > 0, <= campo 3 y distinto de campo 4
<b>Campo5</b>	UBICACIÓN del PRECIO	Numérico > 0, <= campo 3 y distinto de campo 4
<b>Campo6</b>	CORREO ELECTRÓNICO DEL COLABORADOR	Caracteres. Combinado con el Usuario, debe existir en el Maestro de Asociados.

Si no se cumple con alguna de las verificaciones precedentes, rechazar el archivo completo moviéndolo a **\$RECHDIR** empleando la función **Mover** y grabar en el log un mensaje aclaratorio, como ser:

```
Se rechaza el archivo por Supermercado inexistente
Se rechaza el archivo por Cantidad de campos invalida
Se rechaza el archivo por Posición producto inválida
Se rechaza el archivo por Posición precio inválida
Se rechaza el archivo por Correo electrónico del colaborador inválido
```

#### 5. Determinar el super\_id

Con el NOMBRE\_SUPER y la PROVINCIA informada en la cabecera, determinar el SUPER\_ID del Maestro de Supermercados

#### 6. Ver si es alta o reemplazo

Con el SUPER\_ID y el USUARIO (que está en el nombre del archivo) buscar si existe algún registro en la lista maestra de precios

##### 6.1 Si no existe ninguno: procesar ALTAS

6.2 Si existe al menos un registro para ese SUPER\_ID + USUARIO y la FECHA (que está en el nombre del archivo) es MAYOR a la FECHA\_ARCHIVO (que está en el registro encontrado), entonces: **procesar REEMPLAZO** (nos vino una lista de precio para el mismo supermercado, del mismo colaborador, pero de una fecha más reciente)

6.3 Si existe al menos un registro para ese SUPER\_ID + USUARIO y la FECHA (que está en el nombre del archivo) es MENOR a la FECHA\_ARCHIVO (que está en el registro encontrado), entonces: **RECHAZAR** el archivo completo moviéndolo a **\$RECHDIR** empleando la función **Mover** y grabar en el log un mensaje aclaratorio, como ser:

```
Se rechaza el archivo por fecha anterior a la existente
```



## 7. Procesar ALTA

Armar un registro de salida a partir del registro de detalle correspondiente excepto que éste tenga algún error, como ser: que alguno de los campos producto o precio no vengan informados (null, espacios, etc) . Si algún registro viene con esta dificultad, saltarlo y continuar con el siguiente. Es responsabilidad del desarrollador prever estas y otras circunstancias de error que pueda tener el registro de input.

Contabilizar la cantidad de “registros de detalle ok” y “registros de detalle nok” para mostrar el resultado en el fin de archivo.

## 8. Procesar REEMPLAZO

- Eliminar TODOS los registros de la lista maestra de precios pertenecientes a la clave

**SUPER\_ID**

**COLABORADOR**

**FECHA ARCHIVO**

- Armar un registro de salida a partir del registro de detalle correspondiente excepto que éste tenga algún error, como ser: que alguno de los campos producto o precio no vengan informados (null, espacios, etc) Si algún registro viene con esta dificultad, saltarlo y continuar con el siguiente. Es responsabilidad del desarrollador prever estas y otras circunstancias de error que pueda tener el registro de input.
- Contabilizar la cantidad de “registros eliminados”, “registros de detalle ok” y “registros de detalle nok” para mostrar el resultado en el fin de archivo.

## 9. Grabar lista maestra de precios

El formato del registro de salida es: SUPER\_ID, COLABORADOR, FECHA nueva ARCHIVO, PRODUCTO, PRECIO

Grabar los registros OK en el archivo \$**MAEDIR/precios.mae**

## 10. Fin de Archivo

- Para evitar el reprocesamiento de un mismo archivo, mover el archivo procesado a: **\$MAEDIR/listas/procesadas** empleando la función **Mover**.
- Grabar en el Log la cantidad de “registros eliminados” (cuando es modificación), “registros ok” y “registros nok”

NOTA: en tiempos de testing verifique que la cantidad de Registros grabados en ok + la cantidad de Registros grabados en nok coincida con el total de registros del archivo - 1

## 11. Repetir hasta que se terminen todos los archivos de listas de precios

## 12. Cerrar el Log

“Fin de **Masterlist**”





## Cálculo de Presupuesto: Rating

### Input

- Archivos de listas de compras “aceptadas” \$**ACEPDIR**/usuario.xxx
- Lista maestra de Precios \$**MAEDIR**/precios.mae
- Tabla de Equivalencias de unidades de medida \$**MAEDIR**/um.tab

### Output

- Archivos de listas “presupuestadas” \$**INFODIR**/listas/usuario.xxx
- Archivos de listas de compras “procesadas” \$**ACEPDIR**/procesadas/usuario.xxx
- Archivos rechazados \$**RECHDIR**/usuario.xxx
- Log \$**LOGDIR**/Rating.\$**LOGEXT**

### Opciones y Parámetros

- A especificar por el desarrollador

### Descripción

El propósito de este comando es leer los archivos que se encuentran en el directorio \$**ACEPDIR**, que contienen de listas de compras “aceptadas” y grabar las listas presupuestadas, según se vayan detectando coincidencias con el archivo maestro de precios.

- Es el tercero en orden de ejecución
- Se dispara automáticamente
- Graba en el archivo de Log a través del Logging
- Mueve los archivos a través del Mover
- No debe procesar dos veces un mismo archivo

### Pasos sugeridos

#### 1. Inicializar el Log

Inicializar el log grabando:

```
Inicio de Rating
```

```
Cantidad de Listas de compras a procesar:<cantidad>
```

Los archivos de input se encuentran en \$**ACEPDIR**

#### 2. Procesar Un Archivo

Grabar en el log

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

#### 3. Verificar que no sea un archivo duplicado

Cada vez que se procesa un archivo, se lo mueve tal cual fue recibido y con el mismo nombre a \$**ACEPDIR**/procesadas

Es por ello que antes de intentar procesar un archivo se debe analizar si ya fue procesado

Si ya fue procesado, rechazar el archivo completo y grabar en el log un mensaje aclaratorio, como ser:



"Se rechaza el archivo por estar DUPLICADO".

El archivo duplicado se lo mueve a \$**RECHDIR** empleando la función **Mover**.

#### 4. Realizar otras validaciones a nivel de archivo si lo considera necesario

Es responsabilidad del desarrollador prever otras circunstancias de error, como ser que el formato del registro no se corresponde con lo esperado, que el archivo esté vacío, etc.

Si ello ocurre, rechazar el archivo completo y grabar en el log un mensaje aclaratorio, como ser:

"Se rechaza el archivo por formato invalido".

"Se rechaza el archivo por estar VACIO".

El archivo rechazado se lo mueve a \$**RECHDIR** empleando la función **Mover**.

#### 5. Por cada Registro del Archivo

##### 5.1 Obtener los distintos precios del producto de la lista maestra de precios

El registro de input posee el siguiente formato: NRO de ITEM, PRODUCTO PEDIDO

Efectuar la comparación (sin tener en cuenta mayúsculas y minúsculas) "por palabra" entre el "PRODUCTO PEDIDO" de la lista de compras y el "PRODUCTO" de la lista maestra de precios.

Se considera "palabra" a:

- Cualquier cadena de caracteres al inicio del campo HASTA el fin del campo o espacio
- Cualquier cadena de caracteres entre espacios (puede haber mas de un espacio entre "palabras")
- Cualquier cadena de caracteres al final del campo PRECEDIDA por espacio

Ejemplos:

Productos de la lista de compras: "Azucar chango 1 kg." "Caldo MAGGI 114 grm" "Fideos soperos semolados MATARAZZO pamperito 3 vegetales 500 gr" "Aceite Girasol Cocinero 1500 CC"									Cantidad de Palabras
Azucar	chango	1	Kg.						4
Caldo	MAGGI	114	grm						4
Fideos	soperos	semolados	MATARAZZO	pamperito	3	vegetales	500	gr	9
Aceite	Girasol	Cocinero	1500	CC					5

Productos de la lista de precios: "Azucar chango 1 kg" "Caldo MAGGI verdura cja 114 grm" "Fideos soperos sem. MATARAZZO pamperito 3 vegetales paq 500 grm" "COCINERO ACEITE DE GIRASOL BOTELLA DE PLASTICO - 1500 CM3"										Cantidad de Palabras
Azucar	chango	1	kg							4
Caldo	MAGGI	pet	250	grm						5
Fideos	soperos	sem.	MATARAZZO	pamperito	3	vegetales	paq	500	grm	10
COCINERO	ACEITE	DE	GIRASOL	BOTELLA	DE	PLASTICO	-	1500	CM3	10

#### Cuando hay "coincidencia"?

Decimos que hay "coincidencia" (encontramos el o los precios) cuando, para un producto pedido de n palabras:



- las primeras n-1 palabras coinciden con las palabras del producto de la lista de precios (no importa el orden)
- La última palabra (que siempre es la unidad de medida) coincide con la unidad de medida del producto de la lista de precios o bien existe "equivalencia". Para determinar las "equivalencias" entre unidades de medida usar la tabla **MAEDIR/um.tab**. Allí encontraremos todas las equivalencias entre una misma unidad de medida. Por ejemplo si en la lista de compras dice CM3 y en la de precios CC, estamos en presencia de la MISMA unidad de medida.

Si consideramos los ejemplos de arriba

Productos de la lista de compras	COINCIDE	Productos de la lista maestra de precios
"Azucar chango 1 kg."	SI	"Azucar chango 1 kg"
"Caldo MAGGI 114 grm"	SI	"Caldo MAGGI verdura cja 114 grm"
"Fideos soperos <b>semolados</b> MATARAZZO pamperito 3 vegetales 500 gr"	NO	"Fideos soperos <b>sem.</b> MATARAZZO pamperito 3 vegetales paq 500 grm"
"Aceite Girasol Cocinero 1500 CC"	SI	"COCINERO ACEITE DE GIRASOL BOTELLA DE PLASTICO - 1500 CM3"

Si extendemos nuestro ejemplo con los siguientes productos de la lista maestra de precios:

Caldo de carne maggi 114 gr 8.20  
Caldo de gallina maggi 114 gr 7.99  
Caldo de verdura maggi 114 gr 8.50

Caldo MAGGI carne cja 114 grm 9.03

MAGGI CALDO DE GALLINA - 114 G 7.99

Todos ellos también coinciden con "Caldo MAGGI 114 grm" y corresponde grabarlos en la lista presupuestada.

## 5.2 Si se encuentran coincidencias, grabarlas en lista presupuestada

Si se encuentran coincidencias, agregar tantos registros como "coincidencias de producto" detectadas en **\$INFODIR/listas/usuario.xxx**

Los campos a grabar son: NRO de ITEM, PRODUCTO PEDIDO, SUPER\_ID, PRODUCTO ENCONTRADO y PRECIO.

## 5.3 Si no se encuentra coincidencia, no hay precio, igual grabar con precio vacio

Si NO se encuentran coincidencias, agregar un solo registro en **\$INFODIR/listas/usuario.xxx** con NRO de ITEM, PRODUCTO PEDIDO y el resto del registro con los campos vacios (SUPER\_ID, PRODUCTO ENCONTRADO y PRECIO).

## 6. Repetir hasta que se termine el archivo.

## 7. Fin de Archivo

Para evitar el reprocesamiento de un mismo archivo, mover el archivo procesado a: **\$ACEPDIR/procesadas** empleando la función **Mover**.

## 8. Repetir hasta que se terminen todos los archivos.

## 9. Cerrar el Log

Cerrar el log grabando "Fin de Rating"



---

## **Impresión de Informes: Reporting**

### **Input**

- Archivos de Listas de Compras presupuestadas \$**INFODIR**/listas/usuario.xxx
- Maestro de Supermercados \$MAEDIR/**super.mae**

### **Output**

- Informes \$**INFODIR**/info.xxx

### **Opciones y Parámetros**

- -a (ayuda)
- -g (grabar)
- -p (precio de referencia) (combinable con m y d)
- -m (menor precio)
- -d (donde comprar)
- -f (faltante)
- -x (filtrar por provincia-supermercado)
- -u (filtrar por usuario)
- Otros parámetros a especificar por el desarrollador

### **Descripción**

Este proceso nos permite realizar consultas sobre las listas presupuestadas

- Es el cuarto en orden de ejecución
- Se dispara manualmente
- No graba en el archivo de log
- Reporting No debe ejecutar si la inicialización de ambiente no fue realizada
- Reporting No debe ejecutar si ya existe otro comando Reporting en ejecución

### **Requisitos:**

- Se deben emplear estructuras Hash en la resolución (requisito indispensable)
- Debe presentar un menú amigable y una **opción (-a)** de ayuda del comando
- Debe permitir al usuario efectuar N consultas sin salir del comando
- Debe mostrar los resultados por pantalla indicando en el encabezado los datos de la consulta, luego el filename, luego el detalle. Si cambia el archivo mostrar un renglón en blanco y seguir con el siguiente.
- Debe permitir grabar un reporte de salida con lo mismo que se muestra por pantalla (**opción -g**). El nombre del reporte debe ser \$/INFODIR/info.xxx dónde xxx es un descriptor siempre distinto que asegura no sobrescribir ningún informe previo.
- Para el caso del filtro -x, ofrecer un menú indicando provincias-supermercados disponibles (con los nombres, no los id). El usuario puede elegir todos, uno o algunos. Excluir de este menú los supermercados "Precios Cuidados".
- En el contexto de este comando llamaremos "precio de referencia" a aquellos correspondientes a los Precios Cuidados. Otra forma de identificar los precios cuidados es



mediante su id de supermercado: todos los SUPER\_ID < 100 corresponden a precios cuidados.

En todos los casos el input son las listas presupuestadas. Si no se pone ningún filtro, usar todas las listas presupuestadas para armar el informe. Si para los filtros+opciones indicadas, no hay detalles, mostrar (o grabar) un mensaje informativo.

### Opción g: opción de grabación

Esta opción es combinable con todas las opciones

Cuando se indica g, se debe grabar INFODIR/info.xxx

Si no se indica g, solo mostrar por pantalla.

**Primera línea para todos los casos:** condiciones de invocación (opciones y filtros)

**Línea de leyenda** (solo para las opciones -mp o -dp) Mostrar la siguiente leyenda “(\*) Precio menor o igual al PC. (\*\*) Precio mayor al PC. (\*\*\*) PC no encontrado.”

**Lista Presupuestada:** filename de la lista presupuestada, cada vez que cambia lista, mostrar el nuevo nombre

Líneas de detalle cuando la opción es -mp o -dp

Cuando la opción es -mp o -dp	Descripción/Fuente/Valor
<b>NOMBRE_SUPER-PROVINCIA</b>	Se obtiene del Maestro de Supermercados concatenando con un guion nombre y provincia
<b>NRO de ITEM</b>	Se obtiene de la lista presupuestada
<b>PRODUCTO PEDIDO</b>	Se obtiene de la lista presupuestada
<b>PRODUCTO ENCONTRADO</b>	Se obtiene de la lista presupuestada
<b>PRECIO</b>	Se obtiene de la lista presupuestada
<b>PRECIO de REFERENCIA</b>	Precio de referencia para ese ítem o leyenda “no encontrado
<b>Observaciones</b>	Uno, dos o tres asteriscos encerrados entre () dependiendo del Resultado de la comparación entre el precio y el precio de referencia

Líneas de detalle cuando la opción es p m d f

Cuando la opción es p m d f	Descripción/Fuente/Valor
<b>NOMBRE_SUPER-PROVINCIA</b>	Se obtiene del Maestro de Supermercados concatenando con un guion nombre y provincia
<b>NRO de ITEM</b>	Se obtiene de la lista presupuestada
<b>PRODUCTO PEDIDO</b>	Se obtiene de la lista presupuestada
<b>PRODUCTO ENCONTRADO</b>	Se obtiene de la lista presupuestada
<b>PRECIO</b>	Se obtiene de la lista presupuestada

Lista presupuestada ejemplo para la explicación del comando				
NRO de ITEM	PRODUCTO PEDIDO	SUPER_ID	PRODUCTO ENCONTRADO	PRECIO
1	Azucar chango 1 kg.	1	CHANGO AZUCAR BLANCA - 1 KG	7.59
1	Azucar chango 1 kg.	112	Azucar CHANGO est 1 kgm	12.19
1	Azucar chango 1 kg.	112	Azucar CHANGO organica est 1 kg	19.89
1	Azucar chango 1 kg.	134	Azucar chango 1 kg	7.59
1	Azucar chango 1 kg.	134	Azucar tetra chango 1 kg	10.66
2	Caldo MAGGI 114 grm	1	MAGGI CALDO DE GALLINA - 114 G	7.99
2	Caldo MAGGI 114 grm	112	Caldo MAGGI carne cja 114 grm	9.03
2	Caldo MAGGI 114 grm	112	Caldo MAGGI verdura cja 114 grm	9.03



2	Caldo MAGGI 114 grm	134	Caldo de carne maggi 114 gr	8.20
2	Caldo MAGGI 114 grm	134	Caldo de gallina maggi 114 gr	7.99
2	Caldo MAGGI 114 grm	134	Caldo de verdura maggi 114 gr	8.50
3	Fideos soperos semolados MATARAZZO pamperito 3 vegetales 500 gr			
4	Aceite Girasol Cocinero 1500 CC	1	COCINERO ACEITE DE GIRASOL BOTELLA DE PLASTICO - 1500 CM3	12.02
5	Aceto balsamico CASALTA manzana 400 cc	112	Aceto balsamico CASALTA manzana bot 400 cc	30.39
5	Aceto balsamico CASALTA manzana 400 cc	134	Aceto balsamico manzana casalta 400 cc	31.50

#### Opción -p: precio de referencia

Para cada ítem de la lista presupuestada, elegir solo los precios de referencia y mostrarlos ordenados por ítem. Para este caso, si tomamos la lista presupuestada ejemplo, los registros que se usarían en el informe son:

1	Azucar chango 1 kg.	1	CHANGO AZUCAR BLANCA - 1 KG	7.59
2	Caldo MAGGI 114 grm	1	MAGGI CALDO DE GALLINA - 114 G	7.99
4	Aceite Girasol Cocinero 1500 CC	1	COCINERO ACEITE DE GIRASOL BOTELLA DE PLASTICO - 1500 CM3	12.02

#### Opción -f: precio faltante

Para cada ítem de la lista presupuestada, elegir solo ítems sin precio. Para este caso, si tomamos la lista presupuestada ejemplo, los registros que se usarían en el informe son:

3	Fideos soperos semolados MATARAZZO pamperito 3 vegetales 500 gr			
---	---	--	--	--

#### Opción -m: menor precio

Para cada ítem de la lista presupuestada, informar el de menor precio que no sea un precio de referencia y mostrarlos ordenados por ítem. Para este caso, si tomamos la lista presupuestada ejemplo, los registros que se usarían en el informe son:

1	Azucar chango 1 kg.	134	Azucar chango 1 kg	7.59
2	Caldo MAGGI 114 grm	134	Caldo de gallina maggi 114 gr	7.99
5	Aceto balsamico CASALTA manzana 400 cc	112	Aceto balsamico CASALTA manzana bot 400 cc	30.39

#### Opción -d: donde comprar

Idem m, pero el listado debe mostrarse agrupado por lugar de compra, cuando cambia el lugar de compra, dejar una línea en blanco.

#### Opción -mp: menor precio + precio de referencia

Idem m pero si existe precio de referencia, siempre informarlo aunque no sea el menor.

En observaciones indicar:

Un \*: cuando el precio es menor o igual al precio de referencia

Dos \*: cuando el precio es mayor al precio de referencia

Tres \*. Cuando el precio de referencia no fue encontrado.

Para este caso, si tomamos la lista presupuestada ejemplo, los registros que se usarían en el informe son:

1	Azucar chango 1 kg.	1	CHANGO AZUCAR BLANCA - 1 KG	7.59
1	Azucar chango 1 kg.	134	Azucar chango 1 kg	7.59
2	Caldo MAGGI 114 grm	1	MAGGI CALDO DE GALLINA - 114 G	7.99
2	Caldo MAGGI 114 grm	134	Caldo de gallina maggi 114 gr	7.99
5	Aceto balsamico CASALTA manzana 400 cc	112	Aceto balsamico CASALTA manzana bot 400 cc	30.39

#### Opción -dp: donde comprar

Idem d, pero si existe precio de referencia, siempre informarlo aunque no sea el menor



---

FIN
-----