



**bizagi**

# Prueba Técnica de Desarrollo

Backend Development  
Engineers

## Tabla de Contenidos

Introducción.....	2
Objetivos.....	2
Funcionalidad mínima esperada por los usuarios.....	2
Evolución a futuro .....	2
Límites del problema .....	3
Preguntas.....	3
Tecnologías .....	4
Entregables .....	4
Evaluación.....	4

## Introducción

Usted acaba de ser contratado como ingeniero de desarrollo para Bizagi Modeler, la herramienta de modelamiento de procesos de Bizagi. Revisando en los foros de sugerencias de los usuarios, usted encuentra que los usuarios solicitan muy frecuentemente que Bizagi Modeler ofrezca una característica de **validación de los diagramas de procesos**. Usted, con una actitud proactiva, decide complacer a los leales usuarios de Bizagi Modeler y diseñar e implementar para ellos esta deseada característica.

Usted inicia una investigación para determinar qué clase de validaciones necesitarían los usuarios de Bizagi Modeler y encuentra este documento: The Rules of BPMN (<http://brsilver.com/the-rules-of-bpmn/>), que contiene un listado de validaciones BPMN que debería cumplir un diagrama de procesos.

## Objetivos

### Funcionalidad mínima esperada por los usuarios

Usted realiza un análisis de esta característica y define que el objetivo principal, es que un usuario mientras está trabajando con un diagrama de procesos pueda validarlo en cualquier momento, **la aplicación debe informarle al usuario qué elementos presentan errores**. Dentro del análisis realizado, usted encuentra que un diagrama de procesos puede tener cero, uno, o muchos errores de validación.

### Evolución a futuro

Como usted es un ingeniero que siempre piensa en los cambios que pueden venir más adelante, se le ocurre que en el futuro, sería interesante que en el website bizagi.com estuviera disponible **este mismo servicio de validación de diagramas de procesos como un servicio web**, de tal forma que usuarios que usen otras herramientas de modelamiento de proceso pueden usar este servicio y así atraerlos a usar Bizagi Modeler.

## Límites del problema

El estándar actual para persistencia e intercambio de diagramas de proceso es XPDL (<http://www.xpdl.org/>), que no es más que un archivo XML que contiene la información un diagrama de procesos. Para simplificar el problema, construiremos la funcionalidad sobre un archivo XPDL.

Junto con este documento encontrará los siguientes archivos:

- Sample.xpdl es un diagrama de procesos construido correctamente.
- Sample 1.xpdl es un diagrama de procesos que no cumple con la validación Style 0115.
- Sample 2.xpdl es un diagrama de procesos que no cumple con la validación BPMN 0102.
- Sample 3.xpdl es un diagrama de procesos que no cumple con la validación Style 0104.
- Sample 4.xpdl es un diagrama de procesos que no cumple con las validaciones Style 0122 y Style 0123.
- Una imagen PNG por cada XPDL que contiene el diagrama de procesos dibujado.

## Preguntas

Debe implementar una solución que **reciba un archivo XPDL y efectúe las siguientes validaciones**:

1. Style 0115
2. BPMN 0102
3. Style 0104
4. Style 0122
5. Style 0123

Estas validaciones se deben hacer sobre los cuatro archivos adjuntos a este documento.

La información de cada validación mínimo debe contener:

1. El mensaje de validación.
2. Id del elemento que presenta el error.
3. Nombre del elemento que presenta el error.
4. XPath en el XPDL del elemento que presenta el error.

A pesar que sólo debe implementar las 5 validaciones solicitadas, tenga en cuenta que la arquitectura y el diseño de la aplicación debe contemplar que **a futuro se pueden incluir más validaciones**.

La interfaz de usuario queda a su criterio, puede ser una interfaz Web, Windows, línea de comandos o una suite de pruebas unitarias.

## Tecnologías

El código se debe escribir en C#. Puede hacerlo sobre la versión de .NET que prefiera (mínimo 2.0).

## Entregables

Usted debe entregar el código fuente en una solución de Visual Studio (mínimo versión 2010), no es necesario entregar los archivos ejecutables.

## Evaluación

Los criterios de evaluación son los siguientes:

1. El correcto funcionamiento de la aplicación.

2. La arquitectura y diseño plasmado en su solución. En especial teniendo en cuenta que a futuro se pueden incluir más validaciones.
3. Estructura y organización de la solución.
4. Estándar de codificación.
5. Capacidad algorítmica.
6. Uso de la tecnología.