

Repositorio de Scripts (Utilitarios)

Es un repositorio que contiene scripts que han sido creados con el fin de brindar soluciones a procesos repetitivos o que brindan soluciones a situaciones de actualización de fallas en los procesos.
¹⁾

Estructura del Repositorio

El repositorio ha sido creado en la siguiente url:

<http://adacsc.co:1443/svn/repository/ADA/SICOF/Documentos/Desarrollos/DB> y contiene la siguiente estructura de directorios: ²⁾

- Oracle
- SQL SERVER
- My SQL

Oracle

Esta es la tecnología en la que se soportan la mayoría de la aplicaciones que componen el ERP. Por lo tanto es el repositorio que contiene la mayor cantidad de scripts, está dividido en los siguientes grupos:

- **01 Utilerías:** Este directorio contiene aquellos scripts que realizan procesos generales en el ERP. Por lo general no están relacionados directamente con aplicaciones ó módulos.
- **02 Módulos:** En este directorio se agrupan los scripts de acuerdo a los módulos del ERP.

SQL SERVER

³⁾

My SQL

⁴⁾

Proceso de Gestión de Scripts

En esta sección se describe el proceso de registro y actualización de scripts en el repositorio.

- **Adición de Scripts:** Para adicionar un script copie el archivo que contiene la(s) sentencia(s) previamente validadas al directorio correspondiente, confirme los cambios y diligencie el [Archivo Leeme.txt](#).
- **Edición de Scripts:** Para editar un script abra el archivo en un editor de texto de su preferencia, modifíquelo, válidelo, guardelo, confirme los cambios y diligencie el [Archivo](#)

[Leeme.txt](#).

- **Eliminación de Scripts:** Solo se permite la eliminación para los dueños de sus propios scripts, para realizar el proceso, actualice el repositorio, elimine el archivo del script, confirme los cambios y diligencie el [Archivo Leeme.txt](#).

Archivo Leeme.txt

Es un archivo muy importante para la adecuada gestión de los repositorios, esta contenido en todos los directorios y su objetivo es llevar el registro e historial de actualizaciones de cada script.

Plantilla de Ejemplo

A continuación se describe el contenido de la información que debe ser diligenciada al momento de agregar o actualizar un script teniendo presente la siguiente imagen:

```
#####  
# ARCHIVO DE IMPLEMENTACION  
#####  
# Resumen:  
# Implementación de Scripts  
#####  
# LISTA DE SCRIPTS DB  
#####  
# Plantilla de Ejemplo (Copiar para adicionar scripts teniendo presente los separadores #)  
# Desarrollador: [Correo ADA del usuario que sube el script. Ejemplo:carlos.torres@ada.co]  
# Script N°: [Identificador del Script. Ejemplo: 001]  
# Script Name: [Nombre del Script. Ejemplo Actualización de Saldo Contables]  
# Script File: [Nombre del Archivo. Ejemplo 001 Actualización de Saldo Contables.sql]  
# Objetivo/Funcionalidad: [Describir lo que hace el script. Ejemplo: Actualiza los saldos contable de una vigencia.]  
# Observaciones:  
# [Consideraciones que se deben tener presente al momento de ejecutar el script, se debe registrar en forma de lista]  
# Ejemplo:  
# [*La vigencia debe estar abierta.]  
#####
```

Descripción de los campos

- **Desarrollador:** Correo del usuario ADA
- **Script N°:** Identificador del Script
- **Script Name:** Nombre del Script
- **Script File:** Nombre del Archivo
- **Objetivo/Funcionalidad:** Describir lo que hace el script
- **Observaciones:** Consideraciones que se deben tener presente al momento de ejecutar el script, se debe registrar en forma de lista.

Diligenciamiento de un registro de script

El siguiente ejemplo muestra como se debe registrar la información en el archivo.

```
#####
# ARCHIVO DE IMPLEMENTACION
#####
# Resumen:
# Implementacion de Scripts
#####
# LISTA DE SCRIPTS DB
#####
# Plantilla de Ejemplo (Copiar para adicionar scripts teniendo presente los separadores #)
# Desarrollador: [Correo ADA del usuario que sube el script. Ejemplo:carlos.torres@ada.co]
# Script N°: [Identificador del Script. Ejemplo: 001]
# Script Name: [Nombre del Script. Ejemplo Actualización de Saldos Contables]
# Script File: [Nombre del Archivo. Ejemplo 001 Actualización de Saldos Contables.sql]
# Objetivo/Funcionalidad: [Describir lo que hace el script. Ejemplo: Actualiza los saldos contable de una vigencia.]
# Observaciones:
# [Consideraciones que se deben tener presente al momento de ejecutar el script, se debe registrar en forma de lista]
# Ejemplo:
# [*La vigencia debe estar abierta.]
#####
# Historial de Actualizaciones:
# Revision History: [Número de la revision. Ejemplo 1.0] - [Usuario ADA. Ejemplo: carlos.torres@ada.co - [Fecha. Ejemplo:
#####
# Desarrollador: carlos.torres@ada.co
# Script N°: 01
# Script Name: Convertir DB a un ambiente de Pruebas.sql
# Script File: 01 Convertir DB a un ambiente de Pruebas.sql
# Objetivo/Funcionalidad: Actualiza la información de una base de datos convirtiendola en
# : en un ambiente de pruebas realizando los sdiguientes procesos:
# : *Cambiar el nombre de la empresa, nit y dependencia
# : *Cambiar nombres de Terceros
# : *Cambiar nombre de empresa genérica.
# : *Ocultar información sencible de los tercero, nit, direccion, telefono, email
# Observaciones:
# *Este proceso es irreversible por lo que se debe estar seguro del ambiente que se desea actualizar
# *El bloque anónimo se debe ejecutar con el usuario PRESUP01
# *El alcance del scripts solo impacta las tablas del Financiero.
#####
# Historial de Actualizaciones:
# Revision History: 1.0 - carlos.torres@ada.co - 21/08/2018 10:57:25 : Liberación Inicial
#####
```

Observaciones

Tener presente las siguientes observaciones para la correcta gestión de scripts.

- Todo script debe tener un registro en el archivo **Leeme.txt** debidamente diligenciado.
- Todo script debe ser validado antes de subirse al repositorio.
- Todos los archivos de scripts deben subirse con la extensión **.sql**
- Todos los usuarios que intervienen en la gestión de los scripts son responsables de su impacto por lo tanto la documentación debe ser clara y precisa.
- Los archivos de scripts deben crearse para un fin específico no debe agrupar procesos en un mismo archivo si impactan procesos diferentes.⁵⁾
- Todos los scripts que contengan procesos secuenciales ó de varias sentencias deben incluirse en un bloque anónimo.
- Todos los scripts que incluyen procesos secuenciales ó de varias sentencias deben incluir un capturador de errores.
- No se pueden incluir **commit** parciales en los scripts a menos que esa condición sea controlada en caso que el script deba volverse a ejecutar en un mismo ambiente. Es responsabilidad del desarrollador evaluar ese impacto.
- *(Opcional)* No debe mezclar instrucciones **DDL** y **DML** en un mismo proceso. Es responsabilidad del desarrollador evaluar ese impacto.

1)

La creación y uso de estos scripts no debe interpretarse como solución definitiva a problemas de la aplicación. Si una situación es recurrente y corresponde a errores en las aplicaciones estos scripts deberán considerarse como soluciones temporales y deben reportarse a fabrica para la corrección del problema en el código fuente.

2)

De acuerdo a la tecnología de base de datos utilizada

Last update: 2018/08/22 14:58 ada:howto:sicoferp:database:repositorioscripts <http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:database:repositorioscripts>

3) 4)

Pendiente por construcción

5)

En estas situaciones por favor cree varios archivos

From:

<http://wiki.adacsc.co/> - **Wiki**

Permanent link:

<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:database:repositorioscripts>



Last update: **2018/08/22 14:58**