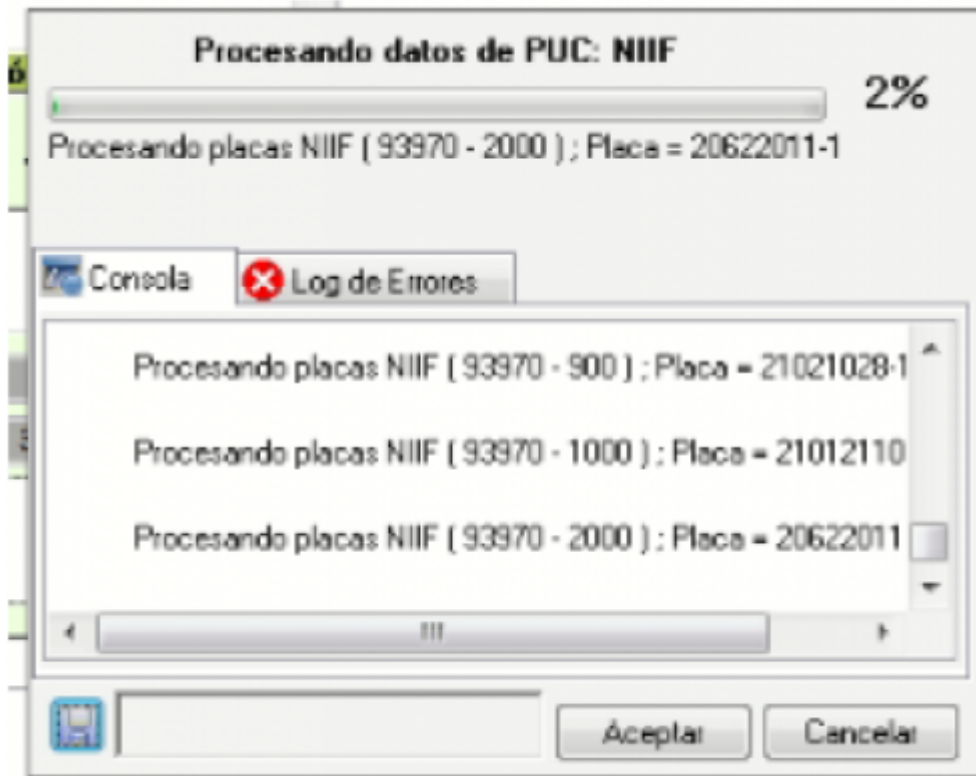


DEPRECIACIÓN ACTIVOS FIJOS

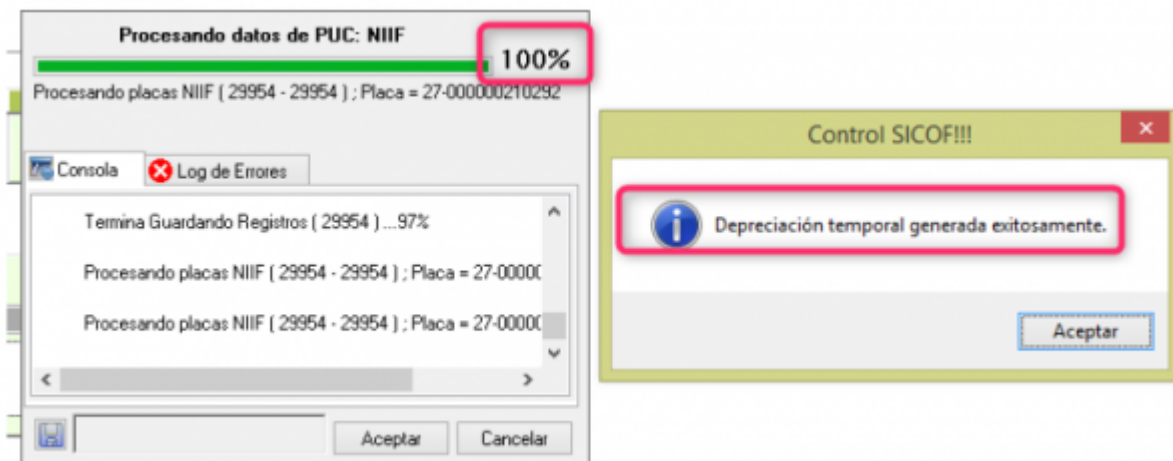
Se ingresa a la opción y se ejecuta primero la depreciación temporal del mes. Al momento de iniciar el proceso de la Depreciación Temporal, se agregó una ventana de validación al proceso, representado en porcentaje para que el usuario visualice que efectivamente el proceso se esté generando.



Al momento de llegar al 100% el proceso de Depreciación Temporal, se pueden presentar 2 escenarios: 1. Depreciación Temporal exitosamente. 2. Depreciación con Errores y Advertencias.

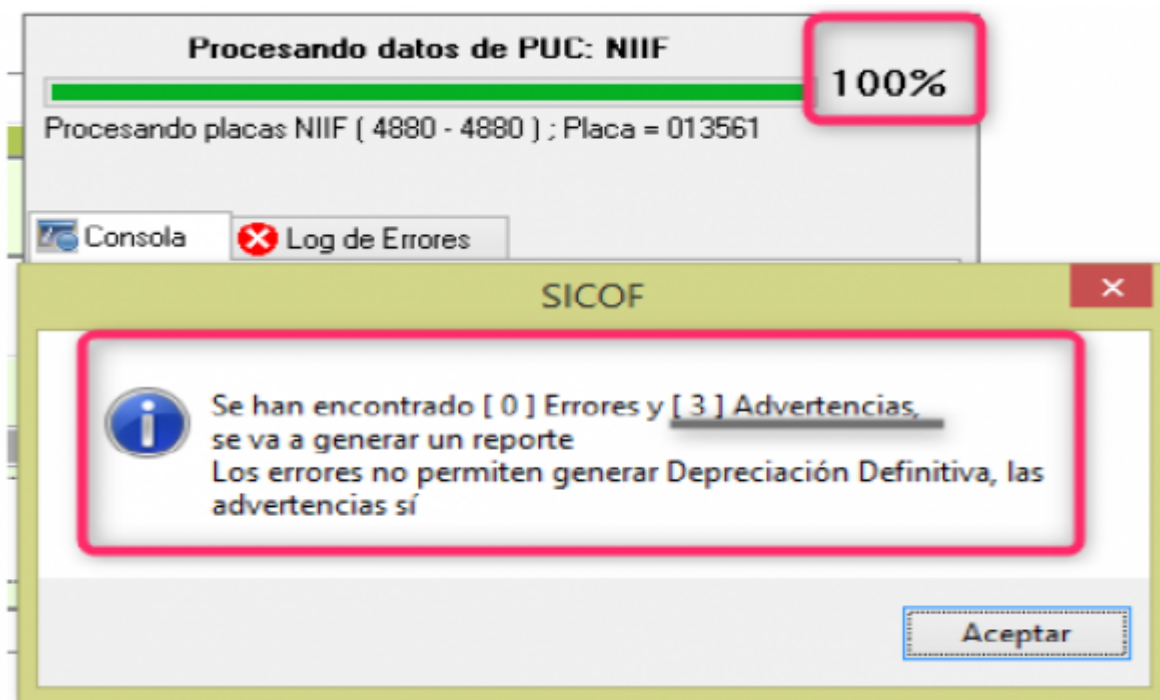
Escenario 1:

Cuando genera de manera Exitosa se da clic en aceptar y paso siguiente es guardar el archivo y realizar las validaciones correspondientes.

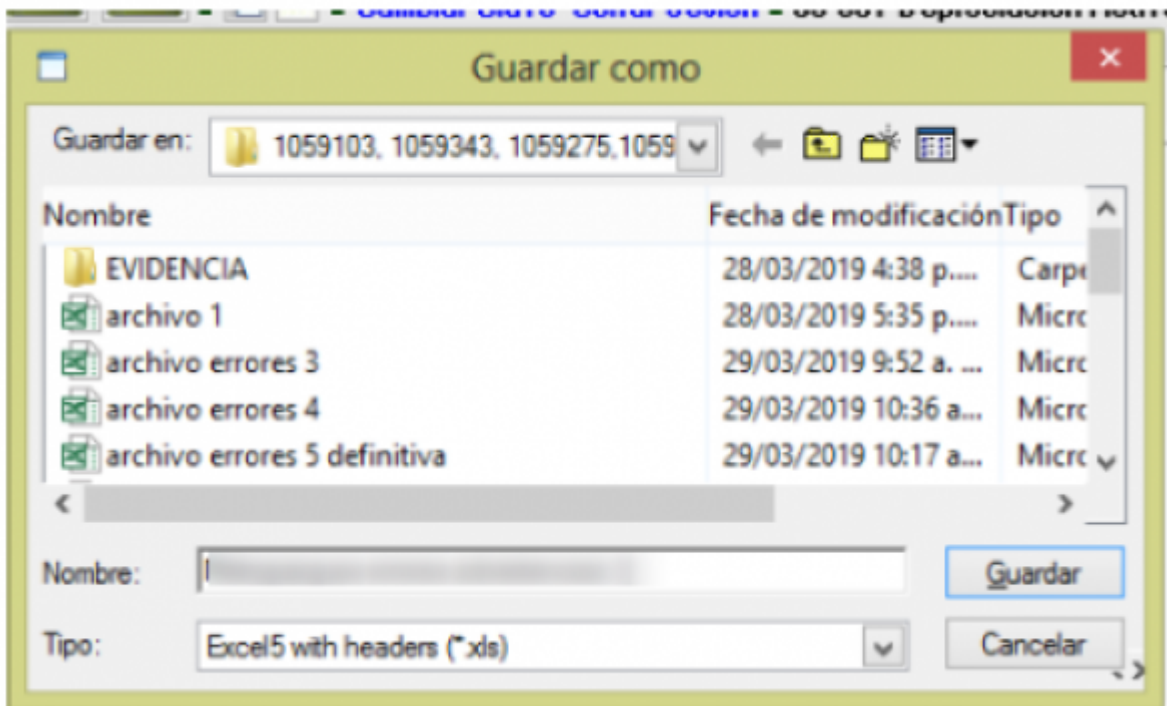


Escenario 2:

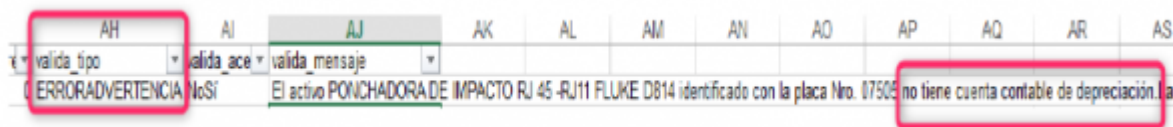
Cuando genera Errores y Advertencias se le debe dar clic en aceptar:



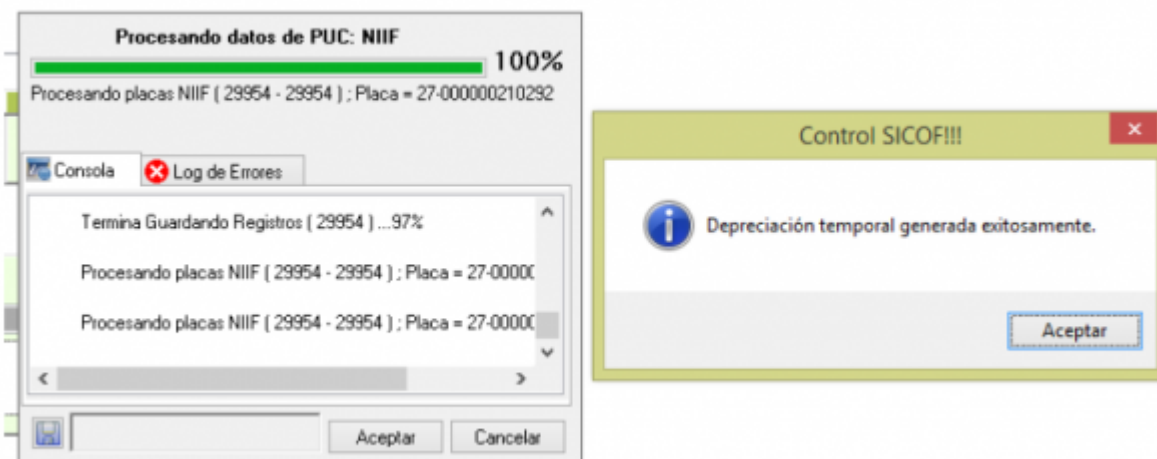
Y este a su vez lo enlaza con su equipo para que guarde el archivo de errores y advertencias:



Al generar el reporte el usuario debe ingresar y corregir los mensajes de validación. Uno de los posibles mensajes es:



Cuando todos los errores se encuentren subsanados por el usuario, finalmente el sistema le permitirá generar la Depreciación Temporal exitosamente.



Notas del proceso

- Las advertencias permiten generar la Depreciación Temporal sin restricción alguna, ya que estas advertencias no generan ningún tipo de asiento; caso contrario a los errores.

- Para informar las placas que tienen problemas de cuenta contable, el formato es el siguiente:

[Campo en Placa(Campo en Depreciación)]:

donde “Campo en Placa” es el título de la columna en la definición de las cuentas contables de las placas, y, “Campo en Depreciación” es el nombre interno del campo de la depreciación que se está afectando para poder ubicarlo en el código.

Corrección de Duplicidad en Novedades de Depreciación y Marcación de Ajustes

1. Contexto del Proceso

El sistema realiza la depreciación mensual de activos fijos utilizando el objeto de usuario `n_cst_depreciacion_activos_new`. Este proceso calcula la cuota mensual, actualiza saldos acumulados y genera registros en tablas temporales (`mov_depreciacion_temp`) antes de la causación definitiva.

2. Descripción del Problema

Se identificaron dos comportamientos incorrectos en el proceso de depreciación masiva:

Duplicidad de Valores en Novedades (Double-Dipping): Cuando un activo (“placa”) tiene una novedad de tipo Adición (ADIC) o Disminución (DISM) aprobada que lo deja totalmente depreciado (saldo cero), el motor de cálculo sumaba nuevamente el valor de dicha novedad al acumulado del mes. Dado que el valor de la novedad ya se había aplicado contablemente al aprobarla, el reporte temporal mostraba una depreciación acumulada inflada o incorrecta. **Omisión de Marcación de Ajustes:** El campo `es_ajuste` en la tabla `mov_depreciacion_temp` no se estaba actualizando. Esto impedía identificar qué registros provenían de movimientos operativos normales y cuáles eran ajustes por novedades de períodos anteriores.

3. Causa Raíz

Lógica de Cálculo: En la función de depreciación masiva, el cálculo de `depreciacion_acumulada` no discriminaba si el valor de la adición ya formaba parte del saldo base en el momento de alcanzar el límite de vida útil. **Falta de Asignación en DataWindow:** El DataWindow de actualización no incluía la columna `es_ajuste` en su sintaxis de actualización, y el script del objeto de negocio no realizaba el SetItem correspondiente.

4. Solución Implementada

A. Control de Acumulados (Lógica de Negocio) En el objeto `n_cst_depreciacion_activos_new.sru`, específicamente en la función `of_depreciacion_masiva_placa_si_deprecia`, se introdujo la variable `ll_val_mes_sin_novedad`.

Lógica: Se separa el cálculo de la cuota base del mes de los valores de ajuste. Corrección: Al calcular la depreciación acumulada para la tabla temporal, ahora se utiliza: Round(luo_data.valor_depreciado + ll_val_mes_sin_novedad, 0) Esto asegura que la novedad (que ya afectó el saldo) no se sume dos veces si el activo llega al final de su vida útil en ese período. B. Persistencia del Campo es_ajuste
DataWindow: Se modificó dgr_movdepreciacion_tmp_new.srd para incluir la columna es_ajuste vinculada a la tabla compras01.mov_depreciacion_temp. Se habilitó la propiedad de actualización (Update) para este campo. Se corrige el insert de la función of_depreciacion_masiva_placa_no_deprecia para que no aplique nuevamente la novedad de ajuste en la depreciación acumulada

Script: Se agregó una validación condicional antes de insertar el registro en el DataStore temporal:
powerbuilder

```
IF luo_data.tipo_movimiento = 'ADIC' OR luo_data.tipo_movimiento = 'DISM' THEN
```

```
    ids_movdepr_tmp.setitem(ll_insert, 'es_ajuste', 'S')
```

```
ELSE
```

```
    ids_movdepr_tmp.setitem(ll_insert, 'es_ajuste', 'N')
```

```
END IF
```

5. Archivos Modificados

ompras\n_cst_depreciacion_activos_new.sru: Lógica de cálculo y asignación de variables.

dgr_movdepreciacion_tmp_new.srd: Estructura de datos para permitir la persistencia del campo es_ajuste.

6. Impacto y Validación

Precisión Contable: Los activos que terminan su vida útil con una novedad reflejan el valor exacto del activo menos el valor residual. Trazabilidad: Los reportes basados en la tabla temporal ahora pueden filtrar ajustes manuales mediante el campo es_ajuste.

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

Permanent link:
http://wiki.adacsc.co/doku.php?id=ada:sicoferp:compras:depreciacion_activos_fijos

Last update: **2026/03/31 21:36**

