**Proceso Diario y de Distribucion**

**Archivos:**

Message.txt

**Logs:**

Proceso Diario Mes.txt

**Scripts: (muchos de los scripts fueron transformados en funciones)**

procesoSuper.sh: Es básicamente el main, que va llamando a los distintos scripts descriptos a continuación, lo único a destacar es que al final manda un mail con el message.txt generado por los scripts mailingProcesoDiario y mailingProcesoDistribución y lo manda a varios responsables.

ProcesoDiario.sh: Realiza la ejecución de los procesos php propios de un día de trabajo.

ProcesoDistribucion.sh: Realiza la ejecución del proceso de distribución, el cual se ejecuta una vez al mes.

comprobadorDeErrores.sh: Recibe 3 Parametros, el $? Que indica si el ultimo comando ejecutado fue ejecutado correctamete o si se produjeron errores, el log donde se registrará la información y el nombre que quiere utilizarse en el log. Ejemplo si quiero decir que se ejecuto distribución.php se lo mando como parámetro ( como un string) y en el log aparecerea “se ejecuto distribución.php”

fechasCierre.sh: Chequea con todas las fechas de cierre de mes, para ver si el dia de hoy es un dia de fecha de cierre, si lo es ejecuta procesoDistribucion y lo registra y se agrega al mail que se enviara a futuro que se realizó el proceso de distribución. (las fechas tienen un + 1 en el dia porque la distribución se hace el dia siguiente al cierre).

mailingProcesoDiario:

Se ejecuta cada vez que se ejecuta proceso super y lo que hace es borarr el message .txt anterior y deja registrado si se realizó o no bien el proceso diario.

mailingProcesoDistribucion: Se fija si se ejecuto correctamente el proceso de distribución y si así fue, lo registra en message.txt el cual luego será enviado por mail, si hay un error también lo deja registrado y lo mando por mail.

**Backup**

**Archivos**

Backup\_contador.txt: Es el contador que se va guardando dia a dia para ir guardando todoslos días en una carpeta consecutiva.

**Log**

Backup\_log.xt

**Scripts: (muchos de los scripts fueron transformados en funciones)**

comprobadorDeErrores.sh: Recibe 3 Parametros, el $? Que indica si el ultimo comando ejecutado fue ejecutado correctamete o si se produjeron errores, el log donde se registrará la información y el nombre que quiere utilizarse en el log. Ejemplo si quiero decir que se ejecuto distribución.php se lo mando como parámetro ( como un string) y en el log aparecerea “se ejecuto distribución.php”

backup\_script.sh: Comprime unas carpetas especificas, se checkea que se compriman bien, y luego se mueven a la carpeta que le corresponde dependiendo de que dia se trate.

**Reportes**

**Scripts: (muchos de los scripts fueron transformados en funciones)**

mailingReportes.sh: Este script recibe como parámetro $?, en base a eso decide si hubo error o no, sea cual sea el caso, envía por mail notificando si se ejecutó bien la generación del reporte.

generarReporte.sh: Genera una consulta SQL, las consultas están guardadas en la carpeta querys, uno solo tiene que ingresar el nombre de la query a la que se querrá acceder de esa carpeta, y la consulta se genera, se chequea si hubo error al generar la consulta, si hubo error se asienta en el log y se envía por mail y si funciona bien también se asienta en el log y se manda por mail. Luego se envía por mail el resultado de la consulta SQL en un archivo adjunto.

**Logs:** reportes\_log

**Carpeta Reportes:** Acá se copia una copia del reporte generado con su fecha y hora de generación por titulo. Aparte de indicar el tipo de reporte de el cual se trata.

**Archivos Necesarios:**

Message.txt

Todas las querys de la carpeta querys.

**CRONTAB**

Como es un servidor Linux no poseemos permisos de super usuario por lo que hay que setear el crontab de la siguiente forma:

Crontab –e (esto es el crontab del usuario, solo puede ejecutar cosas propias del usuario y solo puede acceder a archivos del usuario)

Al ejecutar ese comando en la consola se abrirá con VI (un editor de texto plano ) el crontab del usuario, para poder escribir hay que apretar la tecla “insert”, luego se escribe lo que se quiere cronear ( siguiendo las reglas de crone donde el primer luego es para minutos, el segundo para horas en especifico, etc etc ). Si se deja por defecto con todo en asteriscos, cada un minuto automáticamente se va a ejecutar el script que le configuremos.

Luego para salir de VI hay que apretar la tecla Esc, y escribir sin comillas “:wq” que significa write and quite, lo que quiere decir que van a guardar los cambios, si uno quisiera no guardar los cambios y salir, el proceso es similar, tecla Esc seguido de escribir “:q”.