SEGUNDO INFORME

PRACTICA PROFESIONAL

CASA EDITORIAL EL TIEMPO

PRESENTADO POR:

LUIS ALEXANDER GÜIZA MALAGÓN

PROFESOR:

YOAN PINZÓN

UNIVERSIDAD NACIONAL DE COLOMBIA

2014

En este informe se trata de mencionar las diferentes actividades realizadas durante el mes de mayo en el banco Falabella, también los avances realizados sobre la aplicación que automatiza un proceso manual dentro del área de seguridad de la información, este proceso agiliza la depuración de empleados dentro de la compañía.

Otro aspecto importante son es el levantamiento de requerimientos del proyecto que se va a automatizar y los avances en un nuevo proyecto referente a un Etichal Hacking que se realiza con la supervisión de las directivas de la compañía éste proyecto es un trabajo en conjunto con el subgerente de seguridad tecnológica.

### MARCO TEÓRICO

En el desarrollo de las aplicaciones PROYECTOS DAME y LIMPIA LOG fue necesario investigar y aprender sobre los siguientes temas: Manejo de DTS, Job, BAPIs y SAP. Puesto que dentro de la universidad estos temas no son tratado a tal profundidad como se manejan en las empresas de desarrollo tales como EL TIEMPO.

Data Transform Sistem (DTS)

Son un conjunto de objetos y utilidades que automatizan la extracción, transformación y carga de operaciones desde una base de datos, los objetos son llamados DTS packages o paquetes DTS.

Los paquetes DTS son conjuntos organizados de conexiones, tareas DTS, transformaciones DTS y restricciones de flujo de trabajo reunidas mediante programación o una herramienta de DTS y guardadas en Microsoft® SQL Server™, Meta Data Services de Microsoft SQL Server 2000, un archivo de almacenamiento estructurado o un archivo de Microsoft Visual Basic®.

Cada paquete contiene uno o más pasos que se ejecutan de forma secuencial o en paralelo cuando se ejecuta el paquete. Cuando se ejecuta el paquete, se conecta con los orígenes de datos apropiados, copia los datos y los objetos de base de datos, transforma los datos y notifica los sucesos a otros usuarios o procesos. Puede modificar los paquetes, protegerlos mediante contraseña, programar su ejecución y obtener su versión.

Una tarea DTS es un conjunto discreto de funcionalidad, ejecutada como un único paso de un paquete. Cada tarea define un elemento de trabajo que debe realizarse como parte de la transferencia de datos y el proceso de transformación de datos, o como un trabajo que debe ejecutarse.

DTS proporciona varias tareas que forman parte del modelo de objetos de DTS y a las que se puede tener acceso gráficamente (a través del Diseñador DTS) o mediante programación. Estas tareas, que puede configurar de forma individual, cubren una amplia gama de situaciones de copia de datos, transformación de datos y notificación[[1]](#footnote-1).

Descripción de Proyectos

Proyecto Depuraciones

Este proyecto tiene tres pasos fundamentales los cuales son planeación, desarrollo y documentación para futuros soportes.

Consta de mejorar el tiempo en la depuración de usuarios, es decir: mejorar el tiempo requerido para borrar un usuario del sistema ya que cada empleado que sale de la compañía tiene que borrarse de todos los grupos, servidores y demás accesos dentro del banco, esta operación evita que los ex empleados tengan la posibilidad de gestionar algún tipo de fraude dentro de la compañía.

La depuración de usuarios se estaba realizando por medio de cruces entre tablas de Excel, ésta operación era realizada por un operario el cual demoraba un promedio de dos horas para desarrollar dicha actividad, teniendo en cuenta que el tiempo se puede reducir con una software, se hace la solicitud para desarrollar dicha aplicación.

Como primer paso de este proyecto fue necesario pedir un espacio dentro de uno de los servidoresSQL del banco para poder implementar la aplicación, junto a ello se solicitó espacio dentro de mismo servidor, esto con el fin de colocar los archivos necesarios del proyecto, en este caso DTSx, que sirven para crear una comunicación entre los datos de Excel que son los que se manejan para realizar la depuración y los datos de lo base de datos.

Teniendo todas las herramientas necesarias, el paso a seguir fue diseñar un diagrama para aclarar el diseño de la aplicación y ciertos requerimientos para presentar en el primer avance del proyecto.

Requerimientos funcionales del primer avance.

* RF\_1: Debe existir un botón para agregar los archivos Excel y poder ingresar los datos de manera automática, los archivos que serán agregados son: cartera.xlsx, PlantaActiva.xls y egresados.xls.
* RF\_2: El proyecto debe entregar el primer informe con un filtro donde el campo ACCOUNT \_STATUS sea igual a OPEN y el campo DEFOULT\_TABLESPACE sea igual a USER.
* RF\_3: La depuración anterior debe ser guardada para constatarla con la siguiente depuración.
* RF\_4: El informe entregado debe ser en un archivo Excel, es decir un .xls

Diagrama de la primer entrega

Para explicar los primeros avances del proyecto fue necesario crear un diagrama parcial el cual expongo a continuación:



Requerimientos funcionales para la entrega final

* RF\_5: La tabla cartera debe cruzar con la tabla Activos de acuerdo al nombre de usuario.
* RF\_6: Del cruce en el requerimiento 5 es necesario que la aplicación entregue dos informes:
  + Los usuarios que están en la tabla Cartera pero no están en Activos.
  + Los usuarios que están en la tabla Activos pero no están en Cartera.
* RF\_7: El cruce de la referencia 5 debe quedar guardado en una tabla para comparar con la siguiente depuración, esta tabla por lo general es llamada ActivosAnteriores.
* RF\_8: La tabla planta activa debe cruzar con la tabla antes mencionada ActivoAnteriores por el documento del usuarios.
* RF\_9: Al igual que en el requerimiento 6 El cruce referenciado en el requerimiento 8 debe entregar dos informes en un documento Excel , dichos informes deben tener los siguientes parámetros:
  + Los usuarios que están en la tabla ActivoAnteriores pero no están en PlantaActiva.
  + Los usuarios que están en la tabla PlantaActiva pero no están en ActivoAnteriores.

Diagrama de la entrega final



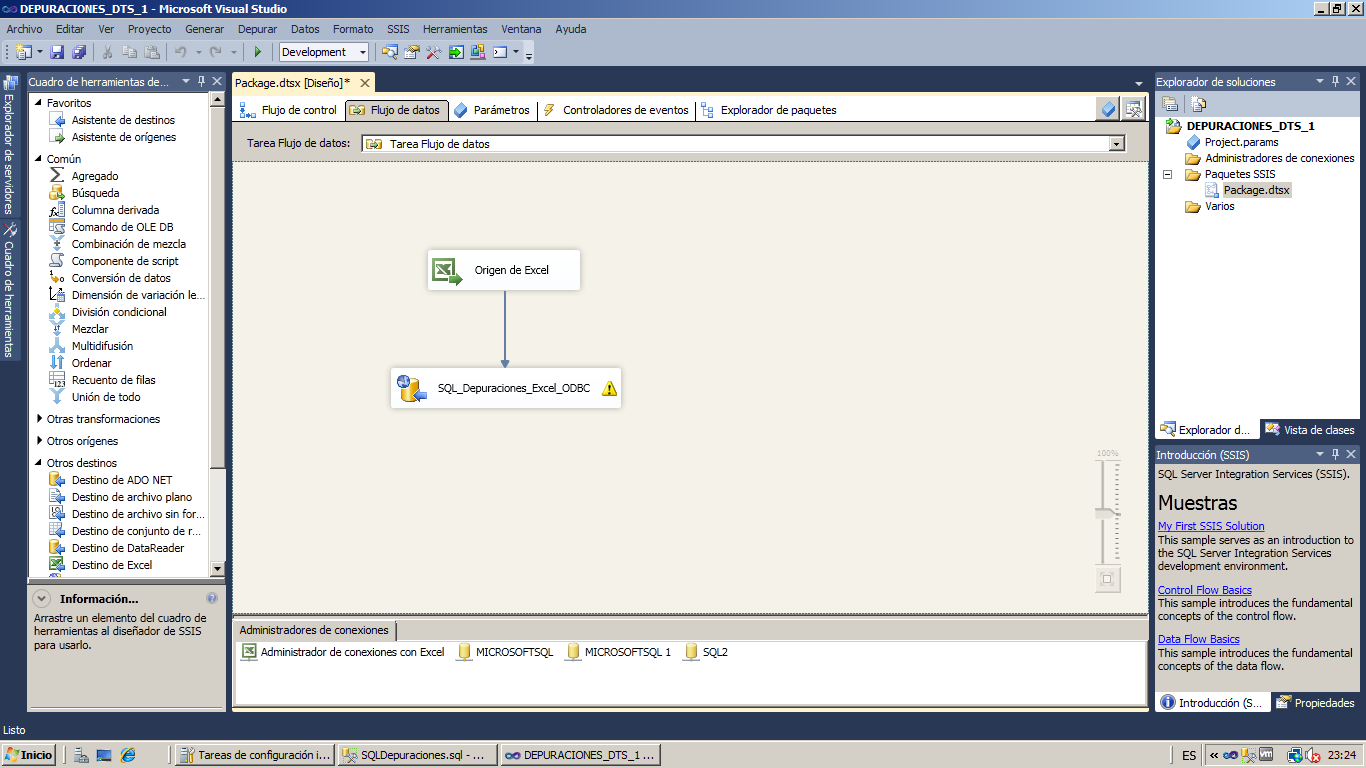
Avances del proyecto

Dado que la primera entrega fue el 5 de mayo, el avance consta de la primer avance, entre los cuales se realizaron los primeros 4 requerimientos expuestos anteriormente.

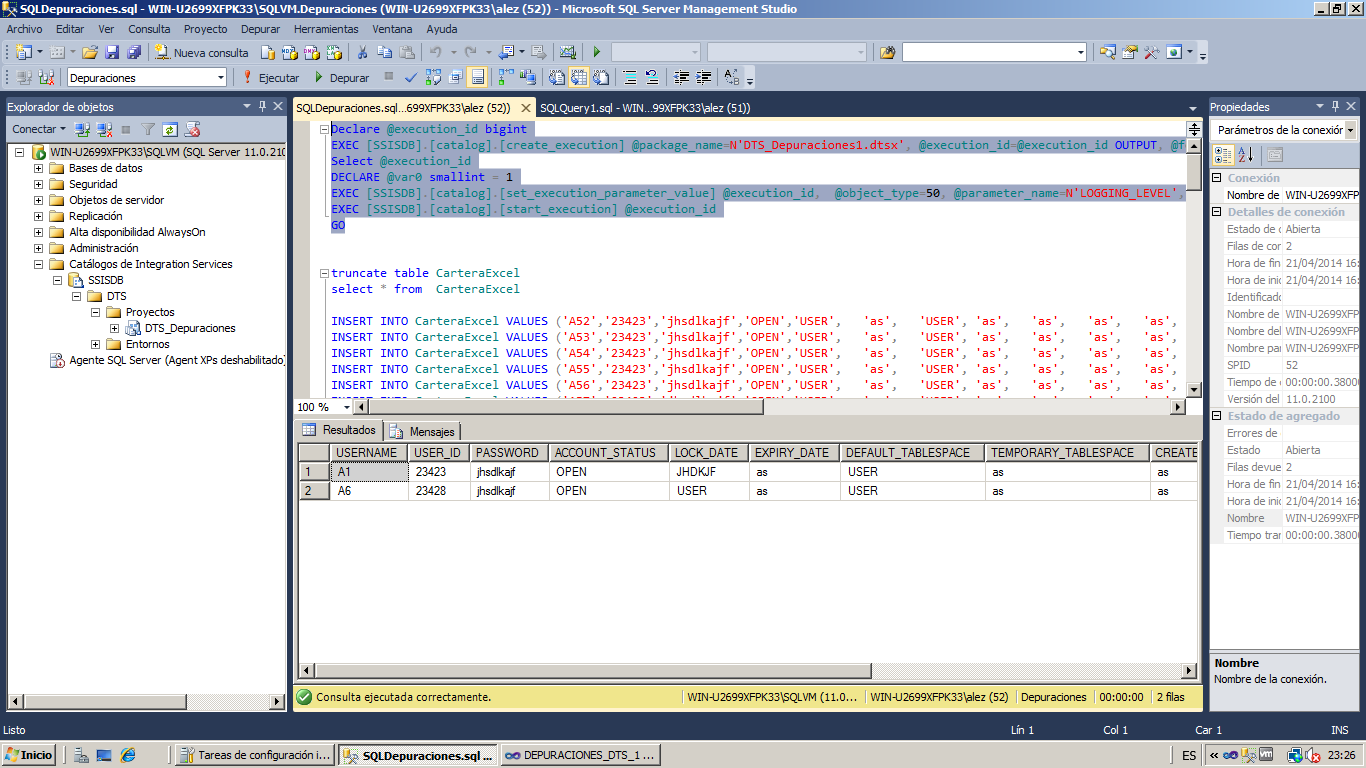
Para desarrollar el primer requerimiento fue necesario hacer uso de la herramienta Integration Services que ofrece Visual Studio, que le permite ala programador integrar servicios de Excel y SQLServicer por medio de DTSx.

A continuación los avances del primer requerimiento:

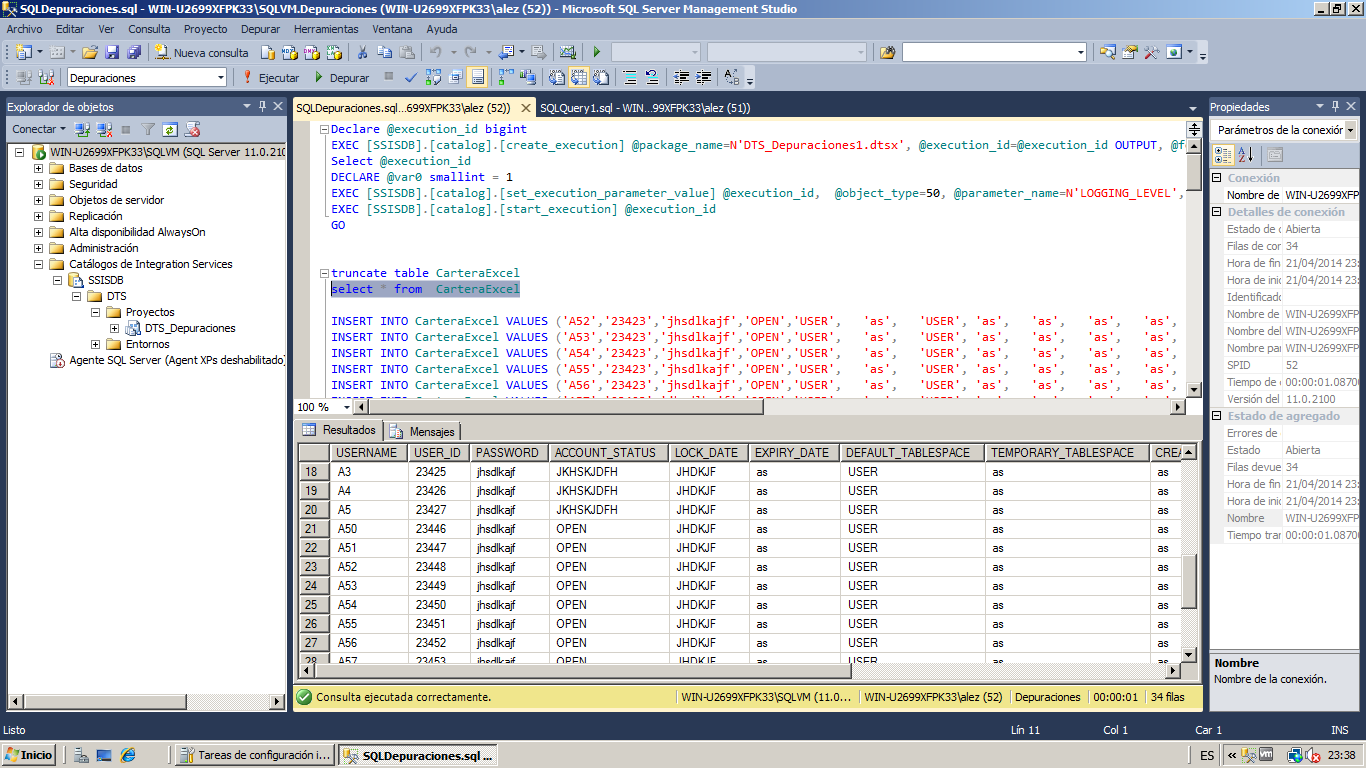
La siguiente imagen muestra el diagrama del DTSx utilizado para el primer requerimiento.



En la siguiente imagen muestra la integración con el motor de base de datos SQLSserver.



Con el código en el cuadro rojo se ejecuta la DTSx.



Con el siguiente código, desarrollo los requerimiento 2,3 y 4.

USE [Depuraciones]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[sp\_Depu\_Cartera] Script Date: 18/05/2014 19:01:43 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER procedure [dbo].[sp\_Depu\_Cartera]

AS

TRUNCATE TABLE CARTERA

INSERT Cartera

SELECT \* FROM CarteraExcel

WHERE ACCOUNT\_STATUS = 'OPEN'AND DEFAULT\_TABLESPACE = 'USER'

-----------------------------------------------------------------------------------------

DECLARE @EXISTE NVARCHAR(50)

WHILE EXISTS (SELECT \* FROM CartERAeXCEL WHERE ACCOUNT\_STATUS = 'OPEN'AND DEFAULT\_TABLESPACE = 'USER')

--DECLARE @EXISTE NVARCHAR(50)

BEGIN

SET @EXISTE = (SELECT TOP 1 USERNAME FROM CarteraExcel WHERE ACCOUNT\_STATUS = 'OPEN'AND DEFAULT\_TABLESPACE = 'USER' ORDER BY USERNAME DESC)

if exists (SELECT \* FROM Cartera inner JOIN CarteraAnt ON CarteraAnt.USERNAME = Cartera.USERNAME WHERE CarteraAnt.USERNAME =@EXISTE)

update Cartera set Costumer = 'eXISTE' WHERE USERNAME = @EXISTE

ELSE

update Cartera set Costumer = 'CREADO' WHERE USERNAME = @EXISTE

DELETE FROM CartERAeXCEL

WHERE USERNAME =(SELECT TOP 1 USERNAME FROM CartERAeXCEL WHERE ACCOUNT\_STATUS = 'OPEN'AND DEFAULT\_TABLESPACE = 'USER' ORDER BY USERNAME DESC )

END

------------------------------------------------------------------------------------------------------------

--DECLARE @EXISTE NVARCHAR(50)

WHILE EXISTS (SELECT \* FROM CarteraAnt WHERE ACCOUNT\_STATUS = 'OPEN'AND DEFAULT\_TABLESPACE = 'USER')

BEGIN

SET @EXISTE = (SELECT TOP 1 USERNAME FROM CarteraAnt WHERE ACCOUNT\_STATUS = 'OPEN'AND DEFAULT\_TABLESPACE = 'USER' ORDER BY USERNAME DESC)

if exists (SELECT \* FROM Cartera inner JOIN CarteraAnt ON CarteraAnt.USERNAME = Cartera.USERNAME WHERE CarteraAnt.USERNAME =@EXISTE)

update Cartera set Costumer = 'EXISTE' WHERE USERNAME = @EXISTE

ELSE

update CarteraAnt SET Costumer = 'ELIMINADO' WHERE USERNAME = @EXISTE

IF NOT EXISTS (SELECT TOP 1 \* FROM Cartera WHERE USERNAME =@EXISTE)

INSERT Cartera SELECT TOP 1 \* FROM CarteraAnt WHERE USERNAME =@EXISTE

DELETE FROM CarteraAnt

WHERE USERNAME =(SELECT TOP 1 USERNAME FROM CarteraAnt WHERE ACCOUNT\_STATUS = 'OPEN'AND DEFAULT\_TABLESPACE = 'USER' ORDER BY USERNAME DESC )

END

---------------------ejecuta dts que va al excel--------------------

Declare @execution\_id bigint

EXEC [SSISDB].[catalog].[create\_execution] @package\_name=N'DTS\_Depuraciones2.dtsx', @execution\_id=@execution\_id OUTPUT, @folder\_name=N'DTS', @project\_name=N'Exportar Datos', @use32bitruntime=False, @reference\_id=Null

Select @execution\_id

DECLARE @var0 smallint = 1

EXEC [SSISDB].[catalog].[set\_execution\_parameter\_value] @execution\_id, @object\_type=50, @parameter\_name=N'LOGGING\_LEVEL', @parameter\_value=@var0

EXEC [SSISDB].[catalog].[start\_execution] @execution\_id

TRUNCATE TABLE CarteraAnt

insert CarteraAnt select \* from Cartera

Actividades Propias del Cargo

Las actividades del cargo, basicamente son: el monitoreo de los tramas con infección (Malware, Adware, Spyware etc) a través de la herramienta Alient Vaunt , la cual tiene un registro de tramas maliciosas y las reporta apenas las detecta. La función específica es mirara esas tramas que la herramienta declara como maliciosas y hacer un análisis sobre el porqué la declaro como una alarma y determinar si es un falso positivo o un incidente.

Para determinar si dicha alarma es un incidente o un falso positivo es necesario consultar el origen, el destino en ciertas páginas de scaneo de url y archivos maliciosos entre los cuales están:

Virus total, urlquery entre otros.

También se analiza el tráfico interno, ya que es posible que la mayor amenaza sean los mismos empleados debido a que conocen el ambiente y la infraestructura de seguridad por ende es tan importante ver lo que sale y entra de la red.

Por otra parte el detector de intrusos Alient Vaunt revisa las vulnerabilidades de la red, un caso puntual es el paso de contraseñas en texto claro, es decir: que en la trama circule la contraseña de algunos usuarios sin encriptación.

A continuación un ejemplo de una página con código malicioso en el transcurso de la semana:

1. <http://www.eumed.net/libros-gratis/2009c/574/CONCEPTOS%20BASICOS%20DE%20DTS.htm> [↑](#footnote-ref-1)