

# **Bizativiti**

## **Grupo 04**

### **Modelo de Casos de Prueba**

#### **Versión 1.0**

#### **Historia de revisiones**

Fecha	Versión	Descripción	Autor
14/09/2013	1.0	Creación de Documento	Diego Cossio
14/09/2013	1.0	Revisión Asistente SQA	Diego Cossio

**Responsable de verificación:** Diego Cossio

# Índice

Casos y procedimientos de Pruebas Unitarias  
Procedimientos de Pruebas de Integración  
Casos y procedimientos de Pruebas del Sistema  
Prototipo

## 1. **Casos y procedimientos de Pruebas Unitarias**

Se realizarán pruebas unitarias sobre cada componente con la herramienta

JUnit.

## 2. Procedimientos de Pruebas de Integración

Al integrar componentes o subsistemas se debe realizar lo siguiente:

Examinar y probar el sistema o subsistema obtenido según los casos de prueba.

Si el sistema o subsistema contiene objetos con error se debe determinar si dichos errores son graves o no.

- En caso que sean graves se considera que la Integración no es correcta y se debe deshacer, terminando aquí el procedimiento de Integración.
- Si no hay errores graves la Integración se considera finalizada.

## 3. Casos y procedimientos de Pruebas del Sistema

### 1. Prototipo

En el prototipo Bizativiti se van a convertir los componentes:

- Evento Start y End
- Manual Task y User Task
- InclusiveGateway y ParallelGateway
- Flow

Se definió con el equipo de verificación el siguiente conjunto de casos de pruebas :

Caso	Casos	Componente	Descripción
1	1.1	Ninguno	Proceso sin componentes
1	1.2	Eventos	Proceso con eventos inicio y fin
1	1.3	Eventos, flow	Proceso con eventos inicio, fin, conector
1	1.4	Evento, inclusivegateway	Proceso con eventos inicio, conector, compuerta inclusive
1	1.5	Evento, parallelgateway	Proceso con eventos inicio, conector, compuerta paralela
2	2.1	Eventos, UserTask	Proceso con eventos inicio , fin, 2Flows y con UserTask
2	2.2	Eventos, ManualTask	Proceso con eventos inicio , fin 2Flows y con ManualTask
3	3.1	Evento, UserTask, ManualTask, Flows	Proceso con eventos inicio, fin, 3Flows y primero UT y después MT

3	3.2	Evento, ManualTask, UserTask, Flows	Proceso con eventos inicio, fin, 3Flows y primero MT y después UT
3	3.3	Evento, ManualTask, Flows	Proceso con eventos inicio, fin, 3Flows y 2 MT
3	3.4	Evento, UserTask, Flows	Proceso con eventos inicio, fin, 3Flows y 2 UT
4	4.1	Evento, inclusivegateway, MT, Flows	Proceso con eventos inicio, fin, 2 inclusiveG, 2MT y 6Flows
4	4.2	Evento, inclusivegateway, UT, Flows	Proceso con eventos inicio, fin, 2 inclusiveG, 2UT y 6Flows
4	4.3	Evento, inclusivegateway, UT, MT, Flows	Proceso con eventos inicio, fin, 2 inclusiveG, 1UT, 1MT y 6Flows
4	4.4	Evento, inclusivegateway, MT, UT, Flows	Proceso con eventos inicio, fin, 2 inclusiveG, 1MT, 1UT y 6Flows
4	4.5	Evento, parallelgateway, UT, Flows	Proceso con eventos inicio, fin, 2 parallelG, 3UT, 7Flows
4	4.6	Evento, parallelgateway, UT, MT, Flows	Proceso con eventos inicio, fin, 2 parallelG, 2UT, 1MT, 7Flows
4	4.7	Evento, parallelgateway, UT, MT, Flows	Proceso con eventos inicio, fin, 2 parallelG, 1UT, 2MT, 7Flows
5	5.1	Evento, inclusiveG, parallelG, UT, Flows	Proceso con eventos inicio, fin, 1 inclusiveG, 1 parallelG, 2UT, 6Flows
5	5.3	Evento, inclusiveG, parallelG, MT, Flows	Proceso con eventos inicio, fin, 1 inclusiveG, 1 parallelG, 2MT, 6Flows
5	5.3	Evento, inclusiveG, parallelG, UT, MT, Flows	Proceso con eventos inicio, fin, 1 inclusiveG, 1 parallelG, 1UT, 1MT, 6Flows

A cada uno de los casos se les crea una entrada generada en bizagi y una salida esperada modelada en activity, luego a la salida esperada se la compara con la salida del prototipo.