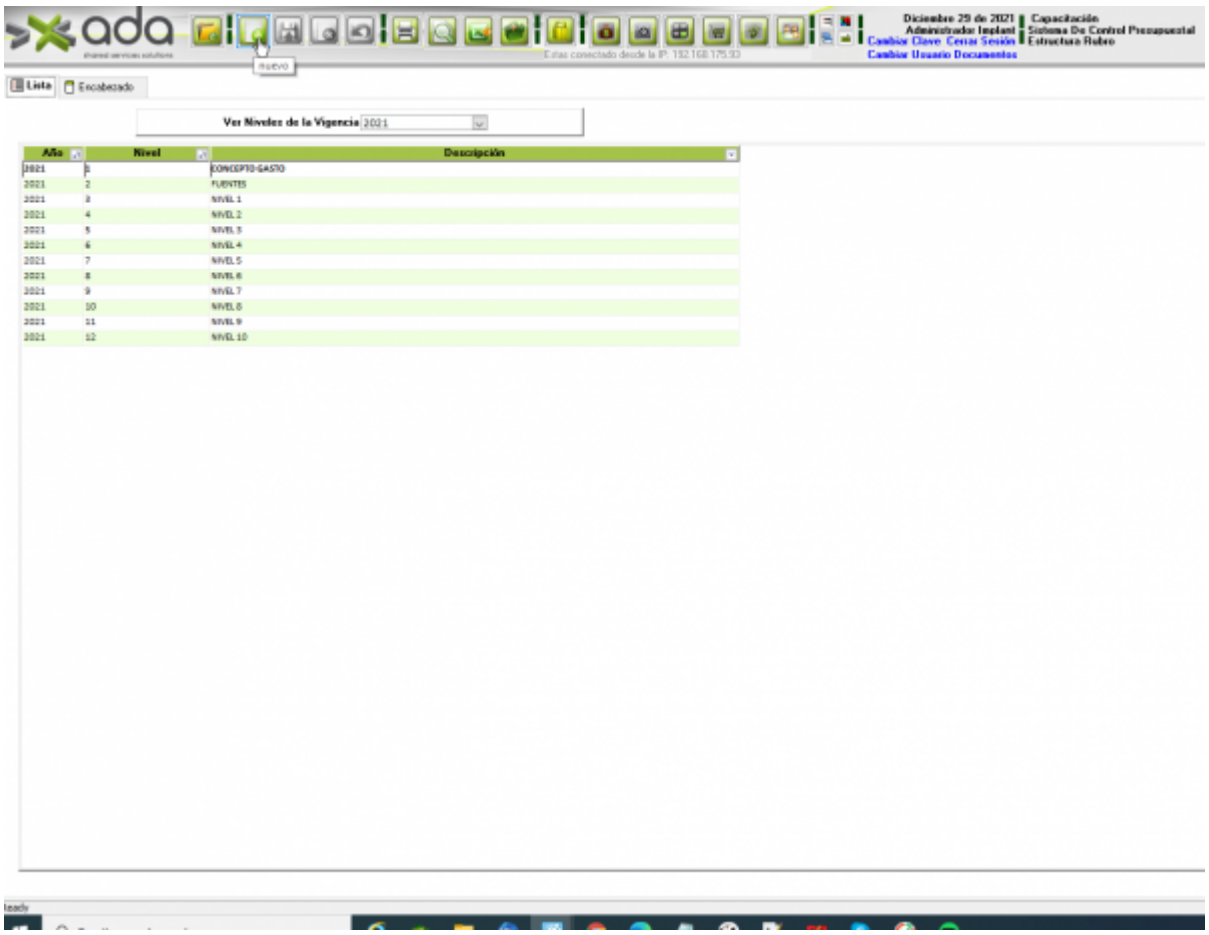
Estructura Rubro

Esta ventana permite generar o modificar la estructura de niveles de rubros de gastos.

Lista Niveles

En esta pestaña se pueden visualizar los niveles de la vigencia seleccionada.



Nivel

En esta pestaña se puede visualizar y modificar los datos de un nivel y de códigos de nivel de la estructura de rubros de gastos.

Corrección en Estructura de Rubros (Ingresos y Gastos)

1. Descripción del Problema

Se detectó que, en las ventanas de configuración de estructura de rubros, al intentar desmarcar la clasificación de un código (ej. desmarcar “Funcionamiento”), el sistema notificaba un guardado exitoso pero no persistía el cambio en la base de datos, restaurando el valor anterior al recargar.

2. Objetos Involucrados

w_niveles_rubro.srw (Estructura de Gastos) w_estructura_ingresos.srw (Estructura de Ingresos)

3. Análisis de Causa Raíz

Sobrescritura en Validación: Dentro de la función wf_valide_codigos, existía una sentencia SetItem que forzaba el valor de la columna reserva basándose en el encabezado del nivel, ignorando y pisando cualquier cambio manual realizado por el usuario en la grilla antes del Update(). Valores Inválidos: Se encontró el uso del carácter 'F' en campos lógicos (tipo char 'S'/'N'), lo que impedía el procesamiento correcto de la lógica de exclusión. Riesgo de Stack Overflow: La validación de filas se ejecutaba mediante recursividad (funcion → funcion + 1), lo que podía causar desbordamiento de pila en estructuras con gran volumen de datos.

4. Soluciones Implementadas

Persistencia de Clasificación:

Se eliminó la línea de código que forzaba el valor de reserva en la validación, permitiendo que el estado definido por el usuario en el DataWindow llegue a la base de datos. Corrección de Lógica Lógica: Se reemplazaron valores erróneos (como 'F') por el estándar 'N' en el evento itemchanged.

Optimización de Código: Se sustituyó la validación recursiva por un bucle iterativo (FOR...NEXT) en el evento ue_grabar, mejorando el rendimiento y la estabilidad del sistema. Gestión de Recursos (Monitor): Se aseguró el cierre y liberación de memoria del objeto iuo_monitor (invocación de ue_finish y destroy) en todos los flujos de salida (éxito y error) para evitar fugas de memoria y ventanas de progreso bloqueadas.