|  |  |  |
| --- | --- | --- |
| Proyecto  Primer Trabajo Practico Sistemas Operativos 75:08 | Grupo  N.º 2 | Fecha de Vencimiento  16 de junio de 2020 18 horas |
| **Integrantes: Padrón, Nombre y Apellido** | **Correo Electrónico** | |
| 1.  2.  3.  4. 96029 - Facundo Walter | -  -  -  fwalter@fi.uba.ar | |

|  |
| --- |
| Instrucciones  En este Word se encuentra el enunciado del TP  En el Excel se encuentra   * Estructuras * los datos para armar las tablas de Códigos de comercios, Códigos de Provincias y Códigos de respuesta Gateway * los datos para armar los lotes de prueba   Armen los archivos con las tablas de códigos y los casos de prueba respetando la estructura indicada  Descripción  Se reciben archivos con información de transacciones efectuadas con tarjetas de crédito desde varios comercios. Cada comercio puede enviar uno o mas archivos. Cada archivo puede contener una o mas transacciones.  El objetivo del TP es generar un archivo financiero por comercio por día, hasta la **“hora de cierre”**. Si la hora de la transacción (campo "isO12\_cLocalTransactionTime") es posterior de la **“hora de cierre”**, la transacción pasa al archivo financiero del día siguiente.  Todo el sistema debe poder ser ejecutado SIN SER ROOT  Trabajo a Realizar   1. Se debe crear un paquete con todas las componentes del sistema MAS un archivo README por afuera del paquete donde se explica que pasos se deben dar para descargar, instalar y ejecutar el sistema. 2. Se debe realizar un script de instalación, uno de inicialización y un proceso que lee el input disponible, crea y/o cierra los archivos financieros, duerme un tiempo x y vuelve a empezar. Dada esta característica del proceso, se deben proveer comandos complementarios para arrancar y detener el proceso. Todos los comandos deben desarrollarse en Shell script 3. Se debe lograr la ejecución de todo el sistema y dar evidencia de ello 4. Se deben generar archivos de log y todos ellos respetar el mismo formato 5. Se debe realizar y entregar la autoevaluación 6. Se debe entregar el TP antes del 16 de junio a las 18 horas    1. Primer paso: Toda la entrega debe estar subida a un repositorio (como GitHub) y dar acceso a so7508@gmail.com    2. Segundo paso: Enviar mail a [so7508@gmail.com](mailto:so7508@gmail.com) con el link y adjuntar el archivo README y este documento completo. |

Contenido

[**Punto 1.** Entrega del TP 2](#_Toc40801903)

[**Punto 2.** README 2](#_Toc40801904)

[**Punto 3.** Autoevaluación 2](#_Toc40801905)

[**Punto 4.** Documentación 3](#_Toc40801906)

[**Punto 5.** Descarga del TP 3](#_Toc40801907)

[**Punto 6.** Script instalador – evidencia de instalación 4](#_Toc40801908)

[**Punto 7.** Script Inicializador – Evidencia de la Inicialización 7](#_Toc40801909)

[**Punto 8.** Proceso 9](#_Toc40801910)

[**Punto 9.** Arranque y Detención 15](#_Toc40801911)

[**Punto 10.** Pruebas 15](#_Toc40801912)

|  |  |
| --- | --- |
| Enunciado y Planilla de Evaluación | SI – NO – NA |
| Entrega del TP | |
| 1. Mail enviado a [so7508@gmail.com](mailto:so7508@gmail.com) antes del 16/06/2020 18 hs |  |
| 1. Mail conteniendo link al repositorio. |  |
| 1. Mail conteniendo README, según lo indicado en el **punto 2** |  |
| 1. Mail conteniendo autoevaluación según lo indicado en el **punto 3** |  |
| 1. Repositorio accesible conteniendo documentación del TP según lo indicado en el **punto 4** y cualquier otro ítem explicativo de la resolución y/o la prueba que quiera agregar (observaciones, recomendaciones, comentarios) |  |
| 1. Repositorio accesible conteniendo la resolución el TP. La descarga debe estar resuelta según lo indicado en el **punto 5** |  |
| 1. Script instalador y evidencia de la instalación según lo indicado en el **punto 6** (incluir log y archivo de configuración en la entrega) |  |
| 1. Script inicializador y evidencia de la inicialización según lo indicado en el **punto 7** (incluir log de inicialización) |  |
| 1. Script proceso y evidencia de la ejecución según lo indicado en el **punto 8** (incluir log del proceso y archivos de prueba) |  |
| 1. Script de arranque y detención, evidencia de su funcionamiento según lo indicado en el **punto 9** |  |

|  |  |
| --- | --- |
| README | |
| **Archivo README** en el cual se explican todos y cada uno de los pasos necesarios para acceder al TP, descargarlo, instalarlo y ejecutarlo  Esto incluye instrucciones detalladas desde como loguearse, hasta como ejecutar cada comando | |
| 1. ¿El README brinda las instrucciones de descarga y las instrucciones para ejecutar el instalador? |  |
| 1. ¿El README explica que deja la instalación y dónde? ¿El README brinda las instrucciones para realizar una reparación de la instalación? |  |
| 1. ¿El README brinda las instrucciones para ejecutar el inicializador? ¿El README brinda las instrucciones para detener o arrancar el proceso? |  |
| 1. ¿El README brinda las instrucciones para efectuar una prueba completa? |  |
| 1. ¿El README brinda las instrucciones para modificar la hora de cierre? |  |

|  |  |
| --- | --- |
| Autoevaluación | |
| Es este documento completo con:   1. Número de grupo y listado de integrantes que efectivamente han participado en la resolución del TP en la primera hoja |  |
| 1. Columna derecha de autoevaluación completa indicando  * SI, cuando la entrega cumple con lo solicitado * NO, si no cumple con lo solicitado o no puede dar evidencia de ello * NA si el ítem no aplica a su solución |  |

|  |  |
| --- | --- |
| Documentación | |
| Ud. puede realizar la documentación en este mismo documento o bien en un documento aparte, lo que debe documentar es: | |
| 1. Hipotesis y Aclaraciones   En un ítem bajo este título describa todas las hipótesis y aclaraciones que han asumido en la resolución del TP.   * Por ejemplo, en una parte del enunciado se indica que se debe establecer un mecanismo para modificar la **“hora de cierre”**. Este mecanismo lo define el grupo, pero debe ser explicado aquí que decidieron. Es un plus si se hace a través de un comando |  |
| 1. Descripción de Problemas   En un ítem bajo este título describa las dificultades que atravesaron durante el desarrollo del TP y como lograron resolverlas. |  |
| 1. Comandos Adicionales ¿ha creado nuevos comandos?   Si crea algún comando auxiliar, indique su nombre y para que lo usa. Si no realiza comandos auxiliares indique “NA”  Ejemplo: si crea un comando aparte para ver en forma amigable el archivo de log, entonces debe indicarlo. |  |
| 1. Archivos Adicionales ¿ha creado nuevos archivos permanentes?   Si crea algún archivo auxiliar, indique su nombre y para que lo usa.  Si no crea nuevos archivos, indique “NA” |  |
| 1. Diagrama de la Estructura   En un ítem bajo este título de un ejemplo de la estructura resultante luego de una instalación estándar.  En la rama que contiene los comandos, liste los nombres de los comandos  En la rama que contiene las tablas de códigos, liste los nombres de los archivos  Indique cual es el nombre donde resguarda los archivos originales [[1]](#footnote-2)  En la rama que contiene los archivos de prueba, liste los nombres de los archivos (los archivos de prueba no deben grabarse en el directorio de novedades, se deben resguardar hasta su uso) |  |

|  |  |
| --- | --- |
| Descarga del TP | |
| DIRECTORIO DE TRABAJO  Toda la descarga debe realizarse bajo un mismo directorio de trabajo llamado Grupoxx donde xx es su número de grupo  Todo el camino (path) que va desde la raíz hasta Grupoxx lo denominaremos genéricamente $GRUPO para facilitar la explicación del TP | |
| 1. ¿está documentado como realizar la descarga? |  |
| 1. Al descargar el paquete, ¿se crea un directorio de trabajo denominado grupoxx y recién a partir de allí se graban todos los archivos que conforman la entrega?   ¿Porque es esto? Para evitar que las distintas resoluciones de cada grupo se mezclan entre ellas. |  |
| 1. Al descargar el paquete, ¿bajan todos los comandos y archivos, o alguno quedó fuera de la descarga? |  |
| 1. Al descargar el paquete, ¿se reservan los scripts y archivos originales en algún subdirectorio? |  |

|  |  |
| --- | --- |
| Script instalador – evidencia de instalación | |
| SUB DIRECTORIO DE CONFIGURACIÓN  El directorio $GRUPO debe contener un subdirectorio /inst en donde se copia el script de instalación  Este nombre es reservado, ningún otro directorio /inst puede ser generado dentro de $GRUPO  Allí también se depositarán el log de la instalación y el archivo de configuración | |
| Propósito | |
| El propósito de este comando es realizar la instalación del sistema | |
| El comando debe soportar los siguientes escenarios | |
| * El sistema nunca fue instalado (se deberá instalar el sistema) * El sistema está bien instalado (no se deberá instalar el sistema, solo mostrar datos de la instalación) * El sistema está instalado, pero está incompleto (se deberá reparar la instalación siempre que se pueda): Invoque la reparación con la opción -r | |
| Configurar los nombres de los directorios. | |
| Solicitar al usuario que defina   * 1. El directorio de ejecutables (por default proponer el directorio /bin)   2. El directorio de tablas del sistema (por default proponer el directorio /tab)   3. El directorio de arribos, es decir, el directorio donde se depositan temporalmente los archivos de novedades (por default proponer el directorio /nov)   4. El directorio de aceptados, es decir, el directorio donde se depositan temporalmente los archivos de novedades que han sido aceptados, (por default proponer el directorio /ok)   5. El directorio de rechazados, es decir, el directorio donde se depositan los archivos que han sido rechazados, (por default proponer el directorio /nok   6. El directorio de procesados, es decir, el directorio donde se depositan los archivos que han sido procesados, (por default proponer el directorio /proc)   7. El directorio de salida, es decir, el directorio donde se depositan los archivos que genera el sistema durante el procesamiento, (por default proponer el directorio /sal) | |
| Evitar nombres de directorios duplicados (dos directorios con igual nombre)  Evitar el nombre de directorio /inst (se considera nombre reservado)  Evitar cualquier otro nombre de directorio que el grupo haya considerado reservado, por ejemplo, el que usará para salvaguardar el paquete original (completo) y reparar el sistema.  Si el usuario ingresa un directorio incorrecto, indicarle al usuario el error y volver a mostrar el valor default para que acepte ese valor o ingrese uno nuevo | |
| Configurar la hora de cierre. | |
| Solicitar al usuario que defina   1. La hora de cierre, es decir, el parámetro que usará el proceso para determinar la hora de cierre del archivo financiero. Por default proponer el valor 180000 (formato hhmmss)   Debe ser una hora válida, si no lo es, indicarle al usuario el error y volver a mostrar el valor default para que acepte ese valor o ingrese uno nuevo | |
| En todos los casos, cuando se Interactúa con el usuario se debe proponer un valor por default | |
| Confirmación de la instalación | |
| Cuando el usuario termina de configurar los nombres de directorios y la hora de cierre, mostrarle por pantalla donde se encuentra el instalador, como va a quedar la estructura, la hora de cierre y solicitarle que confirme para continuar | |
| *Ejemplo:*  TP SO7508 1º Cuatrimestre 2020. Copyright © Grupo xx  Directorio padre: $grupo  Script Instalador: $grupo/inst/instalador.sh  Log de la instalación: $grupo/inst/instalador.log  Configuración de la instalación: $grupo/inst/instalador.conf  Directorio de ejecutables: $grupo/bin  Directorio de tablas: $grupo/tab  Directorio de novedades: $grupo/nov  Directorio de aceptados: $grupo/ok  Directorio de rechazados: $grupo/nok  Directorio de procesados: $grupo/proc  Directorio de salidas: $grupo/sal  Hora de Cierre: 180000  Estado de la instalación: LISTA  ¿Confirma la instalación? (SI-NO): \_ | |
| No debe crear ningún directorio configurable por el usuario hasta no tener la confirmación de la instalación | |
| Si el usuario indica **No**, volver a solicitar los directorios, pero esta vez mostrando como default lo recientemente ingresado, en lugar de los propuestos inicialmente (es decir, que el script debe tener memoria de los valores ingresados previamente)  Si el usuario indica **Si** crear la estructura de directorios definida y mover cada archivo a su lugar correspondiente | |
| Grabar archivo de configuración | |
| |  |  | | --- | --- | | Configuración de la instalación: $grupo/inst/instalador.conf | | | Separador de campos: - (guion medio) | | | Campo | Descripción/fuente/valor | | Identificador | Identificador (nombre corto usado luego como variable de ambiente), por ejemplo: GRUPO HCIERRE DIRBIN DIRNOV DIRPROC | | Valor | Valor ingresado para la ruta del directorio / valor ingresado para la hora de cierre |  * Para el directorio $grupo crear un registro con identificador GRUPO * Para el directorio /inst crear un registro con identificador DIRINST * Para el Directorio de ejecutables crear un registro con identificador DIRBIN * Para el Directorio de tablas crear un registro con identificador DIRTAB * Para el Directorio de novedades crear un registro con identificador DIRNOV * Para el Directorio de aceptados crear un registro con identificador DIROK * Para el Directorio de rechazados crear un registro con identificador DIRNOK * Para el Directorio de procesados crear un registro con identificador DIRPROC * Para el Directorio de salidas crear un registro con identificador DIRSAL * Para la Hora de Cierre crear un registro con identificador HCIERRE   Puede crear más registros si lo considera necesario  Este archivo luego debe ser leído por el script inicializador para configurar las variables de ambiente | |
| Log de la Instalación | |
| Todo lo que se muestra al usuario por pantalla y sus respuestas debe ser registrado en el log[[2]](#footnote-3) de la instalación  El log de la instalación no debe reescribirse, aun cuando se produce cancelación del script o reinstalación.  A medida que se hace la creación de directorios y su llenado informarlo mediante mensajes (por pantalla y en el log de la instalación) | |
| |  | | --- | | Log de la instalación: $grupo/inst/instalador.log |  |  |  | | --- | --- | | Separador de campos: - (guion medio) | | | Campo | Descripción/fuente/valor | | Fecha | Formato a Elección. Es la fecha y hora en el momento de grabación del registro. | | Tipo de Error | Indica que tipo de error es: Informativo (INF), Alerta (ALE) Error (ERR) | | Mensaje | Mensaje de log propiamente dicho | | Origen | Es el nombre del Comando, función o rutina que produce el evento que se registra en el log | | Usuario | Es el login del usuario que graba el registro |   ESTE FORMATO DEBE SER IGUAL PARA TODO EL SISTEMA | |
| 1. ¿Hay Indicación correcta en el README de cómo se ejecuta este script? ¿se explica en la documentación cual es el mecanismo para saber si el sistema está o no instalado? |  |
| 1. ¿el instalador permite Configurar los nombres de los directorios y la hora de cierre? |  |
| 1. ¿el instalador evita nombres duplicados? Ejemplo: /misarchivos (para el de ejecutables) /misarchivos (para las novedades) ¿evita /inst (valor reservado)? 2. ¿permite nombres de directorios con espacios intermedios (Ejemplo: /mis novedades) 3. ¿Valida la Hora de Cierre? |  |
| 1. ¿Funciona la confirmación de la instalación?   ¿Confirma la instalación? (SI-NO): \_ |  |
| 1. Si la instalación no se confirma ¿se reinicia el proceso? ¿tiene memoria de lo que fue cargando previamente el usuario? ¿muestra correctamente los defaults? |  |
| 1. Si la instalación se confirma ¿crea los directorios en ese momento y mueve los archivos a su lugar? |  |
| 1. Si la instalación se confirma ¿crea el archivo de configuración con el contenido solicitado? 2. ¿respeta el formato de registro indicado? |  |
| 1. Durante toda la ejecución, ¿graba en el log? 2. ¿se registra toda la interacción entre el usuario y el script tanto cuando la instalación es exitosa como cuando la instalación es cancelada o reparada? 3. ¿conserva el log luego de una reinstalación o cancelación? 4. ¿respeta el formato de registro indicado? |  |
| 1. ¿el instalador detecta cuando el sistema está bien instalado y no lo instala? |  |
| 1. ¿el instalador detecta cuando el sistema se debe reparar 2. y lo repara?   Nota: la reparación puede ser implementada en el inicializador, documente en las hipotesis donde decide resolver la reparación. |  |

|  |  |
| --- | --- |
| Script Inicializador – Evidencia de la Inicialización | |
| Propósito | |
| El propósito de este comando es preparar el entorno de ejecución del TP (variables de ambiente) y asegurar que estén dadas todas las condiciones para la ejecución del sistema e invocar al proceso  Lo más importante en este script es su forma de invocación, la cual debe asegurar que cuando invoca al proceso no se generen sub-shells, si esto ocurre el valor de las variables de ambiente pierden visibilidad en el proceso.  Siempre los pasos de la explicación son a modo indicativo, se puede alterar el orden siempre y cuando no se altere el resultado final esperado  IMPORTANTE  Todo el sistema debe poder ser ejecutado | |
| **El comando debe soportar los siguientes escenarios** | |
| * El sistema nunca fue inicializado (se deberá **inicializar el sistema**) * El sistema ya fue inicializado exitosamente (no se deberá reinicializar) * El sistema no se puede inicializar porque falta algún componente (se deberá explicar el motivo y como reparar la instalación) | |
| Verificar directorios | |
| Leer el archivo de configuración y verificar que existen los directorios  Si falta algún directorio:  Opción a) mostrar por pantalla / grabar (en el log) un mensaje explicativo de la situación e indicar cómo hacer para reparar la instalación. Terminar el script  Opción b) mostrar por pantalla / grabar (en el log) un mensaje explicativo de la situación y reparar la instalación. Continuar el script | |
| Verificar archivos | |
| Verificar que las tres tablas de códigos y todos los comandos existan  Si falta algún archivo  Opción a) mostrar por pantalla / grabar (en el log) un mensaje explicativo de la situación e indicar cómo hacer para reparar la instalación. Terminar el script  Opción b) mostrar por pantalla / grabar (en el log) un mensaje explicativo de la situación y reparar la instalación. Continuar el script | |
| Verificar permisos | |
| Verificar que las tres tablas de códigos tengan permiso de lectura y los comandos permisos de ejecución.  Si falta algún permiso corregirlo | |
| Setear variables de ambiente | |
| Todos los identificadores del archivo de configuración deben convertirse en variables de ambiente | |
| Estas variables deben permanecer durante toda la ejecución del sistema y ser accesibles por el proceso u otros comandos relacionados  El único que lee el archivo de configuración es este script, el resto de los comandos trabajan con las variables de ambiente directamente. | |
| Arrancar el proceso | |
| Invocar al script PROCESO  ADVERTENCIA: no invocar el proceso si ya hay uno corriendo. Avisar cuando pasa eso | |
| Mostrar el process id y la hora de cierre | |
| Mostrar y grabar en el log el process id y brindar una explicación acerca de como detener / arrancar el proceso manualmente.  Mostrar y grabar en el log la hora de cierre y brindar una explicación acerca de cómo se debe hacer para modificar este parámetro  NOTA: ambas explicaciones deben estar documentadas en el README | |
| Grabar log mientras se ejecuta el script | |
| Todo lo que va sucediendo durante la inicialización se debe grabar en el log y mostrarse por pantalla  El nombre del log debe ser igual al nombre del comando con extensión .log  El directorio donde se graba el log del inicializador es DIRPROC  El log no debe reescribirse, debe conservar la historia | |
| 1. ¿Hay documentación completa del script?   Indicación en el README de cómo se ejecuta este script  Documentación sobre cómo se resuelve la reparación de la instalación  Documentación sobre cómo se resuelve la modificación de la hora de cierre |  |
| 1. ¿el comando detecta cuando el sistema nunca fue inicializado y lo inicializa? |  |
| 1. ¿detecta cuando el sistema ya fue inicializado exitosamente y no reinicializa? |  |
| 1. Si falta algún directorio, archivo, ¿se resuelve en el script? 2. Si falta algún directorio o archivos, ¿se explica cómo reparar la instalación? |  |
| 1. Si falta algún permiso, ¿los corrige? |  |
| 1. ¿setea las variables de ambiente y estas se exportan al siguiente script? |  |
| 1. ¿arranca el proceso? ¿Se muestra el process id? |  |
| 1. En el log, ¿hay líneas del progreso de la inicialización? 2. ¿respeta el formato del registro del log?   Ejemplo: “la verificación de directorios es exitosa” “corrección de permisos efectuada” |  |
| 1. ¿En el log, graba el pid cuando arranca el proceso? ESTO ES MUY IMPORTANTE PARA REALIZAR LOS TESTEOS |  |

|  |  |
| --- | --- |
| Proceso | |
| Propósito | |
| Leer las novedades y generar los archivos de salida (archivos financieros) | |
| Tipo de proceso | |
| Este comando es un proceso del tipo demonio, daemon o dæmon (de sus siglas en inglés Disk And Execution Monitor).  Se ejecuta en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo).  Cada vez que termina su trabajo duerme un tiempo “x” y vuelve a empezar  Si itera mas de 10000 veces, terminar el programa  Se debe dar una herramienta (comando auxiliar) para detener el proceso manualmente antes que llegue a 10000 ciclos  Si se detiene manualmente, debe existir un comando auxiliar para arrancarlo o bien brindar indicaciones claras acerca de cómo arrancar de nuevo el proceso usando el inicializador | |
| Los pasos enumerados a continuación son a modo indicativo, el grupo puede alterar el orden en el cual realiza los pasos siempre y cuando no se altere el resultado final esperado.  Considerar que el contenido del archivo de novedades es siempre “liviano” “con pocos registros” | |
| Precondiciones | |
| Determinar antes de empezar a trabajar si el sistema está inicializado, porque si no lo está el comando no puede ejecutar. Sin ambiente no hay proceso. | |
| Para un correcto testeo del script en el directorio de arribos no debe haber ningún archivo  Los archivos se depositan manualmente simulando la llegada por ftp | |
| Contar ciclos | |
| Inicializar el ciclo del proceso en 1. Ir sumando uno cada vez que se repite un ciclo. Grabar en log “voy por el ciclo xx” | |
| Lectura | |
| **Leer el directorio de arribos** y si existen novedades; verificarlas (determinar que el nombre del archivo es correcto, no este vacío, sea legible, no venga duplicado, etc)  Si no hay nada para procesar (no existen novedades) dormir un tiempo x y volver a empezar | |
| Ver si la novedad es procesable | |
| Si hay archivos en el directorio de arribos, verificar que el nombre externo de los archivos sea transactiondate-statecode-merchantcode   |  |  | | --- | --- | | Nombre del Archivo de Novedades (Ejemplo: 0504-B-34567890) | | | Separador de campos: - (guion medio) | | | Campo | Descripción/fuente/valor | | TransactionDate | Formato: mmdd (dos digitos para el mes, dos digitos para el dia) Ejemplo: 0504 | | StateCode | 1 Carácter. Ejemplo: B | | MerchantCode | 8 Caractreres. Ejemplo: 34567889 |   El StateCode debe existir en la tabla de provincias  El Merchantcode debe existir en la tabla de comercios y su estado debe ser “HABILITADO”  La transactiondate debe ser de formato mmdd | |
| Si el nombre del archivo es correcto, Verificar que el archivo no este vacío | |
| Si no está vacío, Verificar que el archivo sea un archivo regular | |
| Si es un archivo regular Verificar que no esté duplicado  Para esto, ver si en el directorio de procesados ya existe un archivo con idéntico nombre. Si ya hay otro archivo con idéntico nombre, asumir que esta duplicado | |
| Aceptar o Rechazar el archivo | |
| Si todas las verificaciones precedentes fueron OK, **el archivo se acepta** y se lo mueve al directorio de aceptados.  Grabar en el log el nombre del archivo aceptado | |
| Si alguna de las verificaciones precedentes falla, el archivo no puede ser aceptado, entonces se lo manda completo al directorio de rechazados.  Grabar en el log el nombre del archivo rechazado y **bien en claro el motivo del rechazo** | |
| Cuando se revisaron todos los nombres de los archivos de arribos se inicia la lectura y procesamiento de cada archivo | |
| Procesar las novedades aceptadas | |
| El archivo de novedades tiene la siguiente estructura   |  |  |  |  | | --- | --- | --- | --- | | Archivo de Novedades ($grupo/$dirnov) | |  |  | | Separador de campos: , (coma) | |  |  | | Campo | Descripción/formato del campo/formato del valor | | | | Campo 1 | idTransaction | "nombre\_del\_campo": valor | Numérico | | Campo 2 | isO03\_cProcessingCode | "nombre\_del\_campo": "valor" | Caracteres | | Campo 3 | isO04\_nTransactionAmount | "nombre\_del\_campo": valor | Numérico | | Campo 4 | isO11\_cSystemTrace | "nombre\_del\_campo": "valor" | Caracteres | | Campo 5 | isO12\_cLocalTransactionTime | "nombre\_del\_campo": "valor" | Caracteres, formato hhmmss | | Campo 6 | isO37\_cRetrievalReferenceNumber | "nombre\_del\_campo": "valor" | Caracteres | | Campo 7 | isO38\_cAuthorizationResponse | "nombre\_del\_campo": "valor" | Caracteres | | Campo 8 | isO39\_cResponseCode | "nombre\_del\_campo": "valor" | Caracteres | | Campo 9 | isO48\_cAdditionalData\_Installments | "nombre\_del\_campo": "valor" | Caracteres | | Campo 10 | isO60\_cReservedPrivate\_HostResponse | "nombre\_del\_campo": "valor" | Caracteres | | Campo 11 | isO62\_cTicketNumber | "nombre\_del\_campo": "valor" | Caracteres | | Campo 12 | isO63\_cReservedPrivate\_BatchNumber | "nombre\_del\_campo": "valor" | Caracteres | | Campo 13 | cGuid | "nombre\_del\_campo": "valor" | Caracteres | | Campo 14 | isO\_MTI\_cMessageType | "nombre\_del\_campo": "valor" | Caracteres | | Campo 15 | isO\_MTI\_cMessageType\_Response | "nombre\_del\_campo": "valor" | Caracteres | | |
| Acerca de los datos | |
| NOTAR que los campos tienen el siguiente formato:  Si es un dato alfabético: “nombre del campo”: “valor”,  Si es un dato numérico: “nombre del campo”: valor | |
| Contar los registros | |
| Contar la cantidad de registros que vienen en el archivo para luego grabar esta cantidad en el log,  Inicializar cada vez que cambia el archivo | |
| Control del registro | |
| Si el registro tiene todos los campos indicados en su estructura, se lo considerará un registro correcto y apto para ser grabado en la salida. | |
| Rechazar un Registro | |
| Si el registro presenta alguna anomalía, no se debe grabar en la salida, en lugar de esto, grabarlo en un archivo general de registros rechazados llamado “RejectedData” que tiene la siguiente estructura   |  |  | | --- | --- | | Archivo de Registros Rechazados: $grupo/$dirnok/rejecteddata | | | Separador de campos: - (guion medio) | | | Campo | Descripción/fuente/valor | | Registro | registro completo de input | | Motivo | motivo del rechazo | | Fuente | nombre del archivo | | |
| Formatear la Salida | |
| Si el registro es apto para grabarlo en la salida se deben crear los siguientes campos:   1. "cOriginalFile": campo nuevo, se obtiene a partir del nombre del archivo de input, ejemplo:” cOriginalFile": "0504-B-34567890" 2. "isO05\_cStateName": campo nuevo, se obtiene a partir del nombre del archivo de input, el campo "StateCode". Se busca ese valor en la tabla de códigos de provincia y se obtiene el StateName. Ejemplo: "isO05\_cStateName": "Buenos Aires" 3. "isO06\_cStateCode": campo nuevo, se obtiene a partir del nombre del archivo de input. Ejemplo: "isO06\_cStateCode": "B" 4. "isO07\_cTransmissionDateTime": campo nuevo, se obtiene concatenando del nombre del archivo de input, el campo "TransactionDate" y el campo "isO12\_cLocalTransactionTime". Ejemplo: "isO07\_cTransmissionDateTime": "0504182346" 5. "isO13\_cLocalTransactionDate": campo nuevo, se obtiene a partir del nombre del archivo de input. Ejemplo: "isO13\_cLocalTransactionDate": "0504" 6. "isO15\_cResponseCodeShortDescription": campo nuevo, se obtiene a partir del campo "isO39\_cResponseCode". Se busca ese valor en la tabla de códigos de respuesta del gateway y se obtiene la descripción corta. Ejemplo: "isO15\_cResponseCodeShortDescription": "AUTORIZADA" 7. "isO42\_cMerchantCode": campo nuevo, se obtiene a partir del nombre del archivo de input. Ejemplo: "isO42\_cMerchantCode": "34567890" 8. "isO49\_cTransactionCurrencyCode": campo nuevo, valor fijo 032 que representa la moneda local argentina. Ejemplo: "isO49\_cTransactionCurrencyCode": "032" 9. "isO04\_cTransactionAmount": reemplaza al campo "isO04\_nTransactionAmount". Se obtiene convirtiendo el valor numérico a caracteres de 12 posiciones completando con ceros a la izquierda. Ejemplo: "isO04\_cTransactionAmount": "000000001400" 10. El resto de los campos que no se mencionan son igual al input | |
| **Registro de Salida** | |
| Armar el registro de salida siguiendo la siguiente estructura: | |
| |  |  |  |  | | --- | --- | --- | --- | | Archivo Financiero ($grupo/$dirsal/<nombre del archivo>) | | | | | Separador de campos: , (coma) | |  |  | | Campo | Descripción/formato del campo/formato del valor | | | | Campo 1 | cOriginalFile | "nombre\_del\_campo": "valor" | Caracteres, Campo Nuevo | | Campo 2 | isO05\_cStateName | "nombre\_del\_campo": "valor" | Caracteres, Campo Nuevo | | Campo 3 | isO06\_cStateCode | "nombre\_del\_campo": "valor" | Caracteres, Campo Nuevo | | Campo 4 | isO07\_cTransmissionDateTime | "nombre\_del\_campo": "valor" | Caracteres, Campo Nuevo, Formato mmddhhmmss | | Campo 5 | isO13\_cLocalTransactionDate | "nombre\_del\_campo": "valor" | Caracteres, Campo Nuevo, Formato mmdd | | Campo 6 | isO15\_cResponseCodeShortDescription | "nombre\_del\_campo": "valor" | Caracteres, Campo Nuevo | | Campo 7 | isO42\_cMerchantCode | "nombre\_del\_campo": "valor" | Caracteres, Campo Nuevo | | Campo 8 | isO49\_cTransactionCurrencyCode | "nombre\_del\_campo": "valor" | Caracteres, Campo Nuevo | | Campo 9 | idTransaction | "nombre\_del\_campo": valor | Numérico | | Campo 10 | isO03\_cProcessingCode | "nombre\_del\_campo": "valor" | Caracteres | | Campo 11 | isO04\_cTransactionAmount | "nombre\_del\_campo": "valor" | 12 caracteres, Campo Nuevo, reemplaza al campo "isO04\_nTransactionAmount". | | Campo 12 | isO11\_cSystemTrace | "nombre\_del\_campo": "valor" | Caracteres | | Campo 13 | isO12\_cLocalTransactionTime | "nombre\_del\_campo": "valor" | Caracteres, Formato hhmmss | | Campo 14 | isO37\_cRetrievalReferenceNumber | "nombre\_del\_campo": "valor" | Caracteres | | Campo 15 | isO38\_cAuthorizationResponse | "nombre\_del\_campo": "valor" | Caracteres | | Campo 16 | isO39\_cResponseCode | "nombre\_del\_campo": "valor" | Caracteres | | Campo 17 | isO48\_cAdditionalData\_Installments | "nombre\_del\_campo": "valor" | Caracteres | | Campo 18 | isO60\_cReservedPrivate\_HostResponse | "nombre\_del\_campo": "valor" | Caracteres | | Campo 19 | isO62\_cTicketNumber | "nombre\_del\_campo": "valor" | Caracteres | | Campo 20 | isO63\_cReservedPrivate\_BatchNumber | "nombre\_del\_campo": "valor" | Caracteres | | Campo 21 | cGuid | "nombre\_del\_campo": "valor" | Caracteres | | Campo 22 | isO\_MTI\_cMessageType | "nombre\_del\_campo": "valor" | Caracteres | | Campo 23 | isO\_MTI\_cMessageType\_Response | "nombre\_del\_campo": "valor" | Caracteres | | |
| Determinar el archivo de salida | |
| Para determinar en qué archivo se debe grabar el registro se debe tener en cuenta el "isO42\_cMerchantCode", la "isO12\_cLocalTransactionTime" y la “hora de cierre”  Si la "isO12\_cLocalTransactionTime" es menor a la “hora de cierre”, el registro se debe grabar en el archivo del mismo mes-dia (ese valor está en el nombre del archivo de input y en el campo "isO13\_cLocalTransactionDate")  Si la "isO12\_cLocalTransactionTime" es mayor o igual a la “hora de cierre”, el registro se debe grabar en el archivo del mes-día siguiente  Tener en cuenta que al sumar un dia puede también cambiar el mes, por ejemplo, si la fecha es 1231 (31 de diciembre) el mes dia siguiente es el 0101. | |
| |  |  | | --- | --- | | Nombre del Archivo Financiero (Ejemplo: 34567890-0531) | | | Separador de campos: - (guion medio) | | | Campo | Descripción/fuente/valor | | MerchantCode | 8 Caractreres. Ejemplo: 34567889 | | TransactionDate | Formato: mmdd (dos digitos para el mes, dos digitos para el dia) Ejemplo: 0531 | | |
| Grabar el registro | |
| Grabar el registro de salida en el archivo correspondiente. Continuar con el siguiente registro hasta el fin de archivo. | |
| Grabar log mientras se ejecuta el script | |
| Todo lo que va sucediendo durante el proceso se debe grabar en el log  El nombre del log debe ser igual al nombre del comando con extensión .log  El directorio donde se graba el log del inicializador es DIRPROC  El log no debe reescribirse, debe conservar la historia  Se debe registrar:  Numero de Ciclo  Nombres de los archivos aceptados  Y por cada archivo que se procesa   * Nombre del archivo que se está procesando * Cantidad de registros de input * Cantidad de registros que se rechazan * Cantidad de registros que se graban en algún archivo de salida | |
| Evitar Reprocesos | |
| Cuando se llega al fin de archivo, este debe ser movido al directorio de procesados para evitar su reproceso | |
| Fin de ciclo | |
| Cuando se termina el ciclo, el proceso duerme un minuto y se reinicia.  Si el ciclo llega a 10000, terminar el programa | |
| 1. ¿el proceso no ejecuta cuando el sistema nunca fue inicializado? |  |
| 1. ¿está documentado dónde y como se realiza este control? |  |
| 1. ¿se previene la ejecución de dos procesos simultáneos? |  |
| 1. ¿está documentado dónde y como se realiza este control? |  |
| 1. ¿cuenta los ciclos? ¿lo escribe en el log? |  |
| 1. ¿detecta cuando arriban las novedades? |  |
| 1. ¿hace bien el control del nombre del archivo de novedades? 2. Valida correctamente la fecha externa (mmdd) 3. Valida correctamente el código de comercio externo? 4. Valida correctamente el código de provincia externo? 5. Si el nombre es incorrecto, ¿lo mueve a rechazados? 6. Si el nombre es incorrecto y el file name tiene espacios intermedios ¿lo mueve a rechazados y desaparece de novedades? |  |
| 1. si el nombre está bien pero el archivo esta vacío, ¿lo mueve a rechazados? |  |
| 1. si el nombre está bien pero el archivo no es un archivo regular, de texto (es una imagen, un pdf, etc.,¿ lo mueve a rechazados? |  |
| 1. si el nombre esta bien pero ya existe en DIRPROC un archivo con idéntico nombre (duplicado) ¿lo mueve a rechazados? |  |
| 1. si el nombre está bien y no viene vacio, es regular y no esta duplicado, ¿ lo mueve a aceptados? 2. ¿Escribe en el log el nombre del archivo aceptado? 3. ¿Escribe en el log el nombre del archivo rechazado y el motivo del rechazo? |  |
| 1. Cuando se terminan las novedades, ¿procesa lo que recibió? 2. Aun cuando no haya recibido novedades, ¿procesa lo que tiene pendiente? |  |
| 1. Si el archivo que esta procesando tiene algun registro dañado, ¿rechaza el registro? 2. Graba correctamente en RejectedData? 3. Cuenta los registros rechazados |  |
| 1. Determina correctamente el destino del registro según su hora de transacción y la fecha de cierre? 2. Si es el ultimo dia del mes, ¿calcula bien el mes siguiente? |  |
| 1. ¿arma el registro de transacciones correctamente? 2. ¿Formatea a caracteres el monto? 3. ¿Crea bien los campos nuevos? |  |
| 1. ¿obtiene la descripción corta del response code? 2. Si el código de respuesta no existe en la tabla, ¿maneja esa situación sin que cancele o se pierda el registro? ¿explica en la hipotesis que se hace? |  |
| 1. ¿graba transacciones? |  |
| 1. ¿mueve de aceptados a procesados? 2. Escribe en el log el resultado? |  |
| 1. ¿duerme y arranca nuevamente otro ciclo? 2. Se detiene cuando supera los 10000 ciclos? |  |
| 1. En el log, ¿hay líneas del progreso de la ejecución? ¿es entendible y útil la información que se graba en el log? 2. Respeta el formato de registro de log? |  |

|  |  |
| --- | --- |
| Arranque y Detención Puede construir una función que haga los dos trabajos, dos funciones separadas, o usar el inicializador en algun modo para que realice algunos de estos trabajos o ambos | |
| STOP  Esta función tiene por objeto detener procesos.  Es complementaria a Start  Evidenciar en un log el resultado de la operación (puede ser compartido con start)  Explicar su uso en el README | |
| START  Esta función tiene por objeto arrancar 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.  Evidenciar en un log el resultado de la operación  Explicar su uso en el README | |
| 1. ¿Hay Indicación correcta en el README de cómo se puede arrancar el proceso? |  |
| 1. ¿Hay Indicación correcta en el README de cómo se puede detener el proceso? |  |

|  |  |
| --- | --- |
| Pruebas | |
| Evidencia de instalación | |
| 1. ¿En el log del instalador, hay evidencia de instalación cancelada, instalación exitosa, reparación, etc? |  |
| Evidencia de inicialización | |
| 1. ¿En el log del inicializador, hay evidencia de ejecución exitosa, ejecución cancelada, etc? |  |
| Evidencia de procesamiento de novedades | |
| 1. ¿En el log del proceso, hay evidencia de aceptación de novedades, rechazo de novedades, grabación de salida, grabación de registros rechazados, etc? |  |
| Evidencia de Arranque – Detención | |
| 1. En el log, hay evidencia que realizaron varias pruebas de arranque - detención  * Arranque con y sin sistema inicializado * Arranque con y sin otro proceso corriendo * Detención |  |

|  |  |
| --- | --- |
| Evidencia de Documentación | |
| 1. Las decisiones tomadas durante el desarrollo, están bien documentadas 2. Es suficiente la explicación del README para que un operador pueda instalar el sistema 3. Es suficiente la explicación del README para que un usuario pueda ejecutar el sistema? |  |

1. cree un subdirectorio dentro de $grupo para salvaguardar el paquete original (completo) y reparar el sistema. Indique en la documentación cual es el nombre de ese directorio [↑](#footnote-ref-2)
2. ¿Qué es un log?

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

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

   A estos 5 valores se los llama estándar W5, por su origen en inglés: when, who, where, what and why. [↑](#footnote-ref-3)