

Maestro de Ubicaciones

En este maestro se configura las diferentes ubicaciones las cuales se le asignan a las placas en el registro de activos fijos.

Se lista la siguiente información:

- Código
- Descripción
- Dirección



Código	Descripción	Dirección
6199	B U-02 . PRINCIPAL CONSEJO SUPERIOR	CARRERA 10 NO 12 - 15
6203	B U-02 INSERVIBLES CONSEJO SUPERIOR	CARRERA 10 NO 12 - 15
6205	B U-02 OBSOLETOS CONSEJO SUPERIOR	CARRERA 10 NO 12 - 15
6206	BODEGA REINTEGRADOS CONSEJO SUPERIOR	CARRERA 10 NO 12 - 15
6207	B U-08. PRINCIPAL TRIBUNALES Y JUZGADOS	CARRERA 10 NO 12 - 15
6208	B U08 INSERVIBLES TRIBUNALES Y JUZGADOS	CARRERA 10 NO 12 - 15
6209	B U08 OBSOLETOS TRIBUNALES Y JUZGADOS	CARRERA 10 NO 12 - 15
6210	B U08 REINTEGRADOS TRIBUNALES Y JUZGADOS	CARRERA 10 NO 12 - 15

Los datos básicos que se deben de ingresar por ubicación son:

- Descripción
- Dirección
- Ubicación Principal
- Borrado



Datos Ubicación

Cod. Ubicación: 6199

Descripción: B U-02 . PRINCIPAL CONSEJO SUPERIOR

Dirección: CARRERA 10 NO 12 - 15

Ubicación Principal Borrado

(Nota: El campo Cod.ubicación se llena automáticamente con el consecutivo SEQ_CONSECUTIVO_EXT_UBICACION)

Documentación de Ajuste Técnico: Módulo de Ubicaciones

1. Identificación del Problema Ruta: Compras > Maestros > Maestro Ubicaciones (w_mae_ubicaciones.srw).

Síntoma: El sistema permitía modificar campos (descripción, departamento, ciudad), reportaba "Datos guardados correctamente", pero los cambios no persistían en la base de datos tras cerrar y volver a abrir el módulo. Causa Raíz: Inconsistencia en el manejo de objetos de transacción. Se realizaba el SetTransObject con ts_transaccion, pero se ejecutaban comandos de control (COMMIT / ROLLBACK) utilizando el objeto SQLCA. Al no ser el mismo canal de comunicación, los cambios quedaban en un "limbo" transaccional y se perdían al cerrar la conexión.

2. Solución Aplicada Se reestructuró el evento ue_grabar bajo los siguientes pilares técnicos:

A. Unificación del Transaction Object Se migraron todas las sentencias SQL y controles de transacción para utilizar exclusivamente ts_transaccion. Esto garantiza que el Update() y el Commit viajen por el mismo túnel de comunicación. Antes: COMMIT Using SQLCA; Después: COMMIT Using ts_transaccion;

B. Sincronización de Interfaz (Buffer) Se añadió una validación estricta al inicio del evento de grabado: Fragmento de código IF idw_registro.AcceptText() <> 1 THEN Return Esto asegura que el último dato digitado por el usuario (ej. la nueva descripción) sea validado y trasladado de la UI al buffer del DataWindow antes de iniciar el proceso de guardado. C. Optimización del Ciclo de Update Se implementó el uso de Update(TRUE, FALSE) seguido de un ResetUpdate() manual tras el éxito del COMMIT. Propósito: Evita que el DataWindow limpie sus "flags" de modificación prematuramente. Si el COMMIT falla, el usuario aún tiene los cambios en pantalla para reintentar sin perder su trabajo.

3. Fragmento de Código Corregido (Lógica Principal) Fragmento de código // Sincronizar buffer

```
IF idw_registro.AcceptText() <> 1 THEN Return
```

```
... validaciones de departamento y ciudad Actualización usando el objeto correcto IF  
idw_registro.Update(TRUE, FALSE) = 1 THEN
```

```
IF IsValid(idw_lista_ubicaciones) THEN  
    idw_lista_ubicaciones.Update(TRUE, FALSE)  
END IF  
COMMIT Using ts_transaccion; // Unificación de transacción  
IF ts_transaccion.SQLCode = 0 THEN  
    idw_registro.ResetUpdate()  
    // Refrescar UI  
ELSE  
    ROLLBACK Using ts_transaccion;  
END IF
```

```
ELSE
```

```
ROLLBACK Using ts_transaccion;
```

```
END IF
```

From:

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

Permanent link:

http://wiki.adacsc.co/doku.php?id=ada:sicoferp:compras:maestro_de_ubicaciones

Last update: **2026/04/10 14:41**

