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

Este capitulo 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 explicara 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 oncesta 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 procesameinto de la información. La clase encargada de ese proceso es **n_cst_transaction** y está en la libreria **sf00core_object.pbl**. Ubiquese en el objeto **application** del proyecto y presione el botono **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 botono **Ok**. De esta forma ya quedará definida la clase transaction del componente.

## Paso 3: Crear las Clases Base del Componente

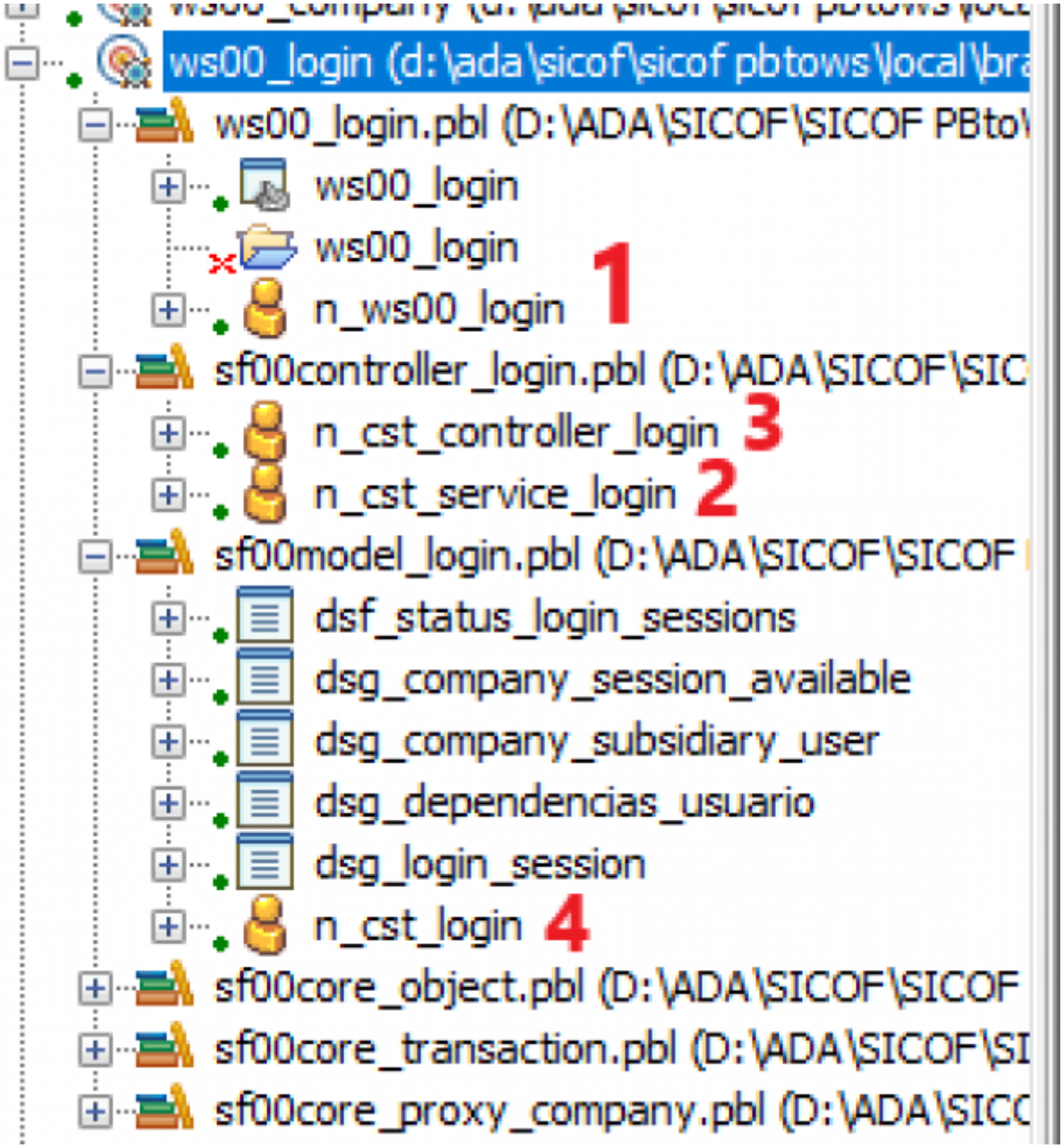
El siguiente paso consiste en definir las clases base de operación del componente las cuales deben ser extendidas (heredadas) del paquete de Resolución y Orquestación de Servicios de la sigueinte forma:

- 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**
- Extener la clase **n_cst_model** para implementar la logico del negocio del componente, es decir en estas clases se generará el código fuente.

**Nota:** Tener presente que las clases invocadoras y controladoras deben agregarse a la libreria sf(XX)controller_(xxxxxxxx).pbl y las clases del modelo ó lógica del negocio debben ser agregadas a la libreria sf(XX)model_(xxxxxxxx).pbl.

A continuación se visualiza una imagen de ejemplo con la implementación de las clases base (tener

presente el numero asociado a cada clase para identificar el tipo)




1. **Clase Lanzadora**: Es generada automáticamente al crear el proyecto en ella se registran los metodos que serán expuestos en el servicio.
2. **Clase Invocadora**: Es la clase que va a ser invocada por la Clase Lanzadora, crea los controladores y lanza los métodos que inician los procesos.
3. **Clase Controladora**: Este tipo de clases realizan validaciones, provven metodos de utilidades e invocan las clases de la lógica del negocio.
4. **Clase de Logica del Negocio**: Contiene el código powerbuilder de los procesos del ERP.

## Paso 4: Modelo de implementación de invocación por capas

A continuación se explica con ejemplos en imagenes el modelo de implementación de invocación por las capas del framework aplicando la arquitectura propuesta. El componente de referencia es el login.

# Modelo de implementación Clase Lanzadora