

PBtoWS - Procesos: Guía Rápida de implementación de la Arquitectura Backend

Este capítulo contiene información relacionada con el proceso de creación de componentes aplicando la [arquitectura](#) propuesta en el desarrollo backend. El objetivo de esta sección es centrar al desarrollador en los aspectos fundamentales que debe tener presente al crear componentes que serán expuestos en servicios SOAP Powerbuilder. Tener presente que sólo se explicará el proceso de implementación de la arquitectura y se excluirán los demás procesos asociados a la creación. Para más información favor consultar los pasos del [Check List Component](#)

Paso 1: Creación del Proyecto

El primer paso consiste en definir la estructura del repositorio del componente. La información relacionada con esta actividad puede ser consultada en el siguiente link [Crear Componente](#)

Paso 2: Establecer la Clase Transacción Proyecto

Una vez creada la estructura del componente el siguiente paso consiste en redefinir la clase que mapeará las conexiones de la base de datos. Esto es necesario ya que la transacción desempeña un papel fundamental para el procesamiento de la información. La clase encargada de ese proceso es **n_cst_transaction** y está en la librería **sf00core_object.pbl**. Ubíquese en el objeto **application** del proyecto y presione el botón **Additional Properties** para desplegar la ventana de propiedades adicionales del proyecto luego seleccione la pestaña **Variable Types** y en el campo **SQLCA** cambie el valor **transaction** por **n_cst_transaction** aplique los cambios y presione el botón **Ok**. De esta forma ya quedará definida la clase transaction del componente.

Paso 3: Crear las Clases Base del Componente

El siguiente paso consiste en definir las clases base de operación del componente las cuales deben ser extendidas (heredadas) del paquete de [Resolución y Orquestación de Servicios](#) de la siguiente forma:

- Extender la clase **n_cst_service** para crear la clase de invocación de servicios
- Extender la clase **n_cst_controller_process** para crear los controladores que serán utilizados por las clases invocadoras **n_cst_service**
- Extender la clase **n_cst_model** para implementar la lógica del negocio del componente, es decir en estas clases se generará el código fuente.

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

Permanent link:
<http://wiki.adacsc.co/doku.php?id=ada:tips:sicoferp:general:pbtows:procesos:guiarapidacomponente&rev=1565616422>

Last update: **2019/08/12 13:27**

