



**bizagi**

# Patrones de Modelado de Procesos

Bizagi Suite

## Tabla de Contenido

Patrones de modelado de procesos .....	4
Implementación de patrones .....	4
Patrones de control básico de flujo .....	4
WCP 1- Secuencia.....	4
WCP 2- División Paralela.....	5
WCP 3 - Sincronización.....	6
WCP 4 – Decisión Exclusiva .....	7
WCP 5- Unión Simple .....	8
Patrones avanzados para ramificación y sincronización. ....	9
WCP 6 – Elección Múltiple.....	9
WCP 7 – Estructura de Unión Sincronizada.....	10
WCP 8 – Unión múltiple.....	11
WCP 9 – Discriminador Estructurado.....	12
WCP 30 – Estructura parcial de unión.....	13
WCP 31 – Bloqueo parcial de unión.....	14
WCP 32 – Cancelación Parcial de Unión.....	15
WCP 33 – Uniones Generalizadas.....	17
WCP 37 – Sincronización a cíclica de unión .....	18
WCP 38 – Sincronización de Unión General.....	19
WCP 41 – Unión de Caminos .....	21
WCP 42 – División del Camino .....	22
Patrones de Múltiples Instancias.....	24

WCP 12 – Múltiples Instancias sin Sincronización. ....	24
WCP 13 - Múltiples instancias con conocimiento a priori del número de ejecuciones en la etapa de diseño. ....	26
WCP 14 – Múltiples instancias con conocimiento en el tiempo de ejecución .27	
WCP 15 – Múltiples Instancias sin conocimiento el tiempo de diseño .....	29
WCP 34 – Unión Parcial Estática de múltiples Instancias .....	31
WCP 35 – Cancelación parcial de unión de múltiples instancias .....	32
WCP 36 – Unión parcial dinámica de múltiples instancias .....	34
Diagrama 24. Ejemplo Unión parcial dinámica de múltiples instancias .....	35
Patrones basados en Eventos.....	36
WCP 16 – Decisión Implícita .....	36
WCP 17 – Ejecución Paralela Intercalada.....	37
WCP 18 - Milestone .....	38
WCP 39 – Sección Crítica. ....	39
WCP 40 Ejecución Intercalada.....	40
Patrones de Cancelación y de Terminaciones Forzadas.....	41
WCP 19 – Cancelación de tareas .....	41
WCP 20 – Cancelación del Caso.....	42
WCP 25 – Cancelación de una Región .....	43
WCP 26 – Cancelación de Múltiples Instancias de una Actividad .....	44
WCP 27 – Completar Múltiples Instancias de una Actividad .....	45
Patrones de Iteraciones .....	45
WCP 26 – Ciclos Arbitrarios.....	45
WCP 10 – Bucle Estructurado.....	46
WCP 22 - Recursividad.....	47
Patrones de Terminación .....	48

WCP 11 Terminación Implícita.....	48
WCP 43 - Terminación Explícita .....	49
Patrones de Activación o de disparo.....	50
WCP 23 – Activación Transitoria .....	50
WCP 24 – Activación Persistente.....	51
REFERENCIAS .....	52

## Patrones de modelado de procesos

### Implementación de patrones

En este documento explicaremos como implementar los patrones de modelado de procesos propuestos por el profesor Van Der Aalst utilizando el Modelador de Procesos de Bizagi. Estos patrones son muy prácticos ya que describen situaciones que comúnmente se encuentran en los procesos de negocio.

A continuación encontrara algunos ejemplos de cada patrón para que pueda entender su funcionamiento y aplicación.

### Patrones de control básico de flujo

Este grupo de patrones muestran los aspectos básicos para el control del flujo de procesos.

#### WCP 1- Secuencia

##### Descripción

Este patrón es utilizado para modelar dependencia entre tareas, es decir, una tarea no puede empezar hasta que otra no haya terminado (ejecución en secuencia) [2].

##### Ejemplo

Cuando una compañía de seguros recibe un reclamo en alguna póliza, es necesario realizar varias actividades con un orden definido. Primero, el cliente debe reportar el reclamo, luego, éste debe ser evaluado para poderme autorizar el pago y finalmente se desembolsa la cantidad reclamada.

No tendría sentido que una reclamación sea evaluada antes de ser reportada, o pagar el dinero de la reclamación antes de evaluarla.

##### Implementación

Para modelar este patrón es necesario conectar las actividades (en el orden de ejecución definido) utilizando conectores de flujo de secuencia como se muestra en el diagrama 1.

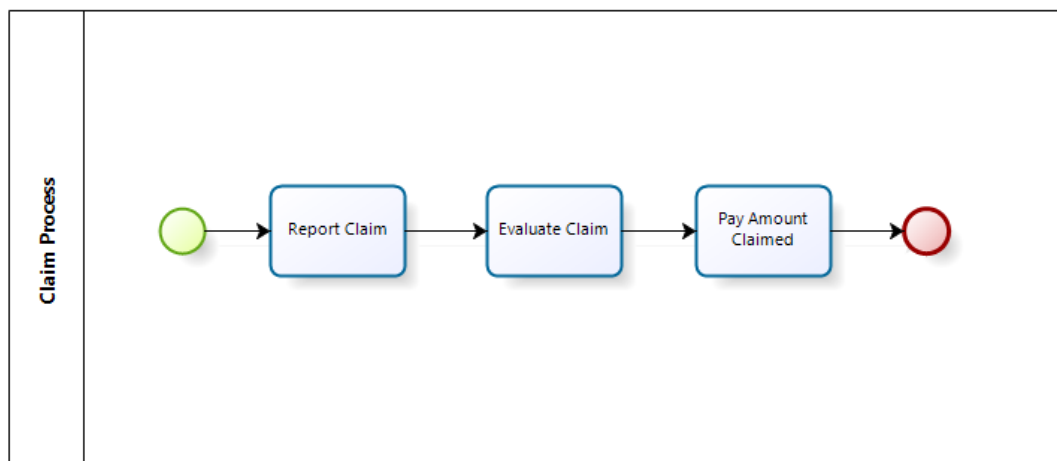


Diagrama 1. Ejemplo del patrón Secuencia

## WCP 2- División Paralela.

### Descripción

Una división paralela es un punto del proceso donde un camino es dividido en dos o más ramas paralelas las cuales son ejecutadas al mismo tiempo [2].

### Ejemplo

Cuando un nuevo empleado llega a la compañía es necesario realizar varias actividades, por ejemplo, darle acceso a la información de su cargo, firmar algunos documentos legales y alistar su puesto de trabajo.

### Implementación

Para implementar la División Paralela, es necesario utilizar la Compuerta Paralela. Esta compuerta activa caminos alternativos sin verificar condiciones.

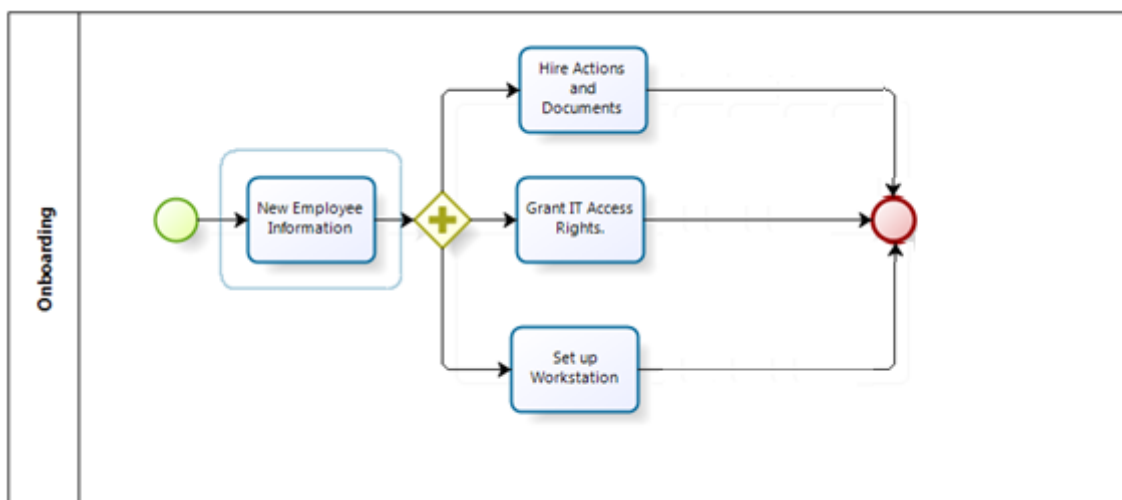


Diagrama 2. Ejemplo del patrón División Paralela

## WCP 3 - Sincronización

### Descripción

La sincronización es un punto en el proceso donde dos o más ramas del proceso se unen en una sola. Se llama sincronización porque se espera a que todas las ramas entrantes se completen antes de continuar con la siguiente actividad [2].

### Ejemplo

Cuando un nuevo empleado llega a la compañía es necesario realizar varias actividades, por ejemplo, darle acceso a la información de su cargo, firmar algunos documentos legales y alistar su puesto de trabajo.

El empleado no puede empezar a trabajar hasta que todas las actividades se hayan completado.

### Implementación

Para este ejemplo vamos a utilizar una Compuerta Paralela como elemento convergente para lograr la unión de todas las ramas.

El patrón de sincronización también puede ser modelado utilizando compuertas inclusivas o exclusivas dependiendo de los requerimientos del negocio.

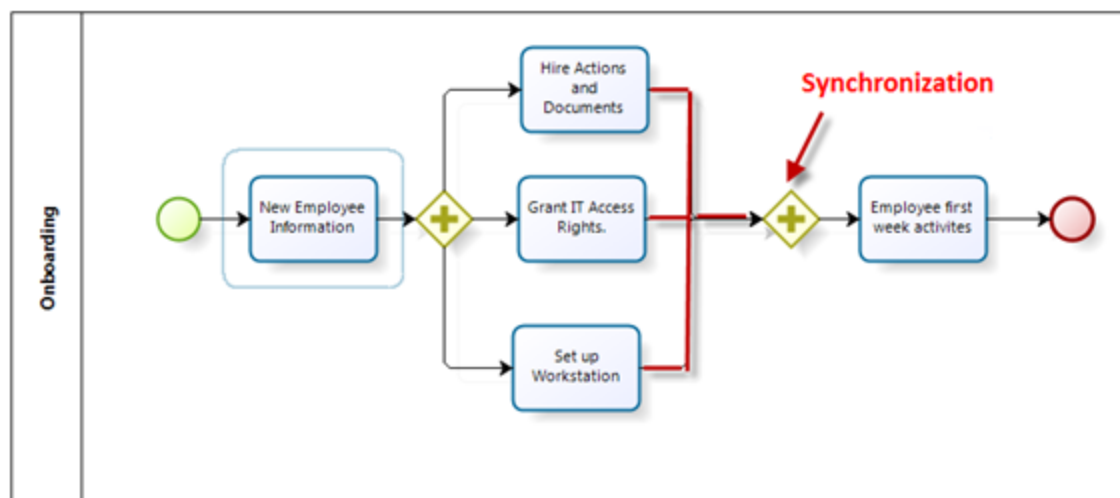


Diagrama 3. Ejemplo del patrón Sincronización

## WCP 4 – Decisión Exclusiva

### Descripción

La decisión exclusiva representa un punto en el proceso donde se debe escoger un solo camino de varios disponibles dependiendo de una decisión o de datos del proceso [2].

### Ejemplo

El Departamento de Contabilidad recibe y paga las facturas de la compañía. El pago de una factura se puede realizar a través de transferencia bancaria, cheque o tarjeta de crédito. Solo es posible realizar el pago de la factura con un medio de pago.

### Implementación

El patrón de decisión exclusiva puede ser modelado a través de una Compuerta Exclusiva. Para este caso la compuerta tiene tres secuencias de flujo pero el proceso solo debe continuar por uno de ellos de acuerdo al cumplimiento de la condición definida.



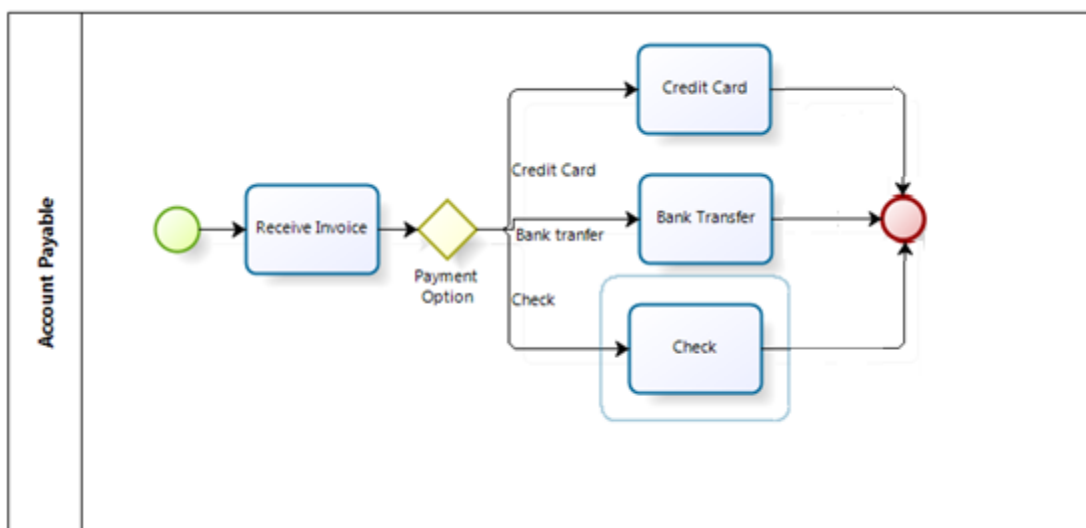


Diagrama 4. Ejemplo del patrón Decisión Exclusiva

## WCP 5- Unión Simple

### Descripción

La unión simple es un punto del proceso donde dos o más caminos alternativos convergen en uno solo. Es un supuesto de este patrón que los caminos alternativos no son ejecutados en paralelo [2].

### Ejemplo

Tomemos el ejemplo que utilizamos en el patrón WCP 4. Suponga que luego de realizar el pago con alguna de las opciones disponibles el ERP financiero es actualizado. Note en el diagrama 4 que la actividad *Actualizar ERP Financiero* solo será ejecutada una vez, esto debido a que solo una de las ramas entrantes es activada.

### Implementación

Para realizar la implementación de este patrón se utilizan conectores de flujo de secuencia para conectar las actividades de las ramas entrantes con la siguiente actividad.

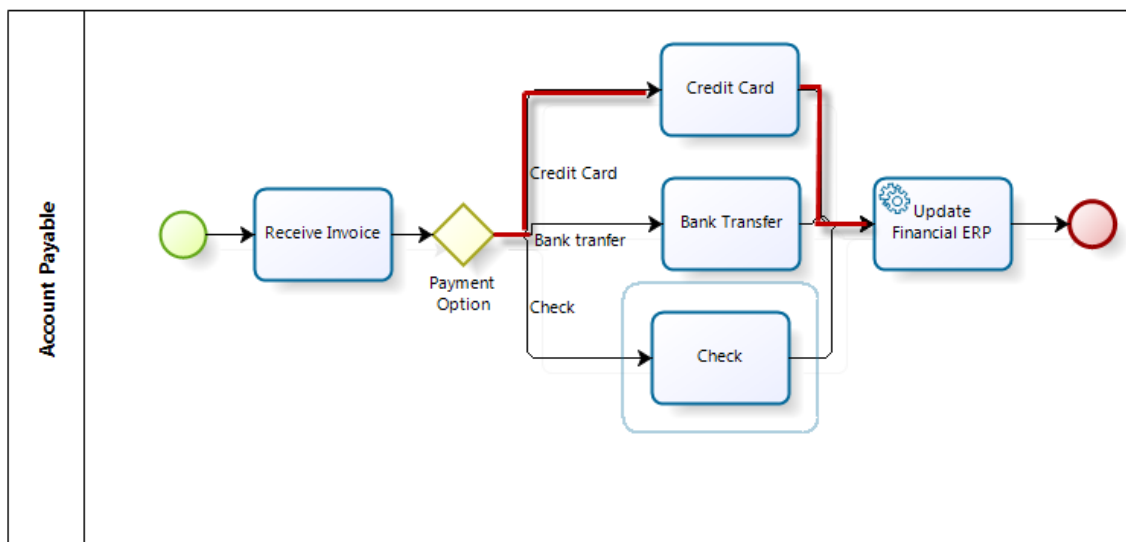


Diagrama 5. Ejemplo del patrón Unión Simple

## Patrones avanzados para ramificación y sincronización.

Este grupo de patrones presenta situaciones donde es necesario utilizar conceptos más complejos para realizar ramificación y sincronización de caminos.

### WCP 6 – Elección Múltiple.

#### Descripción

El patrón de elección múltiple es utilizado para modelar puntos en el flujo de proceso donde varios caminos son escogidos dependiendo de datos del proceso y/o puntos de decisión [2].

#### Ejemplo

Durante un proceso de auditoria es muy común encontrar no conformidades dentro del proceso auditado. Las no conformidades deben ser evaluadas y corregidas por el dueño del proceso.

La corrección de una no conformidad puede ser realizada de diferentes formas, por ejemplo, con una acción correctiva, una acción preventiva, una acción inmediata o una combinación entre las opciones anteriores.

## Implementación

Para implementar el patrón de elección múltiple se utiliza una Compuerta Inclusiva. Ésta compuerta es utilizada para activar uno o más caminos dependiendo de datos del proceso.

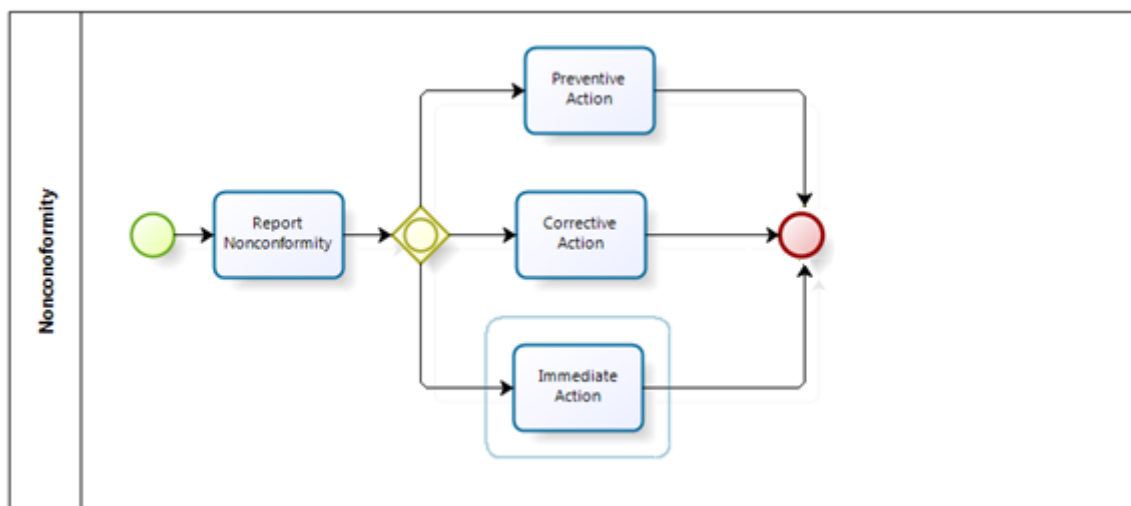


Diagrama 6. Ejemplo del patrón de Elección Múltiple

## WCP 7 – Estructura de Unión Sincronizada.

### Descripción

Es un punto en el proceso donde múltiples caminos que fueron activados antes en el proceso convergen en una sola rama, el proceso con continua hasta que todas las actividades de las ramas entrantes hayan terminado [2].

### Ejemplo

Continuando con el ejemplo utilizado en WCP 4, suponga ahora que la no conformidad no puede ser cerrada hasta que las actividades que fueron seleccionadas hayan terminado.

## Implementación

Para implementar el patrón se deben utilizar dos compuertas inclusivas, una como elemento divergente (activar algunas de las ramas salientes) y otro como elemento de sincronización o convergente (para esperar que todas las ramas activadas lleguen a un determinado punto para continuar con el proceso).

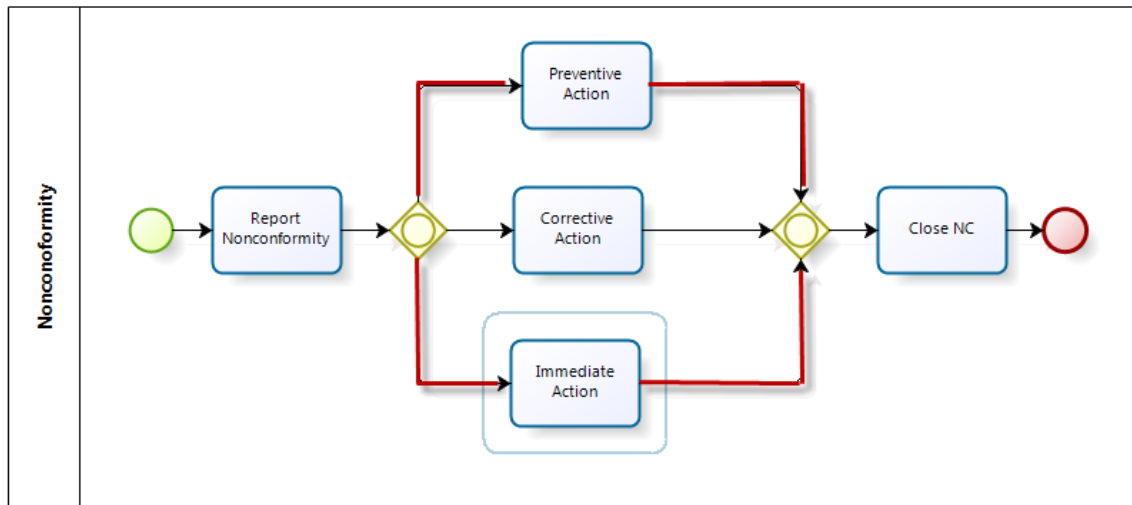


Diagrama 7. Ejemplo del patrón de Estructura de Unión Sincronizada

## WCP 8 – Unión múltiple.

### Descripción

El patrón de unión múltiple es usado para converger dos o más ramas en un solo camino. Cada activación de una de las ramas entrantes resulta en la activación de la siguiente actividad en el proceso [2].

### Ejemplo

Durante el Proceso de Selección de personal es necesario verificar las referencias del nuevo empleado. Es muy importante verificar las referencias personales y laborales suministradas. Cada vez que una referencia es verificada, el Jefe de Recursos Humanos debe ser notificado.

### Implementación

El patrón utiliza una compuerta paralela para habilitar los dos caminos. Cada vez que la actividad de uno de los caminos es terminada, la siguiente actividad, en este caso *Informar sobre Referencias*, es ejecutada.

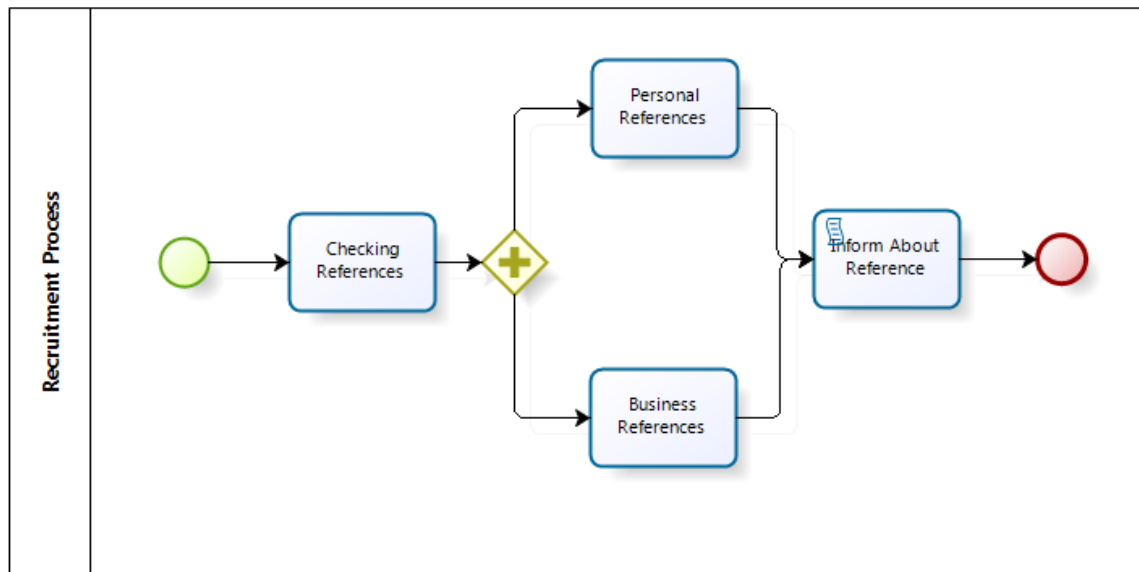


Diagrama 8. Ejemplo del patrón Unión Múltiple

## WCP 9 – Discriminador Estructurado.

### Descripción

Este patrón describe un punto en el proceso donde se espera a que una de las ramas entrantes haya sido completada antes de continuar con la siguiente actividad, las otras ramas son omitidas después de ser completadas. Una vez todas las ramas entrantes han sido completadas, el discriminador se reinicia [2].

### Ejemplo

Un empleado solicita un préstamo a su compañía. Para otorgar el crédito es necesario que la solicitud sea aprobada por su jefe o por el departamento financiero. Cuando alguno de los dos de su aprobación, el dinero puede ser desembolsado al empleado.

### Implementación

Este patrón puede utilizar compuertas Paralelas o Inclusivas para dividir los caminos. Para modelar el discriminador es necesario utilizar una Compuerta Compleja la cual espera hasta que una de las dos ramas entrantes sea completada.

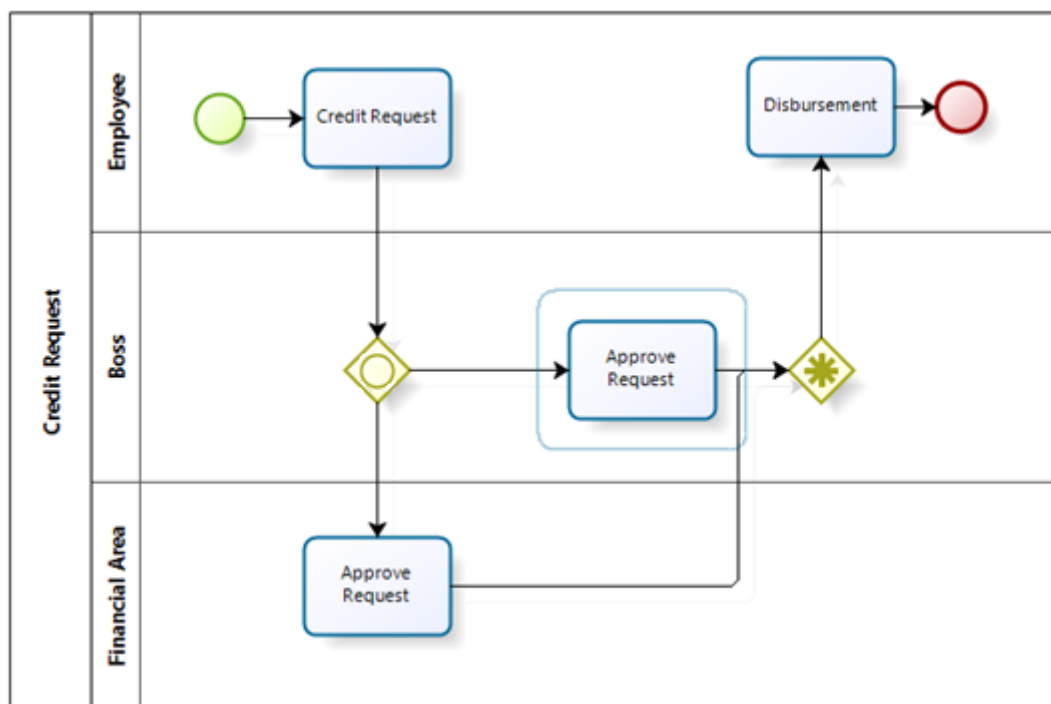


Diagrama 9 Ejemplo del patrón Discriminador Estructurado

## WCP 30 – Estructura parcial de unión.

### Descripción

El patrón modela la convergencia de M ramas que han sido habilitadas en un punto anterior del proceso en un solo camino. El proceso continua cuando N de las M ramas han sido completadas [2].

### Ejemplo

Un empleado solicita un préstamo a la compañía donde labora. Para otorgar el crédito es necesario que la solicitud sea aprobada por su jefe y por el departamento financiero.

### Implementación

Esta compuerta puede utilizar una Compuerta Paralela o Inclusiva para dividir los caminos. Para modelar la estructura parcial de unión se utiliza una Compuerta Compleja la cual espera hasta que las dos ramas entrantes sean completadas.

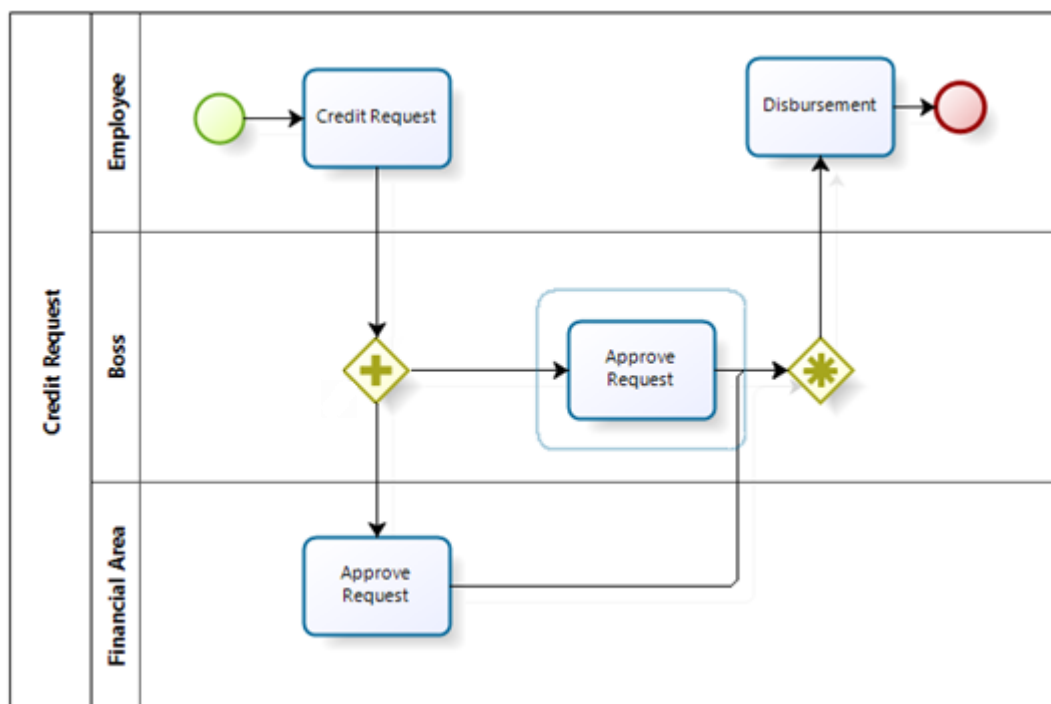


Diagrama 10. Ejemplo de Estructura parcial de unión

## WCP 31 – Bloqueo parcial de unión

### Descripción

El patrón representa la convergencia de dos o más ramas, que han sido activadas antes en algún punto del proceso, en un solo camino. El proceso continúa cuando N de las M ramas habilitadas anteriormente han terminado.

La unión es reiniciada cuando todas las ramas entrantes han sido ejecutadas una vez en la misma instancia del proceso. La ejecución de las ramas entrantes se bloquea hasta que el elemento de unión ha sido reiniciado.

### Ejemplo

Una compañía decide invertir en propuestas de negocios. Estas propuestas son enviadas por estudiantes de varias universidades, cada una de ellas es evaluada de forma individual por 3 jueces. Si 2 de los 3 jueces dan su aprobación, el aspirante será informado acerca de la decisión.

Cuando el último de los jueces decida sobre la propuesta, será posible empezar a estudiar una nueva.

## Implementación

El patrón utiliza una compuerta Paralela o una compuerta Inclusiva para dividir la rama entrante. Una compuerta compleja es utilizada para validar el número de aprobaciones (N) necesarias para informar sobre la decisión.

Para bloquear las ramas entrantes hasta que la unión sea reiniciada, se utiliza un Evento Condicional. El evento evalúa si hay alguna aprobación en progreso, y bloqueará la entrada de nuevas propuestas hasta que todos los jueces hayan dado su veredicto de la propuesta en curso.

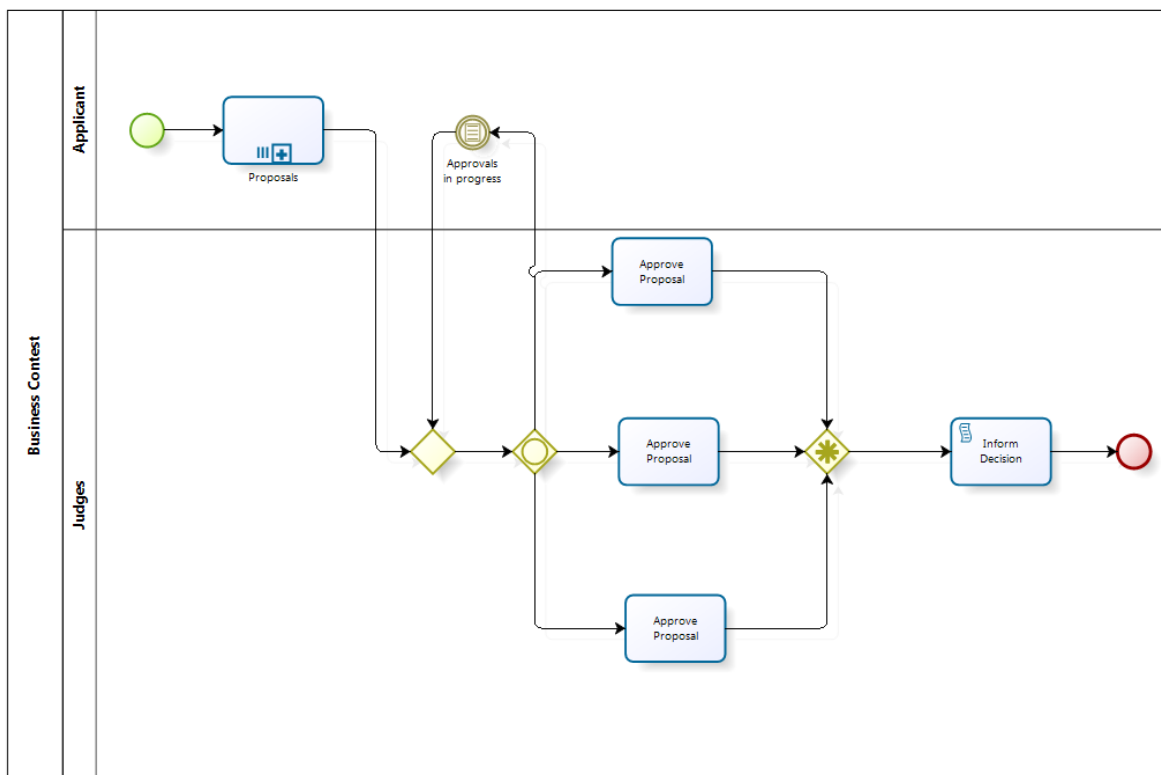


Diagrama 11. Ejemplo de Bloqueo Parcial de Unión.

## WCP 32 – Cancelación Parcial de Unión.

### Descripción

El patrón describe la convergencia de dos o más ramas que han sido habilitadas en un punto anterior del proceso en un solo camino. El control del flujo es pasado a la siguiente actividad una vez N de las M ramas han sido activadas. La activación de la unión cancela todas las ramas entrantes que no han sido terminadas.



## Ejemplo

Una compañía decide invertir en propuestas de negocios. Estas propuestas son enviadas por estudiantes de varias universidades, cada una de ellas es evaluada de forma individual por 3 jueces. Si 2 de los 3 jueces dan su aprobación, el aspirante será informado acerca de la decisión, la aprobación restante será desactivada y se empezara el proceso de aprobación con una nueva propuesta.

## Implementación

El patrón puede utilizar una compuerta Paralela o una compuerta Inclusiva para dividir la rama entrante. Una compuerta compleja es utilizada para validar el número de aprobaciones (N) necesarias para informar sobre la decisión (para continuar con el proceso).

Para bloquear las ramas entrantes hasta que la unión sea reiniciada se utiliza un Evento Condicional. El evento evalúa si hay alguna aprobación en progreso, y bloqueará la entrada de nuevas propuestas hasta N de los jueces hayan dado su veredicto de la propuesta en curso.

Una vez las N de las M ramas necesarias han sido terminadas, un evento escalable es lanzado para cancelar las tareas de aprobación pendientes.

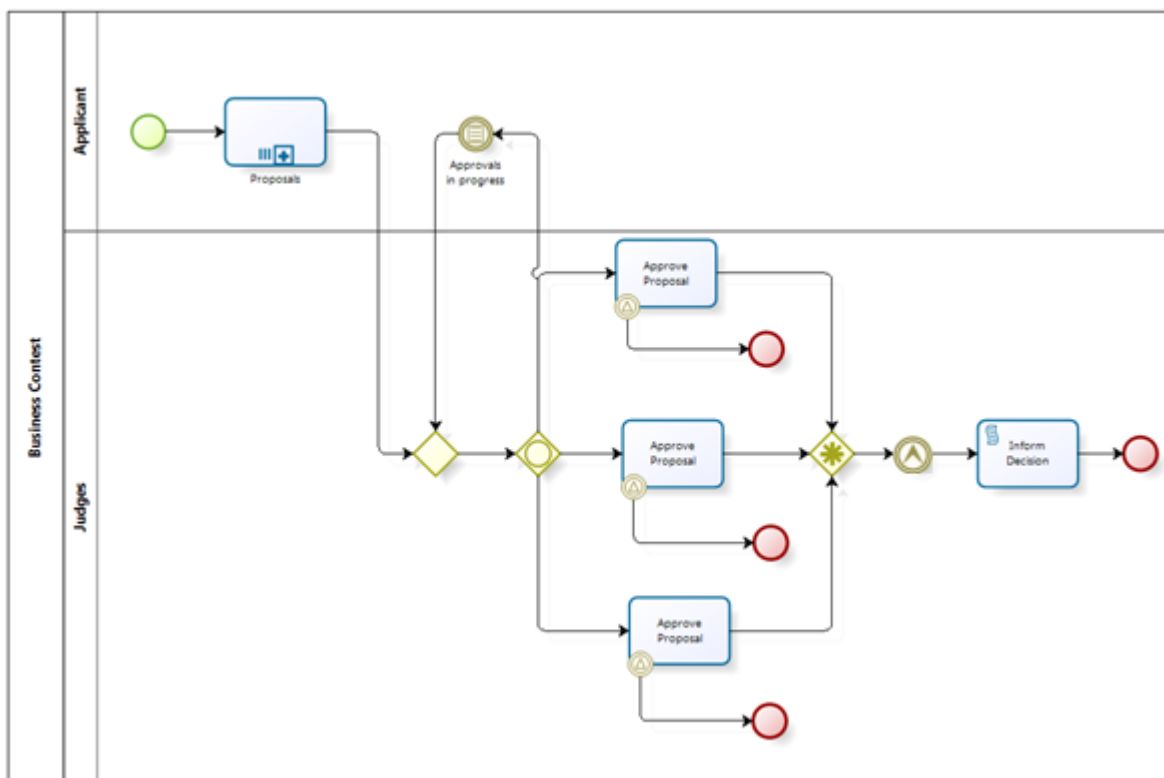


Diagrama 12. Ejemplo de Cancelación Parcial de Unión

## WCP 33 – Uniones Generalizadas

### Descripción

El patrón de generalización de uniones describe como unir dos o más ramas que han sido habilitadas en un punto anterior del proceso en un solo camino de salida. A diferencia del patrón de sincronización, éste soporta situaciones donde uno o más caminos entrantes reciben múltiples activaciones de una misma instancia del proceso [2].

### Ejemplo

Cuando un nuevo empleado llega a la compañía es necesario realizar varias actividades, por ejemplo, darle acceso a la información de su cargo, firmar algunos documentos legales y alistar su puesto de trabajo.

El empleado no puede empezar a trabajar hasta que todas las actividades hayan sido completadas.

## Implementación

Para este ejemplo vamos a utilizar una Compuerta Paralela como elemento convergente para lograr la unión de todas las ramas que fueron activadas anteriormente.

El patrón de sincronización también puede ser modelado utilizando compuertas Inclusivas o Exclusivas dependiendo de los requerimientos del negocio.

La sincronización de todos los casos del sub proceso es realizado internamente, esto significa que el sub proceso se considera terminado una vez haya cumplido la condición o propiedad establecida en su definición. Cuando esto ocurra, su camino alcanzará la compuerta paralela de convergencia y está esperara por los otros caminos activados para continuar con el flujo del proceso.

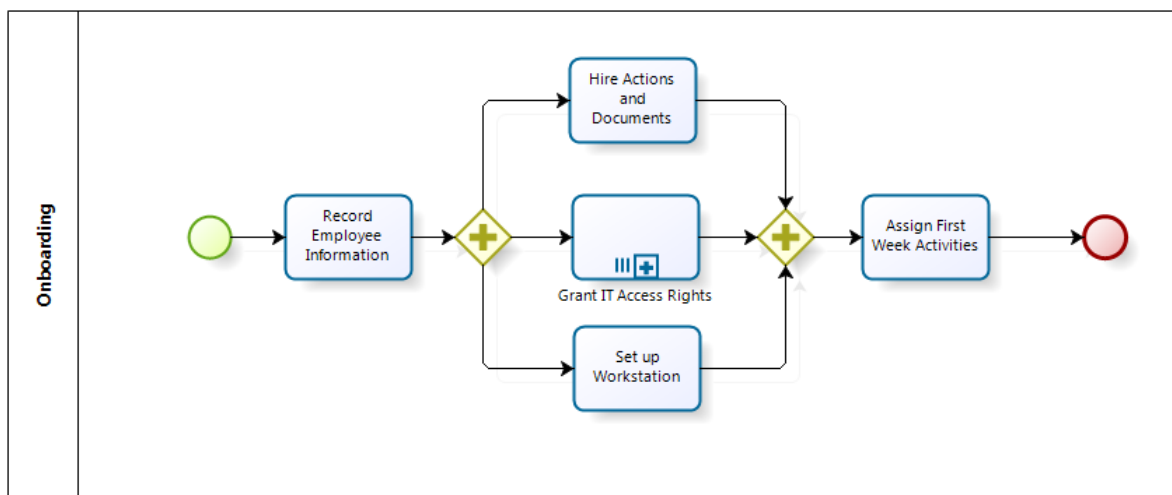


Diagrama 13. Ejemplo de Uniones Generalizadas

## WCP 37 – Sincronización a cíclica de unión

### Descripción

El patrón describe la convergencia de dos o más ramas, habilitadas en un punto anterior del proceso, en un solo camino. La determinación del número de ramas necesarias para la sincronización es realizada de acuerdo a la información disponible del proceso.

### Ejemplo

El departamento de Mercadeo de una compañía puede realizar su campaña publicitara a través de tres medios de comunicación: radio, televisión y periódico. Si

el departamento decide realizar la campaña a través de radio o televisión, los anuncios son emitidos al mismo tiempo. Si Mercadeo decide usar el periódico, puede decidir si la campaña es lanzada al mismo tiempo u otro momento.

Implementación.

Este patrón puede ser modelado utilizando Eventos Condicionales. Estos eventos son usados para coordinar la sincronización entre los caminos necesarios sin necesidad de forzar la sincronización de todos los caminos.

Los eventos condicionales permiten al flujo del proceso continuar cuando las actividades indicadas han sido terminadas.

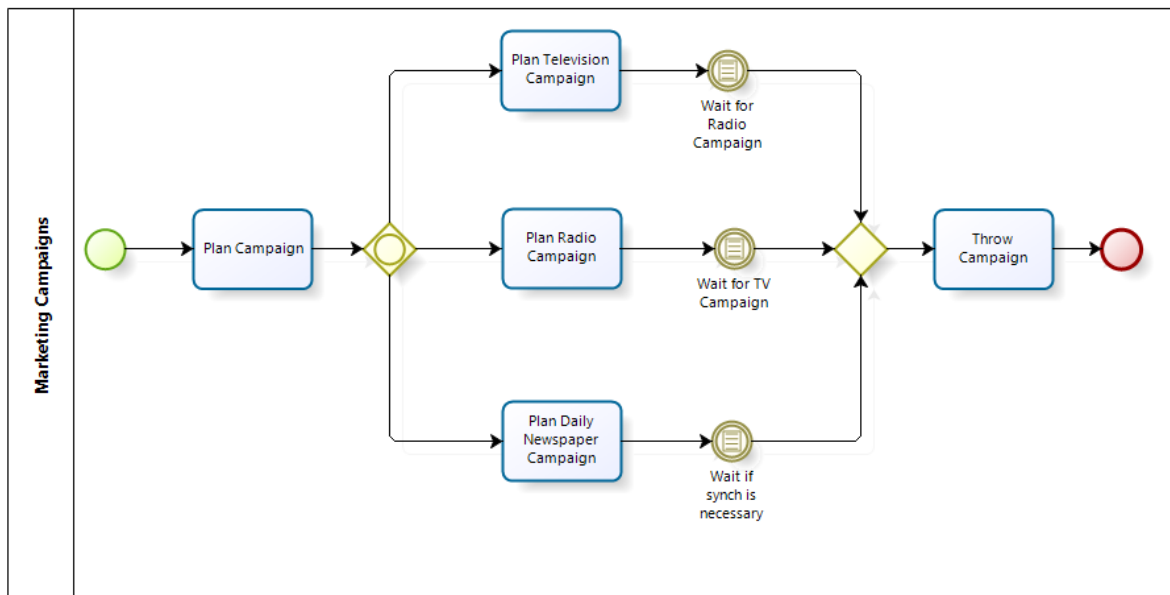


Diagrama 14. Ejemplo de sincronización a cíclica de unión.

## WCP 38 – Sincronización de Unión General.

Descripción

El patrón describe la convergencia de dos o más ramas que han sido abiertas antes en un solo camino. El proceso continúa cuando todas las ramas entrantes han sido activadas o no es posible que alguna de ellas se active en el futuro [2].

## Ejemplo

Una pareja se va a casar. Una agencia de ceremonias de matrimonio coordina las actividades principales durante ese día especial. La agencia incluye la planeación de la boda, recibir la pareja que se va a casar, sus padres y hermanos el día de la ceremonia. Sin embargo los hermanos de la pareja se encuentran en el exterior, por tanto se espera que lleguen en el último momento. Si ellos no pueden llegar a tiempo, no se les debe esperar y la boda debe comenzar sin ellos.

## Implementación

El patrón utiliza eventos de interrupción. Este tipo de eventos se adjuntan a los límites de la actividad y activan un flujo especial.

Dependiendo de las personas que asistan a la ceremonia, la compuerta Inclusiva habilita la recepción. Una o más actividades tienen eventos de interrupción adjuntos para permitir que el proceso continúe si se presenta un problema en la actividad, en este caso la situación es la posibilidad de que los hermanos no lleguen a tiempo.

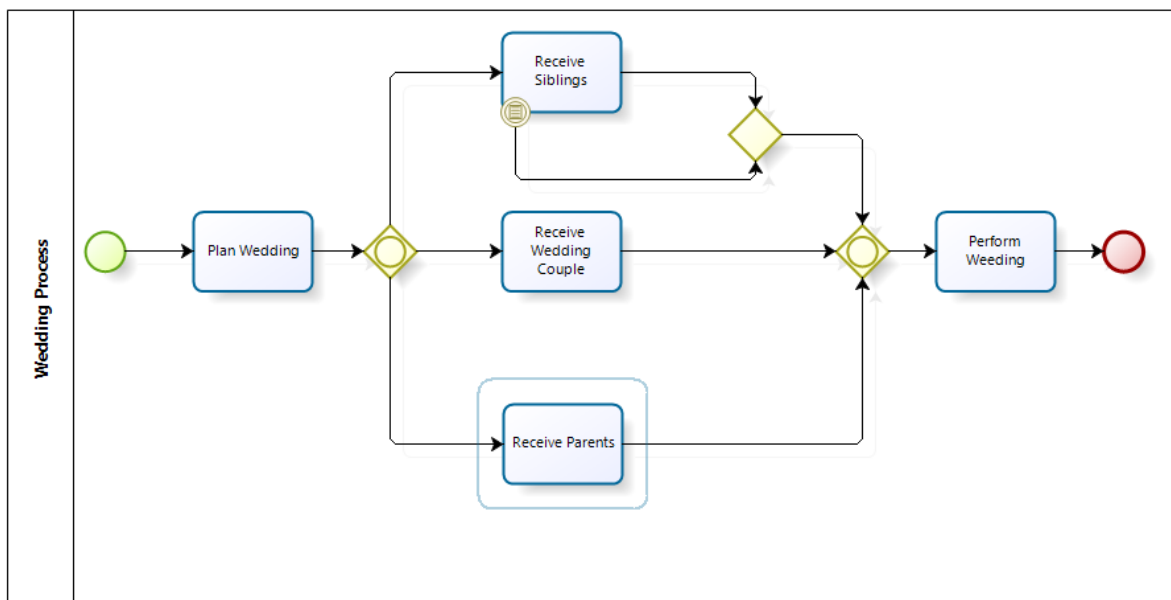


Diagrama 15. Ejemplo del patrón de Sincronización de unión general

## WCP 41 – Unión de Caminos

### Descripción

El patrón describe un punto en el proceso donde un número de ejecuciones, de un mismo camino en una misma instancia del proceso, convergen en un solo camino de ejecución [2].

### Ejemplo

Una compañía decide evaluar la percepción del ambiente de trabajo entre los empleados. Una encuesta es enviada a 100 empleados. Cuando todos los empleados hayan enviado sus resultados, el analista puede evaluar la información.

### Implementación

Para implementar este patrón es necesario utilizar un sub proceso para crear las 100 instancias de la actividad *Completar encuesta*. En este caso, el sub proceso *Enviar Encuesta* es instanciado 100 veces.

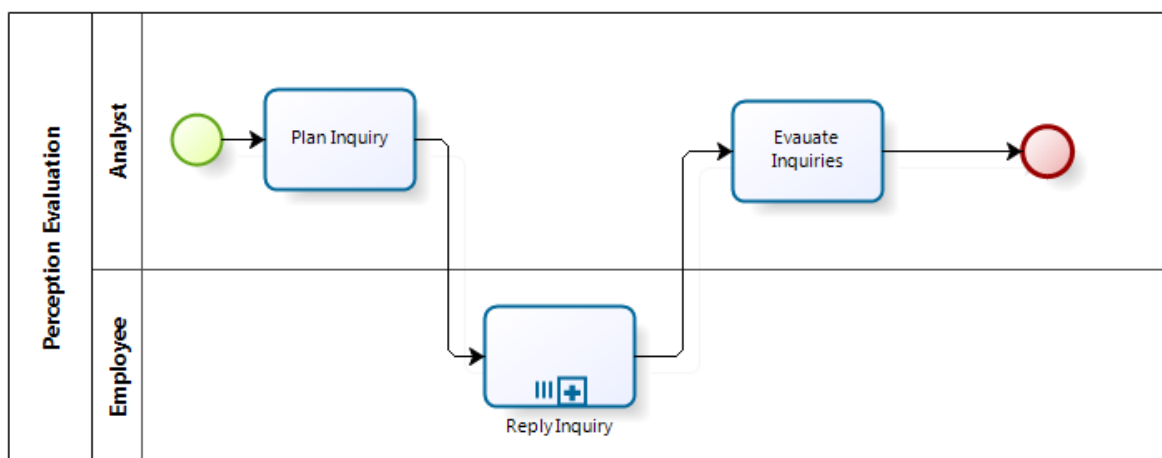
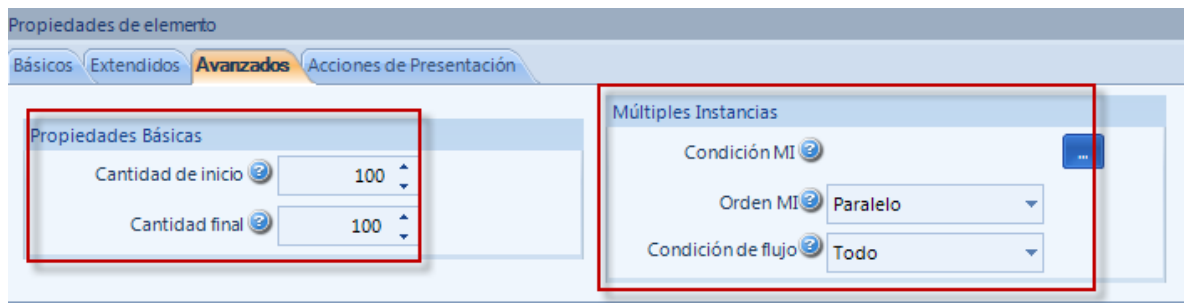


Diagrama 16. Ejemplo del patrón Unión de Caminos.

Para configurar el comportamiento del sub proceso para la condición que se desea, es necesario definir las siguientes propiedades:



Propiedades de elemento

Básicos Extendidos **Avanzados** Acciones de Presentación

**Propiedades Básicas**

Cantidad de inicio 100

Cantidad final 100

**Múltiples Instancias**

Condición MI

Orden MI Paralelo

Condición de flujo Todo

Cantidad de Inicio = 100: Esto significa que el proceso debe ser instanciado 100 veces.

Cantidad Final= 100: El sub proceso se considera como terminado cuando se hayan completado todas (en este caso) las 100 instancias.

Condición MI= Ninguna. Esto quiere decir que no hay condición para el número de instancias para el sub proceso.

Orden MI = Paralelo: Las instancias del sub proceso serán ejecutadas en paralelo.

Condición de Flujo = Todo: El proceso continua cuando todas las instancias del sub proceso han terminado.

## WCP 42 – División del Camino

### Descripción

El patrón describe un punto del proceso donde un determinado número de ejecuciones de un mismo camino deben ser instanciados [2].

### Ejemplo

Una compañía decide evaluar la percepción del ambiente de trabajo entre los empleados. Una encuesta es enviada a 100 empleados. Una vez se responden todas las encuestas, estas son evaluadas por el analista.

### Implementación.

Para implementar este patrón es necesario utilizar un sub proceso para crear las 100 instancias de la actividad *Completar encuesta*. En este caso, el sub proceso *Enviar Encuesta* es instanciado 100 veces.

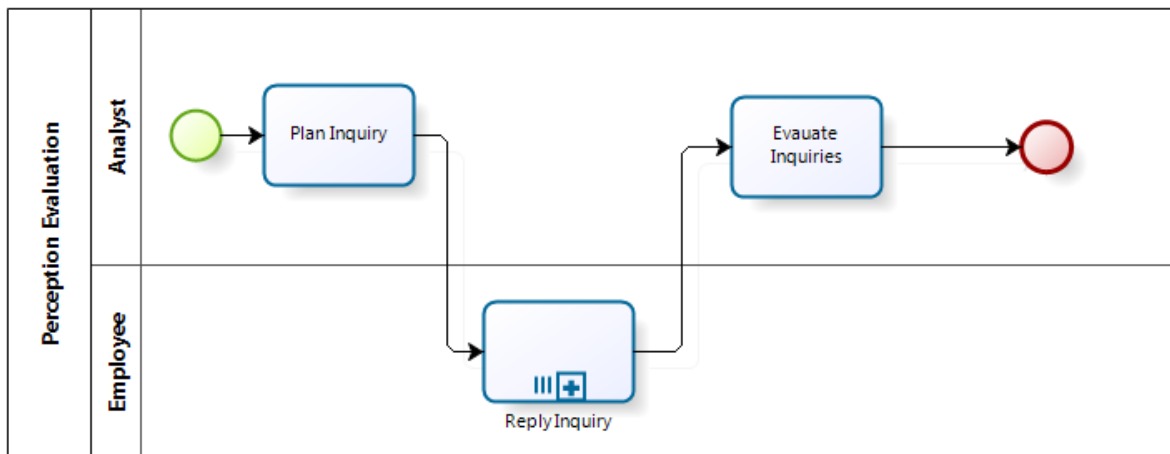
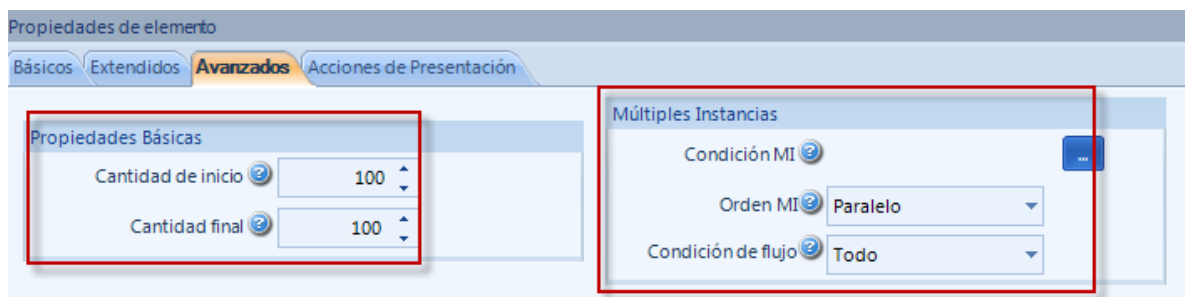


Diagrama 17. Ejemplo del patrón división del camino

Para configurar el comportamiento del sub proceso para la condición que se desea, es necesario definir las siguientes propiedades:



Cantidad de Inicio = 100: Esto significa que el proceso debe ser instanciado 100 veces.

Cantidad Final= 100: El sub proceso se considera como terminado cuando se hayan completado todas (en este caso) las 100 instancias.

Condición MI= Ninguna. Esto quiere decir que no hay condición para el número de instancias para el sub proceso.

Orden MI = Paralelo: Las instancias del sub proceso serán ejecutadas en paralelo.

Condición de Flujo = Todo. El proceso continua una vez todas las instancias del sub proceso han finalizado.



## Patrones de Múltiples Instancias

Los patrones de múltiples instancias describen situaciones donde hay varias ejecuciones de una misma actividad durante el proceso (donde comparten la misma implementación) [1]. Múltiples instancias pueden ser necesarias por las siguientes tres razones.

1. Una actividad puede iniciar múltiples instancias de sí mismo cuando se activa.
2. Una actividad es inicializada muchas veces debido a la recepción de diferentes disparadores, por ejemplo como parte de un bucle.
3. Uno o dos actividades del proceso comparten la misma definición de implementación.

### WCP 12 – Múltiples Instancias sin Sincronización.

#### Descripción

El patrón es utilizado para modelar actividades que han sido instanciadas muchas veces en el proceso y no necesitan ser sincronizadas para que el flujo continúe. Este patrón también es conocido como MultiCamino sin sincronización o "Spawn of facility".

#### Ejemplo

Una compañía está planeando implementar un nuevo proyecto. Las partes interesadas (Stakeholders) son informadas sobre el alcance del proyecto y pueden enviar sus comentarios o sugerencias sobre él si así lo desean. El proceso continúa aunque no se reciba ninguna sugerencia.

#### Implementación

Para modelar este proceso es necesario utilizar un sub proceso que crea múltiples instancias de una actividad. El sub proceso *Sugerencias* es habilitado para cada una de las partes interesadas con el objetivo de que hagan sus comentarios. El proceso continúa con su flujo normal cuando el sub proceso ha sido activado.

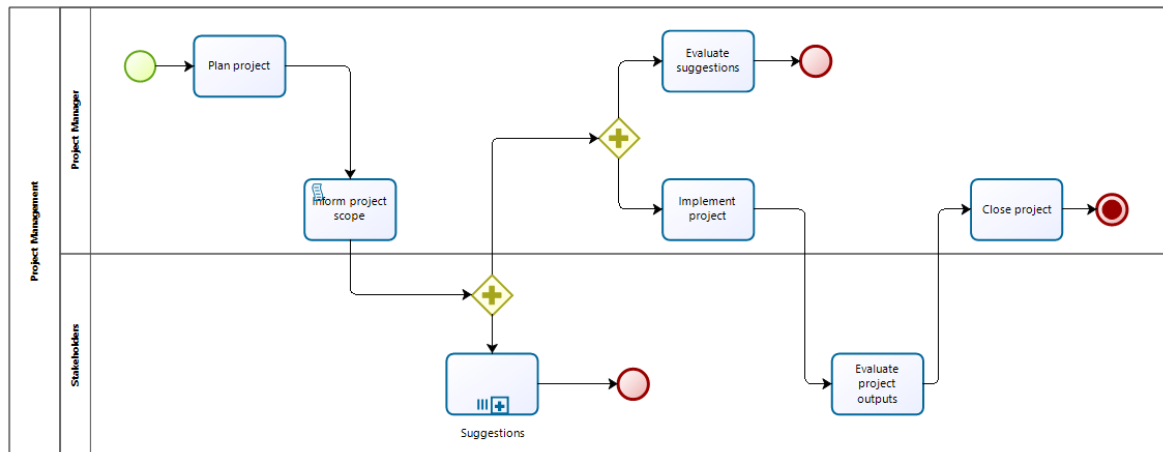



Diagrama 18. Ejemplo del patrón múltiples instancias sin sincronización

Para configurar el comportamiento del sub proceso para la condición que se desea, es necesario definir las siguientes propiedades:



Cantidad de Inicio = 15: Esto significa que el proceso debe ser instanciado 15 veces. Suponga que existen 15 Stakeholders.

Cantidad Final= 15: El sub proceso se considera como terminado cuando se hayan completado todas (en este caso) las 100 instancias.

Condición MI= Ninguna. Esto quiere decir que no hay condición para el número de instancias para el sub proceso.

Orden MI = Paralelo: Las instancias del sub proceso serán ejecutadas en paralelo.

Condición de Flujo = Ninguno. El proceso continúa cada vez que se finaliza una instancia del sub proceso.

## WCP 13 - Múltiples instancias con conocimiento a priori del número de ejecuciones en la etapa de diseño.

### Descripción

El patrón permite instanciar una actividad varias veces en el proceso. El número de instancias es conocido y se ejecutan al mismo tiempo. Es necesario terminar todas las actividades antes de continuar con el flujo del proceso.

### Ejemplo

Una compañía decide evaluar la percepción del ambiente de trabajo entre los empleados. Una encuesta es enviada a 100 empleados. Cuando todos los empleados hayan enviado sus resultados, el analista puede evaluar la información.

### Implementación

Para implementar este patrón es necesario utilizar un sub proceso para crear las 100 instancias de la actividad *Completar encuesta*. En este caso, el sub proceso *Enviar Encuesta* es instanciado 100 veces.

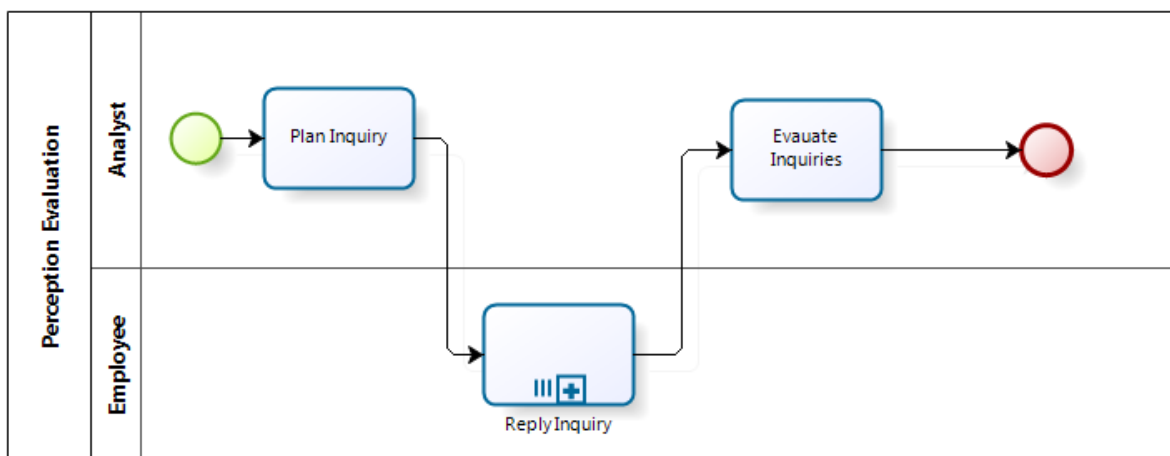
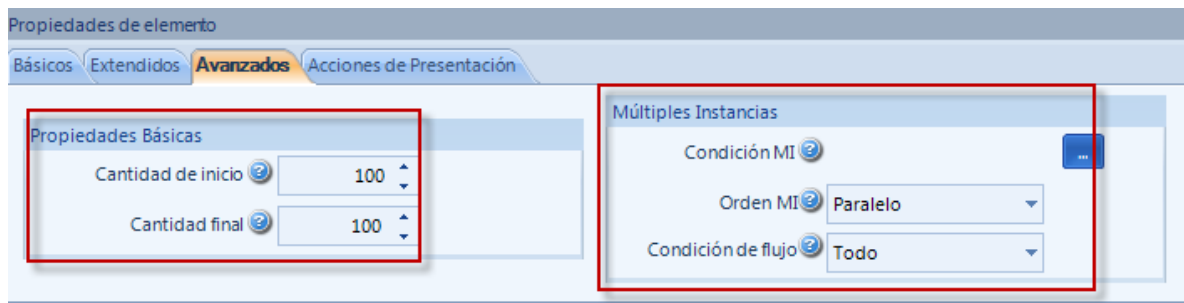


Diagrama 19. Ejemplo del patrón Múltiples instancias con conocimiento a priori del número de ejecuciones en la etapa de diseño.

Para configurar el comportamiento del sub proceso para la condición que se desea, es necesario definir las siguientes propiedades:



Cantidad de Inicio = 100: Esto significa que el proceso debe ser instanciado 100 veces.

Cantidad Final= 100: El sub proceso se considera como terminado cuando se hayan completado todas (en este caso) las 100 instancias.

Condición MI= Ninguna. Esto quiere decir que no hay condición para el número de instancias para el sub proceso.

Orden MI = Paralelo: Las instancias del sub proceso serán ejecutadas en paralelo.

Condición de Flujo = Todo: El proceso continua cuando todas las instancias del sub proceso han terminado.

## WCP 14 – Múltiples instancias con conocimiento en el tiempo de ejecución

### Descripción

El patrón permite instanciar una actividad muchas veces durante el proceso. La cantidad de instancias es conocida hasta el momento de la ejecución del proceso. Las actividades son ejecutadas de forma paralela y es necesario terminar todas las instancias abiertas antes de continuar con la siguiente actividad.

### Ejemplo

Una compañía decide evaluar la percepción del ambiente de trabajo entre los empleados. Un grupo de empleados es escogido de forma aleatoria para contestar una encuesta. Cuando todas las encuestas han sido respondidas, el analista comienza con la evaluación y análisis de ellas.

## Implementación

Para modelar este patrón es necesario utilizar un sub proceso el cual crea las múltiples instancias de una actividad. En este caso, el sub proceso *Responder Encuesta* es instanciado tantas veces como empleados sean escogidos en la actividad anterior.

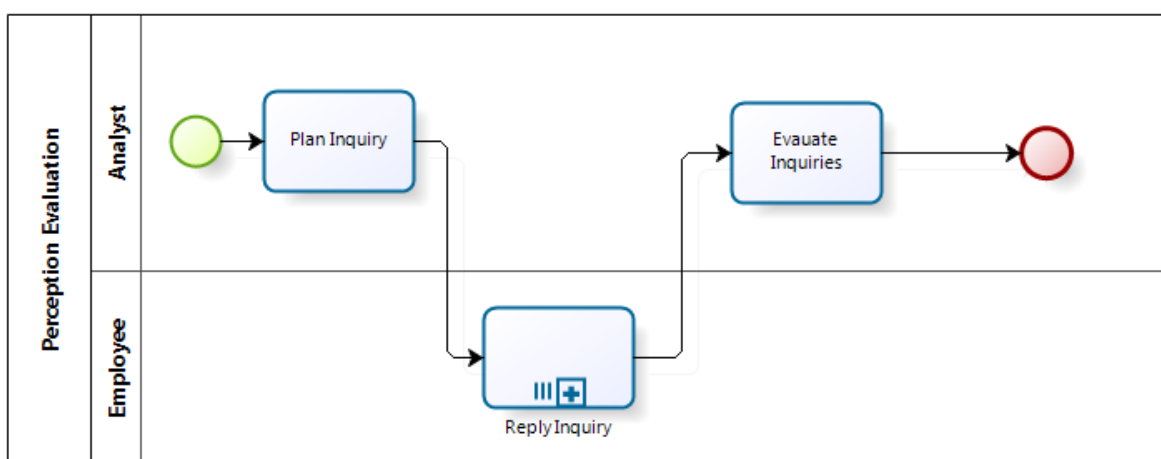
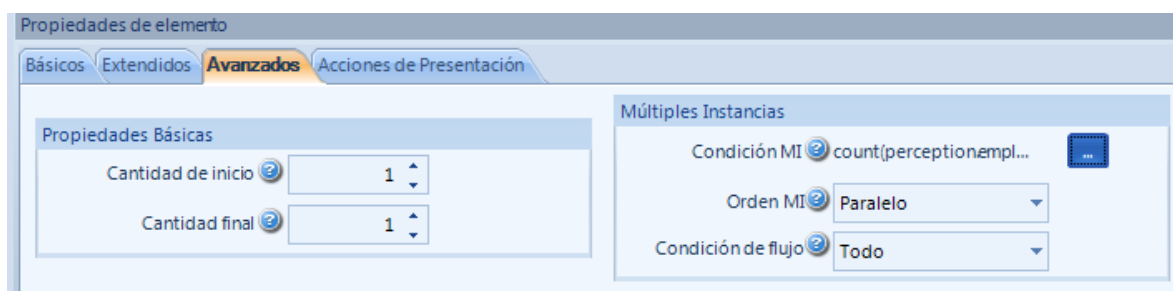


Diagrama 20. Múltiples instancias con conocimiento en el tiempo de ejecución

Para configurar el comportamiento del sub proceso para la condición que se desea, es necesario definir las siguientes propiedades:



Condición MI = `count(Perception.Employees)`: Esto significa que se crearan tantas instancias del sub proceso como empleados han sido seleccionados para contestar la encuesta.

Orden MI = Paralelo: Las instancias del sub proceso se ejecutan en paralelo.

Condición de Flujo = Todo: El proceso terminara cuando todas las instancias hayan terminado.

## WCP 15 – Múltiples Instancias sin conocimiento el tiempo de diseño

### Descripción

Este patrón permite a una actividad ser instanciada muchas veces durante el proceso. La cantidad de instancias no es conocida ni durante la etapa de diseño ni en el tiempo de ejecución. Las actividades son ejecutadas al mismo tiempo, pero mientras se están ejecutando, se pueden crear nuevas instancias de ellas. Es necesario terminar todas las instancias para poder continuar con el flujo del proceso.

### Ejemplo

El equipo del proyecto realiza una reunión de comité para empezar el proyecto. Durante la reunión se planeas varias actividades que se deben realizar posteriormente. Sin embargo, una situación inesperada puede surgir y nuevas actividades deben ser realizadas.

El equipo del proyecto puede crear nuevas actividades, de ser requerido, antes de que todas las instancias hayan terminado.

### Implementación

El diagrama 21 muestra cómo se debe modelar el patrón. En la actividad *Reunión de comité* se crean un número inicial de las instancias (actividades), luego, un evento es utilizado para permitir la creación de nuevas tareas. Una vez las tareas han sido completadas, el evento es deshabilitado y no se permite crear nuevas actividades.

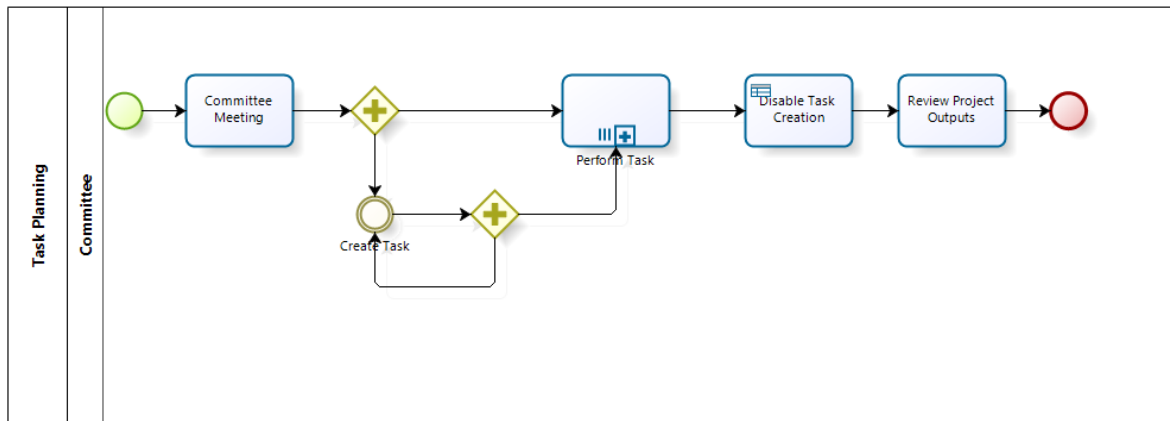
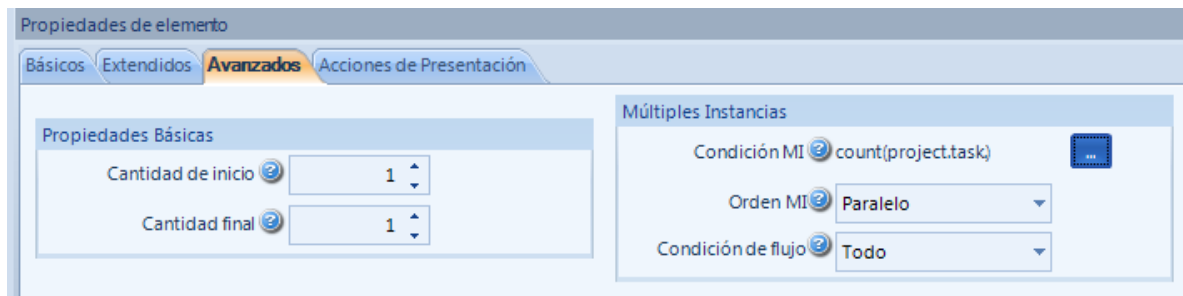


Diagrama 21. Ejemplo de Múltiples Instancias sin conocimiento el tiempo de diseño

Para configurar el comportamiento del sub proceso para la condición que se desea, es necesario definir las siguientes propiedades:



Condición MI = count(Project.Task): Esto significa que se crearan tantas instancias del sub proceso como tareas han sido creadas.

Orden MI = Paralelo: Las instancias del sub proceso se ejecutan en paralelo.

Condición de Flujo = Todo: El proceso terminara cuando todas las instancias hayan terminado.

## WCP 34 – Unión Parcial Estática de múltiples Instancias

Descripción:

Este patrón permite instanciar una actividad varias veces. El número de instancias es conocido durante la fase de diseño o en el tiempo de ejecución. Las actividades son ejecutadas de forma paralela y solo es necesario que terminen N de las M ( $N < M$ ) instancias creadas para continuar con

Ejemplo

Una compañía decide evaluar la percepción del ambiente de trabajo entre los empleados. Una encuesta es enviada a 100 empleados, el analista espera a que 60 de los 100 empleados envíen sus resultados para empezar con el análisis.

Implementación

Para implementar este patrón es necesario utilizar un sub proceso para crear las 100 instancias de la actividad *Completar Encuesta*. La compuerta compleja se utiliza para controlar que al menos 60 encuestas han sido respondidas antes de continuar con el proceso.

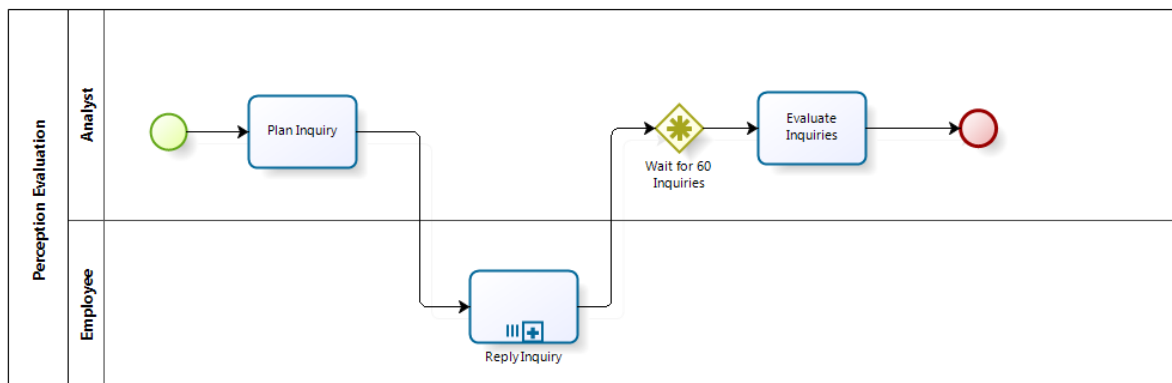


Diagrama 22. Ejemplo Unión Parcial estática de múltiples Instancias

Para configurar el comportamiento del sub proceso para la condición que se desea, es necesario definir las siguientes propiedades:





Cantidad de Inicio = 100: Esto significa que el proceso debe ser instanciado 100 veces.

Cantidad Final= 100: El sub proceso se considera como terminado cuando se hayan completado todas (en este caso) las 100 instancias.

Condición MI= Ninguna. Esto quiere decir que no hay condición para el número de instancias para el sub proceso.

Orden MI = Paralelo: Las instancias del sub proceso serán ejecutadas en paralelo.

Condición de Flujo =Ninguno: El proceso continúa cada vez que se finaliza una instancia del sub proceso.

## WCP 35 – Cancelación parcial de unión de múltiples instancias

Descripción:

Este patrón permite instanciar una actividad múltiples veces. El número de instancias es conocido durante la fase de diseño o en el tiempo de ejecución. Las actividades son ejecutadas de forma paralela y solo es necesario que terminen N de las M ( $N < M$ ) instancias creadas para continuar con el resto del proceso. Una vez las N instancias han sido completadas, las instancias faltantes ( $M - N$ ) son canceladas.

Ejemplo:

Una compañía decide evaluar la percepción del ambiente de trabajo entre los empleados. Una encuesta es enviada a 100 empleados, el analista espera a que 60 de los 100 empleados envíen sus resultados para empezar con el análisis. Las 40

encuestas restantes son canceladas, es decir no será posible que los 40 empleados faltantes envíen sus resultados.

Implementación:

Para implementar este patrón es necesario utilizar un sub proceso para crear las 100 instancias de la actividad completar encuesta.

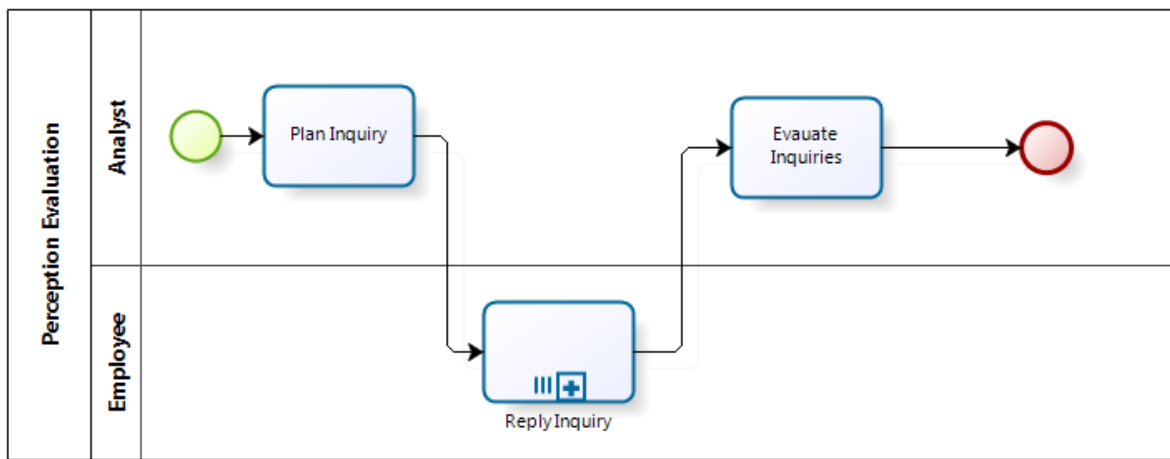


Diagrama 23. Ejemplo Cancelación parcial de unión de múltiples instancias

Para configurar el comportamiento del sub proceso para la condición que se desea, es necesario definir las siguientes propiedades:



Cantidad de Inicio = 100: Esto significa que el proceso debe ser instanciado 100 veces.

Cantidad Final= 100: El sub proceso se considera como terminado cuando se hayan completado 60 de las 100 instancias.

Condición MI= count(Percepcion.Employee). Esto quiere decir que se crean tantos casos como empleados sean incluidos en la actividad anterior.

Orden MI = Paralelo: Las instancias del sub proceso serán ejecutadas en paralelo.

Condición de Flujo = Todo: El proceso continua cuando todas las instancias del sub proceso han terminado.

## WCP 36 – Unión parcial dinámica de múltiples instancias

### Descripción

Este patrón permite instanciar una actividad múltiples veces. El número de instancias es conocido durante la fase de diseño o en el tiempo de ejecución. Las actividades son ejecutadas de forma paralela, pero es posible crear nuevas instancias de las tareas.

La condición de terminación es validada cada vez que una instancia termina. Una vez la condición ha sido evaluada como cierta, la siguiente tarea se activa aunque existan instancias sin terminar, la terminación de las demás tareas es irrelevante y luego de alcanzar la condición no se pueden crear nuevas instancias de las tareas [1].

### Ejemplo

Una compañía decide evaluar la percepción del ambiente de trabajo entre los empleados. La encuesta es enviada a algunas personas. Es posible enviar nuevas encuestas a otros empleados antes de que las encuestas ya enviadas hayan sido respondidas.

El analista espera hasta que 60 encuestas hayan sido respondidas para poder realizar la evaluación. Las demás encuestas pueden ser enviadas, pero no se tendrán en cuenta.

### Implementación

Para implementar el patrón es necesario utilizar un sub proceso para crear todas las instancias necesarias. En este caso, el sub proceso Responder Encuesta es instanciado tantas veces como sea necesario.

Una compuerta compleja es utilizada para controlar que cuando se completen 60 encuestas el proceso continúe con la actividad *Evaluar Encuestas*.

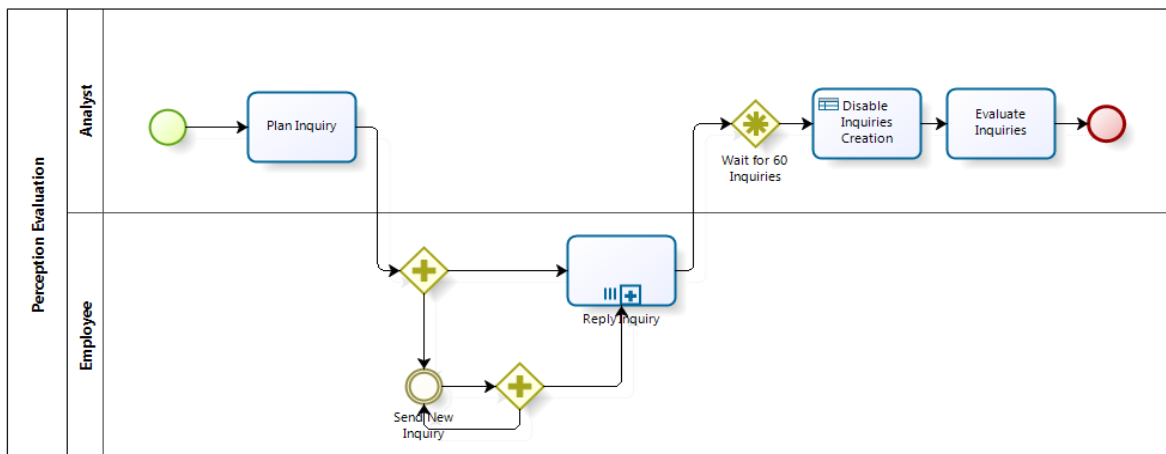


Diagrama 24. Ejemplo Unión parcial dinámica de múltiples instancias

Para configurar el comportamiento del sub proceso para la condición que se desea, es necesario definir las siguientes propiedades:



Condición MI = count(Perception.Employee): Esto significa que se crearan tantas instancias del sub proceso como tareas han sido creadas.

Orden MI = Paralelo: Las instancias son ejecutadas en forma paralela.

Condición de Flujo= Ninguno: El proceso continúa cada vez que una instancia es terminada.

## Patrones basados en Eventos.

Los patrones de este grupo aplican los conceptos de eventos

### WCP 16 – Decisión Implícita

#### Descripción

El patrón describe un punto en el proceso donde una rama es escogida de varias disponibles. La decisión es tomada de acuerdo a los datos del proceso. Cuando una rama es escogida, las demás se deben deshabilitar.

#### Ejemplo

Cuando un cliente realiza una solicitud de crédito es necesario solicitarle diferentes documentos; Si el cliente no trae los documentos antes de 5 días se le debe contactar y preguntar si continúa o no en el proceso. Si el cliente no continúa en el proceso no es necesario esperar por los documentos y el proceso debe terminar. Si el cliente trae los documentos, no es necesario contactarlo.

#### Implementación

Para implementar el patrón es necesario utilizar una compuerta exclusiva basada en eventos. Esta compuerta representa un punto en el proceso donde solo una de las ramas disponibles debe ejecutar. Las ramas restantes deben ser deshabilitadas.

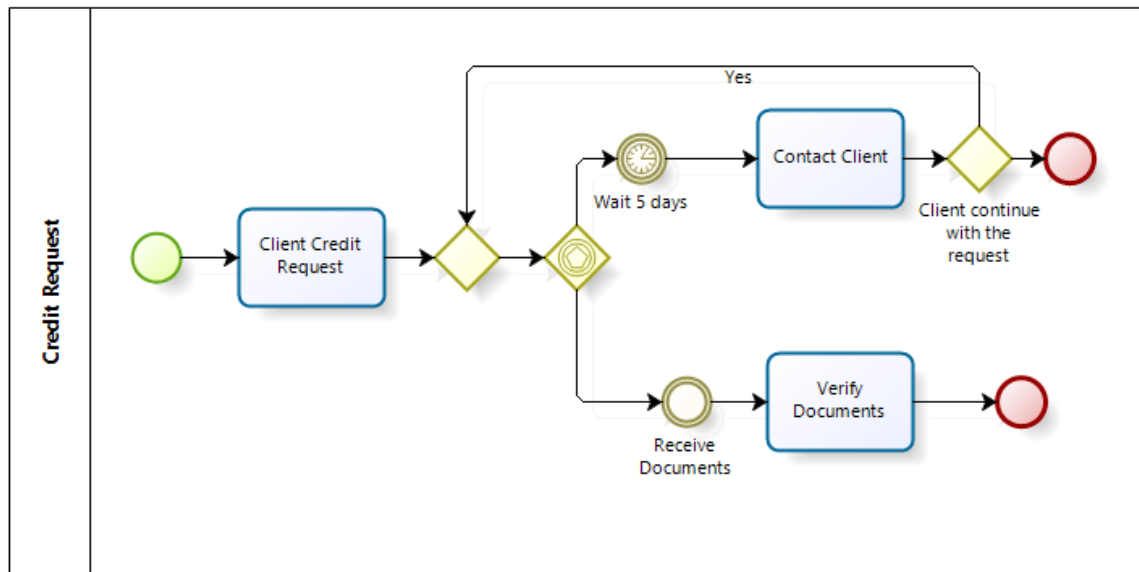


Diagrama 25. Ejemplo Decisión Implícita

## WCP 17 – Ejecución Paralela Intercalada

### Descripción

Un conjunto de actividades tienen un orden de ejecución definido. Cada actividad puede ser ejecutada una vez y se puede completar en cualquier momento de acuerdo al orden definido. Sin embargo, como requisito adicional, dos actividades no pueden ser ejecutadas al mismo tiempo [1].

### Ejemplo

El Jefe de Recursos Humanos estable que durante los procesos de selección los aspirantes deben presentar tres pruebas de aptitud: una prueba psicológica, un test de inteligencia y un test de conocimientos específicos.

El orden en que se realizan las pruebas es importante. El test de inteligencia debe ser realizado antes de la prueba psicológica. La prueba de conocimientos debe ser realizada en cualquier momento. No es posible realizar dos pruebas al mismo tiempo.

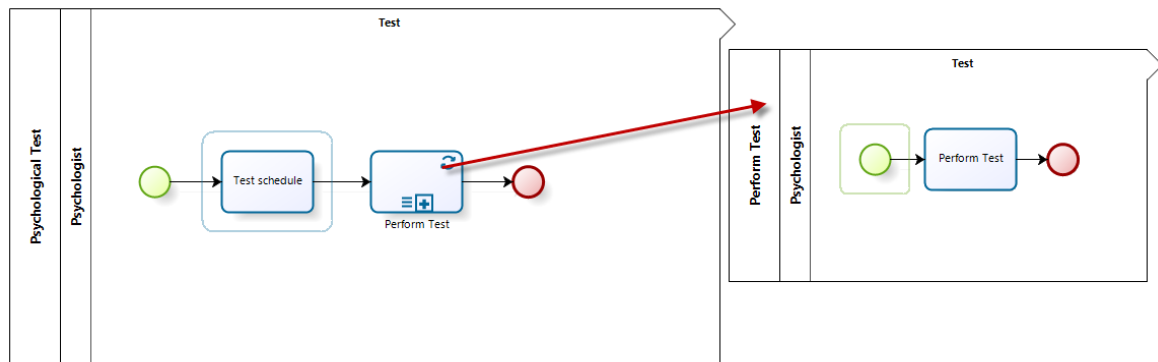
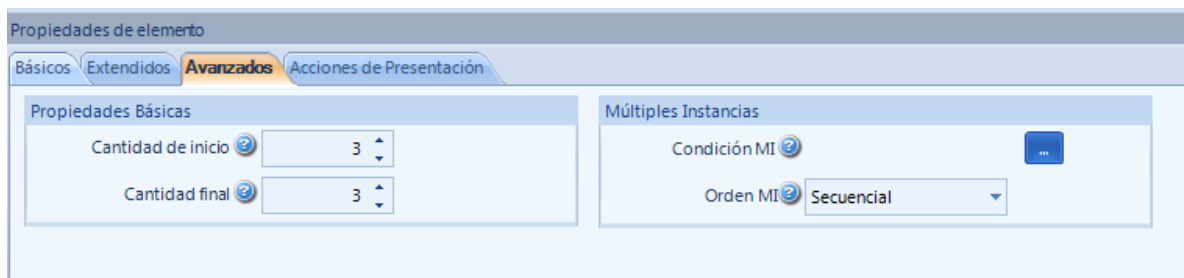


Diagrama 26. Ejemplo Ejecución Paralela Intercalada

## Implementación

Para modelar el patrón se utiliza un sub proceso múltiple, el sub proceso solo incluye una actividad. Para definir el orden en el cual se realizan las pruebas es necesario establecer como propiedades del sub proceso los siguientes datos.



La secuencia en que las pruebas se realizan es definida en la primera actividad del proceso.

## WCP 18 - Milestone

### Descripción

El patrón define que una actividad es habilitada solamente cuando el proceso alcance un estado específico (Hito). Si el proceso ha avanzado más allá de ese estado, la tarea no puede volver a ser habilitada.

### Ejemplo

En una agencia de viajes es posible cambiar las fechas de las reservas de vuelos, carros y hoteles mientras las facturas no hayan sido impresas.

## Implementación

El patrón utiliza una compuerta exclusiva basada en eventos para controlar la disponibilidad de la actividad *Cambiar reservas*. Una vez la actividad *Imprimir Factura* se ha terminado, no es posible cambiar las reservas.

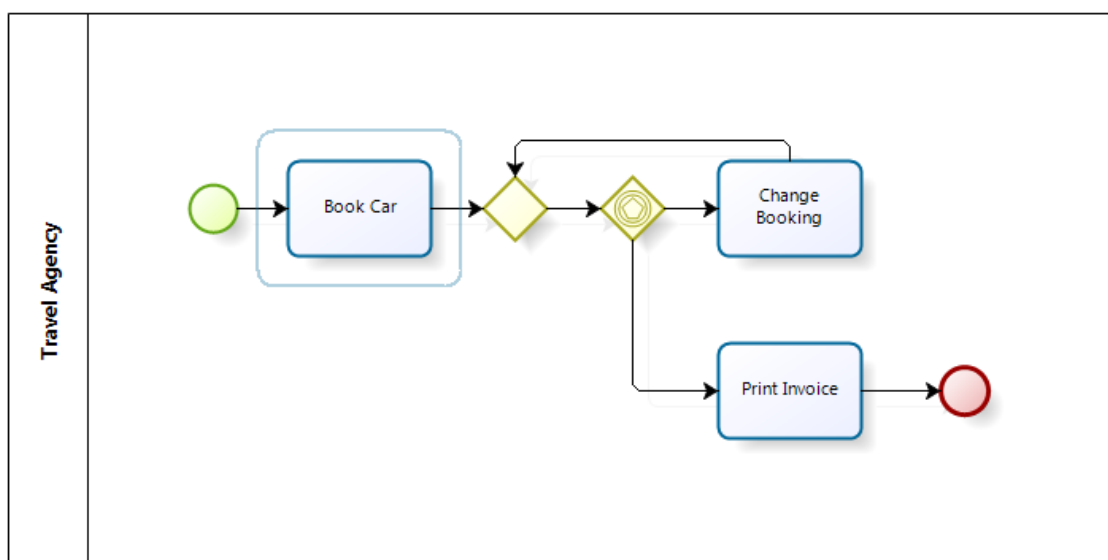


Diagrama 27. Ejemplo del patrón Hito

## WCP 39 – Sección Crítica.

### Descripción

El patrón de Sección Crítica describe la identificación de dos o más sub procesos o actividades como secciones críticas. Cuando una de estas secciones críticas es activada, es decir la actividad dentro de la sección es habilitada, las otras secciones críticas no pueden ser activadas.

El proceso espera hasta que la sección crítica sea completada para permitir la ejecución de otra sección crítica.

### Ejemplo

Dos administradores tienen acceso al servidor para realizar cambios en la configuración. Mientras que uno de ellos se encuentre trabajando en el servidor no es posible que el otro administrador realice cambios. Él o ella deberá esperar hasta que el primer administrador termine su trabajo.



## Implementación

El patrón es modelado con la ayuda de una Compuerta Basada en Eventos para controlar que los dos administradores no trabajen de forma simultánea en la ejecución de tareas críticas. En este caso la tarea crítica es la actualización del servidor y solo puede ser realizada por un administrador.

Una vez que el servidor ha sido actualizado, una compuerta exclusiva evalúa si se necesitan más cambios para volver a habilitar las tareas críticas o si el proceso puede terminar.

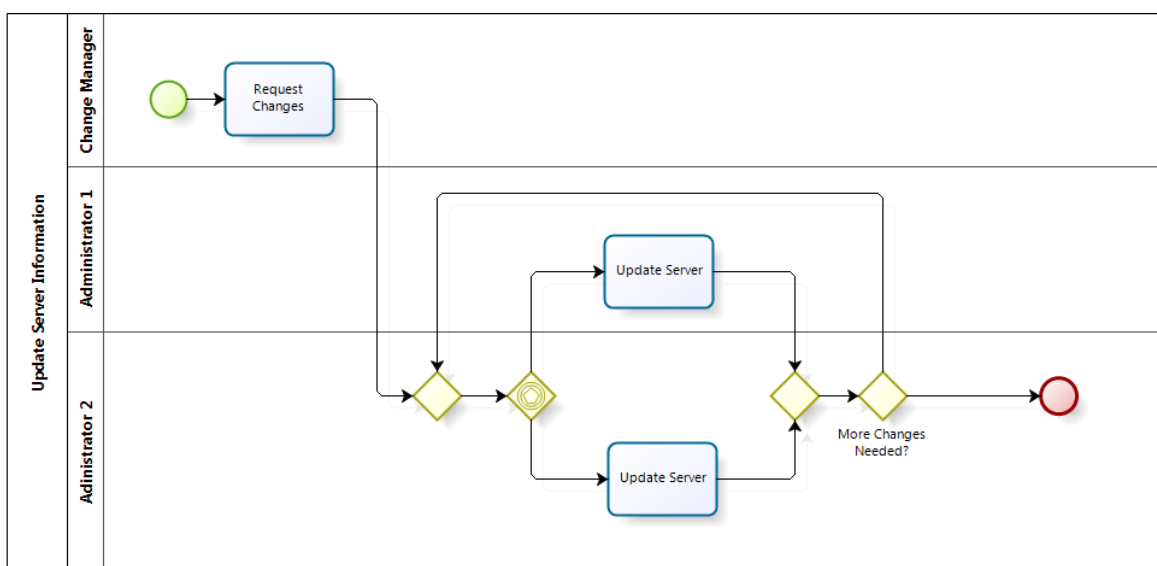


Diagrama 28. Ejemplo de Sección Crítica.

## WCP 40 Ejecución Intercalada

### Descripción

El patrón describe que cada actividad de un conjunto de actividades debe ser ejecutada una vez. Las tareas pueden ser realizadas en cualquier orden, sin embargo, no es posible realizar dos tareas al mismo tiempo.

### Ejemplo

Un estudiante debe presentar tres pruebas para ser admitido a la Universidad: una prueba psicológica, una prueba de admisión y una de conocimientos generales.

El orden en que se ejecutan las pruebas no es importante, sin embargo no es posible realizar varias pruebas al mismo tiempo.

## Implementación

El patrón utiliza una compuerta Inclusiva para controlar que una actividad debe ser ejecutada en cada caso. La primera actividad indica cuál es la prueba a realizar, la compuerta inclusiva habilita la actividad seleccionada y la compuerta exclusiva controla que todas las actividades han sido realizadas para continuar con el flujo normal del proceso.

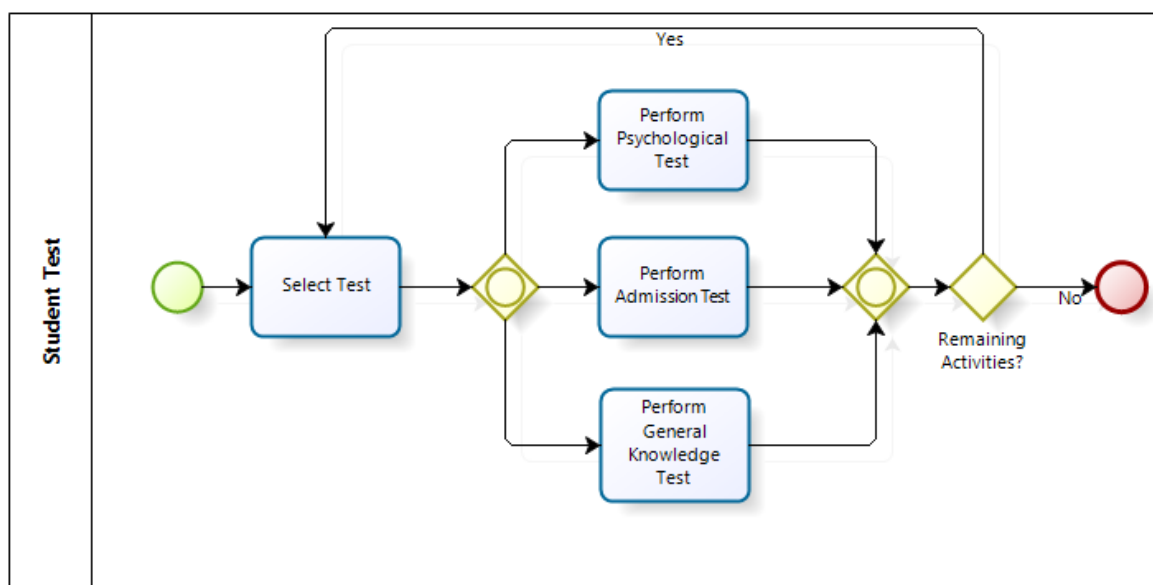


Diagrama 29. Ejemplo ejecución Intercalada

## Patrones de Cancelación y de Terminaciones Forzadas

Los patrones de este grupo aplican los conceptos de cancelación y de suprimir actividades.

### WCP 19 – Cancelación de tareas

#### Descripción

El patrón de Cancelación de Tareas da la posibilidad de cancelar o suprimir una tarea que está siendo ejecutada [2].

#### Ejemplo

Durante una solicitud de viajes es necesario reservar varias cosas, por ejemplo, hotel, carro, vuelo y entregar un adelanto al empleado. El empleado debe poder cancelar en viaje antes de terminar las reservas.

### Implementación

El patrón es modelado con una compuerta basada en eventos y un evento intermedio. La compuerta habilita la actividad Realizar las Reservas y el evento que cancela el proceso. Solo una de estas dos cosas puede ser pasar.

Si se cancela el viaje no se debe realizar reservas, y solo es posible cancelar el viaje antes de entregar el adelanto al empleado.

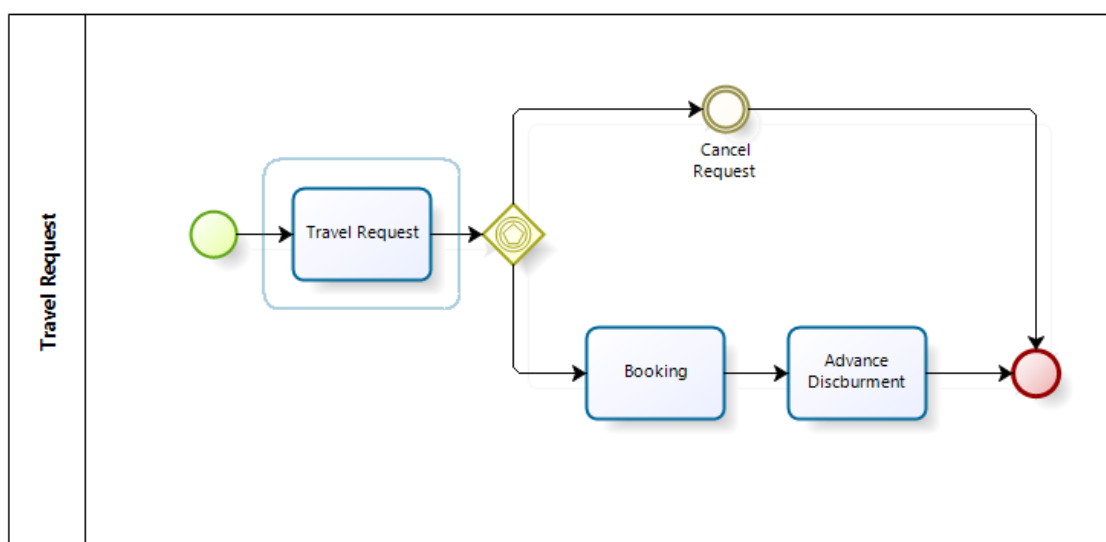


Diagrama 30. Cancel Ejemplo de Cancelación de Tareas

## WCP 20 – Cancelación del Caso

### Descripción

El patrón describe la eliminación de una instancia completa del proceso. Esto incluye tareas que están siendo ejecutadas y las tareas que todavía no lo han sido [2].

### Ejemplo

Cuando un incidente ocurre, este debe ser reportado a la Mesa de Ayuda de la compañía. Luego de reportar el incidente, la Mesa de Ayuda debe resolverlo y tener la posibilidad de cancelar el caso en cualquier momento.

## Implementación

Para implementar el patrón se utiliza una compuerta paralela y un evento de fin terminal. Una vez que el proceso alcance el fin terminal, las actividades pendientes son canceladas y el caso termina.

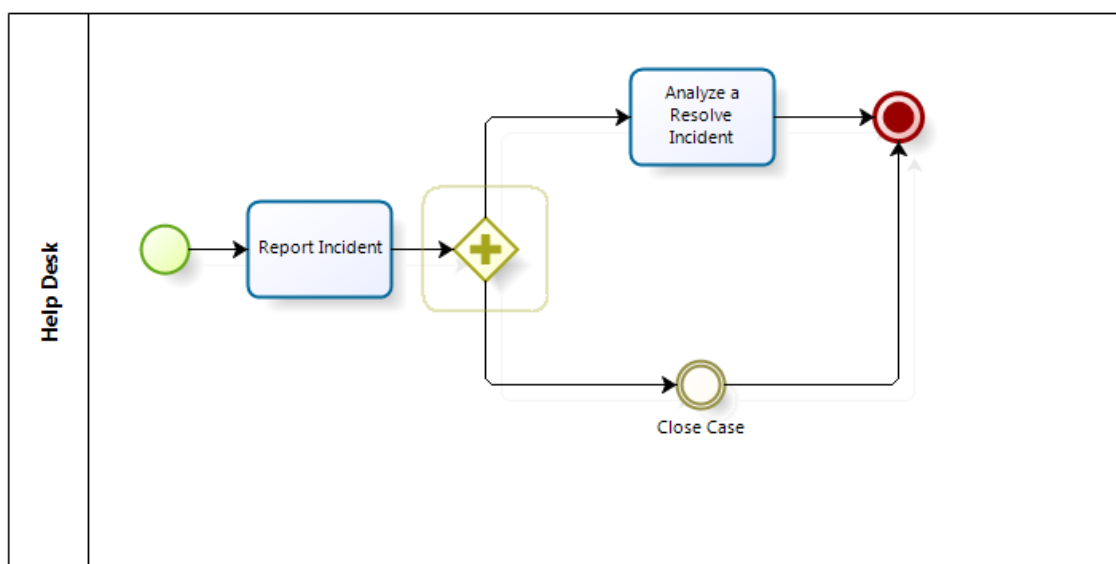


Diagrama 31. Ejemplo de Cancelación del Caso

## WCP 25 – Cancelación de una Región

### Descripción

El patrón describe la posibilidad de cancelar un grupo de actividades en un proceso. Si algunas de las actividades se están ejecutando, estas son canceladas.

### Ejemplo

Cuando se realiza una solicitud de viajes es necesario gestionar distintas reservaciones. Si no es posible realizar alguna de las ellas, el viaje debe ser cancelado.

### Implementación

Bizagi soporta este patrón a través de un sub proceso transaccional. El sub proceso Transaccional permite la interrupción de las actividades y continuar por un flujo especial a través de un evento intermedio de error.

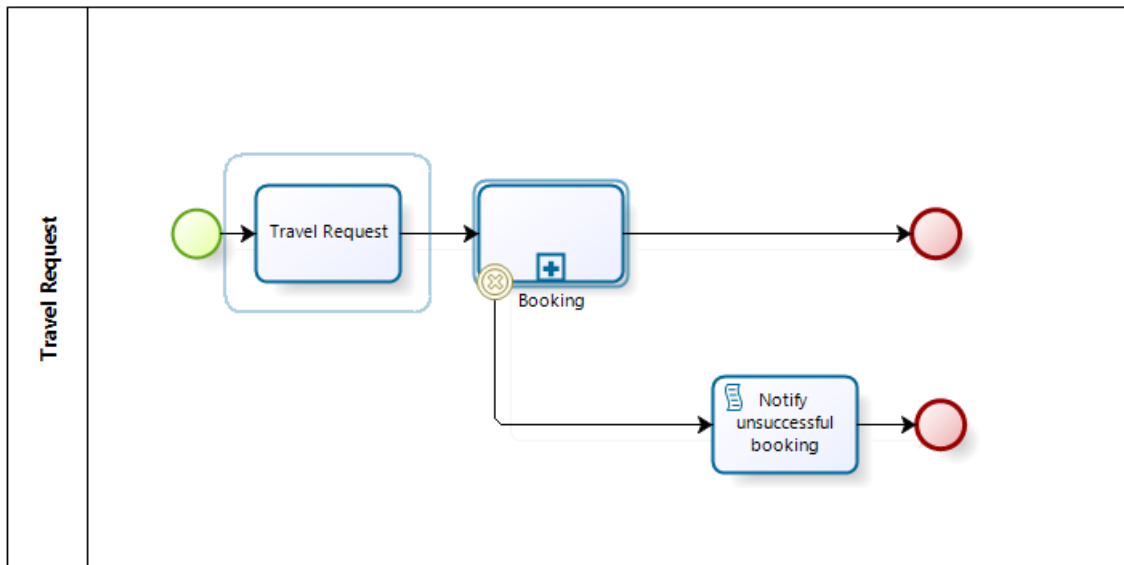


Diagrama 32. Ejemplo Cancelación de una Región

## WCP 26 – Cancelación de Múltiples Instancias de una Actividad

### Descripción

El patrón permite cancelar todo un grupo de múltiples instancias de una actividad removiendo aquellas que no han terminado. Las instancias que ya fueron completadas no se afectan [2].

### Ejemplo

Un Gerente del Proyecto define y asigna la creación de entregables para la ejecución de un proyecto. Cada uno de los entregables debe realizarse de forma secuencial. El Gerente del Proyecto debe poder cancelar la ejecución de los entregables en cualquier momento.

### Implementación

El patrón puede ser modelado utilizando eventos de interrupción. Este tipo de eventos se adjuntan al borde la actividad e interrumpen su ejecución cuando son lanzados.

En este caso, cuando se cumple una condición especial (relacionada a la cancelación de los entregables que no han sido terminados) las instancias pendientes del sub proceso *Entregables* serán canceladas y el proceso termina por el flujo normal.

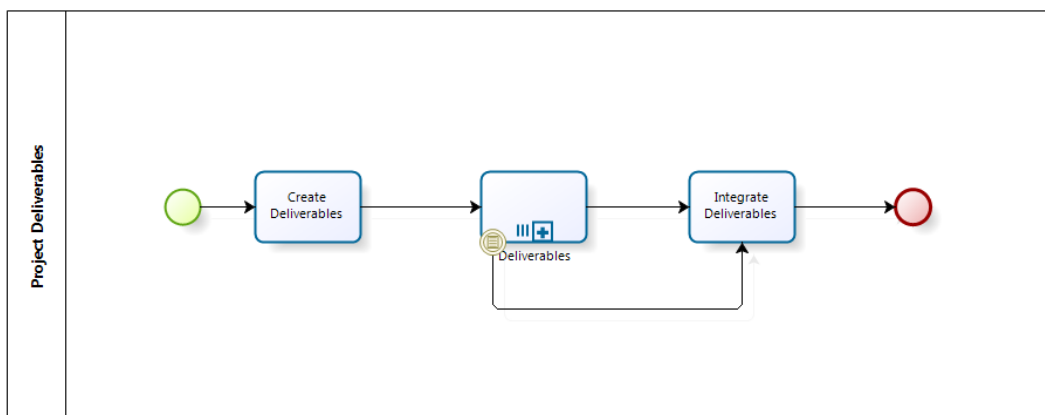


Diagrama 33. Ejemplo Cancelación de Múltiples Instancias de una Actividad

## WCP 27 – Completar Múltiples Instancias de una Actividad

### Descripción

El patrón describe la habilidad de completar todo un grupo de múltiples instancias de una tarea removiendo aquellas que no han terminado.

### Implementación

*Bizagi no soporta este patrón*

## Patrones de Iteraciones

### WCP 26 – Ciclos Arbitrarios

#### Descripción

El patrón modela un punto en el flujo del proceso donde una o más actividades son realizadas varias veces.

#### Ejemplo

Los entregables de un proyecto necesitan la aprobación de varias personas. El número de aprobaciones es definido por el Gerente del Proyecto.

## Implementación

El patrón utiliza una compuerta exclusiva que controla el número de veces que se ha aprobado el entregable, cuando se cumpla la condición (número de aprobaciones requeridas) el proceso continúa por el flujo normal.

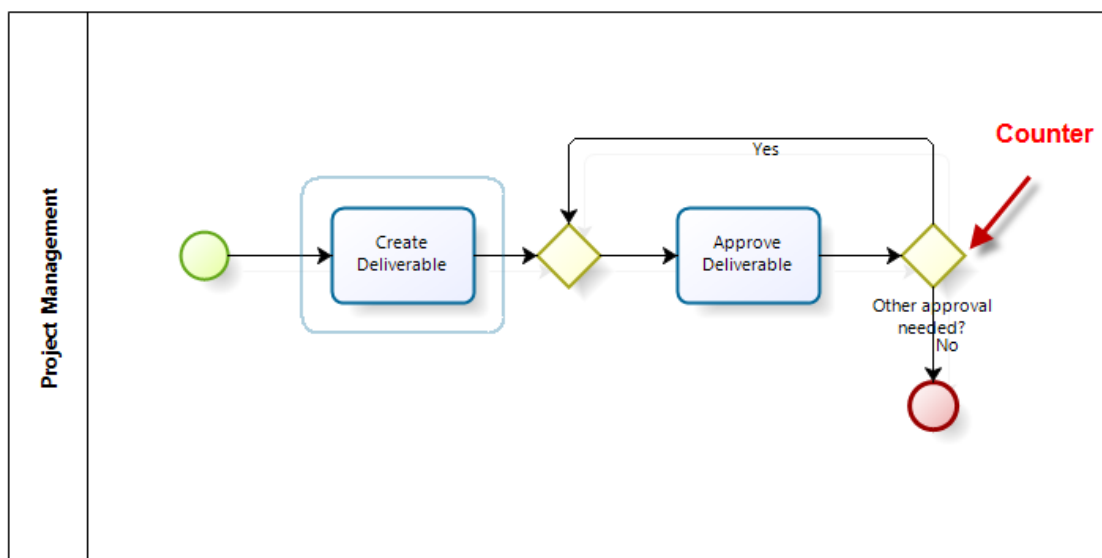


Diagrama 35. Ejemplo de Ciclos Arbitrarios.

## WCP 10 – Bucle Estructurado.

### Descripción

El patrón describe la posibilidad de ejecutar una actividad o un sub proceso varias veces. El ciclo o bucle tiene una validación de condición asociado a él. La condición es verificada al inicio o al final del ciclo para determinar si se continúa. La estructura del ciclo tiene una sola entrada y un solo punto de salida [2].

### Ejemplo

En el proceso de *Cuentas por Pagar* se reciben facturas. Si la factura no es aprobada se devuelve al proveedor y se espera su envío de nuevo. Este proceso es realizado hasta que la factura sea aprobada.

## Implementación

Para implementar el patrón es necesario utilizar una compuerta exclusiva. La compuerta valida una condición, en este caso, si la condición se cumple, el proceso continúa por el flujo normal, sino, se devuelve a la actividad *Recibir y Aprobar Factura*.

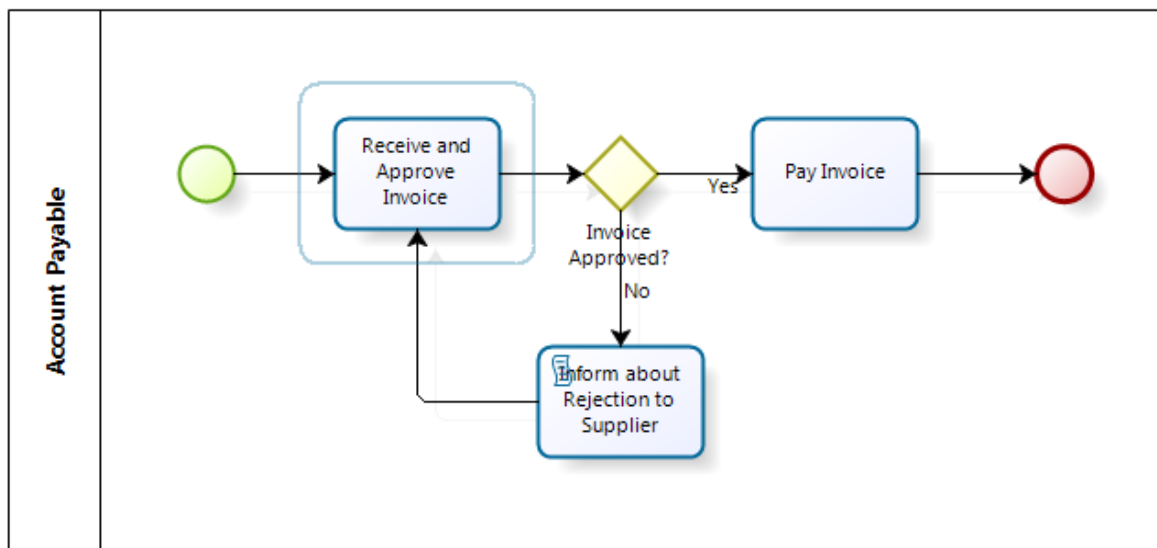


Diagrama 36. Ejemplo bucle Estructurado

## WCP 22 - Recursividad

### Descripción

El patrón de recursividad describe la posibilidad de una tarea de invocarse a sí misma dentro del contexto donde está definida.

### Implementación

*Bizagi no soporta este patrón*



## Patrones de Terminación

Los patrones de terminación son utilizados para determinar cuándo las instancias de un proceso deben terminar de acuerdo a las condiciones del negocio.

### WCP 11 Terminación Implícita

#### Descripción

El patrón es utilizado para determinar cuándo la instancia de un proceso se considera completa. Un determinado proceso (o sub proceso) debe terminar cuando no hay tareas pendientes o cuando no hay posibilidad de crear nuevos trabajos en el futuro [1].

#### Ejemplo

Considere la Logística Inversa en una empresa manufacturera. Cuando se recibe una solicitud de cambio se deben realizar dos actividades para que el proceso pueda terminar. La primera actividad es generar una orden de cambio y la segunda es la de documentar las causas de la devolución.

#### Implementación

El patrón de terminación implícita se utiliza para el siguiente ejemplo.

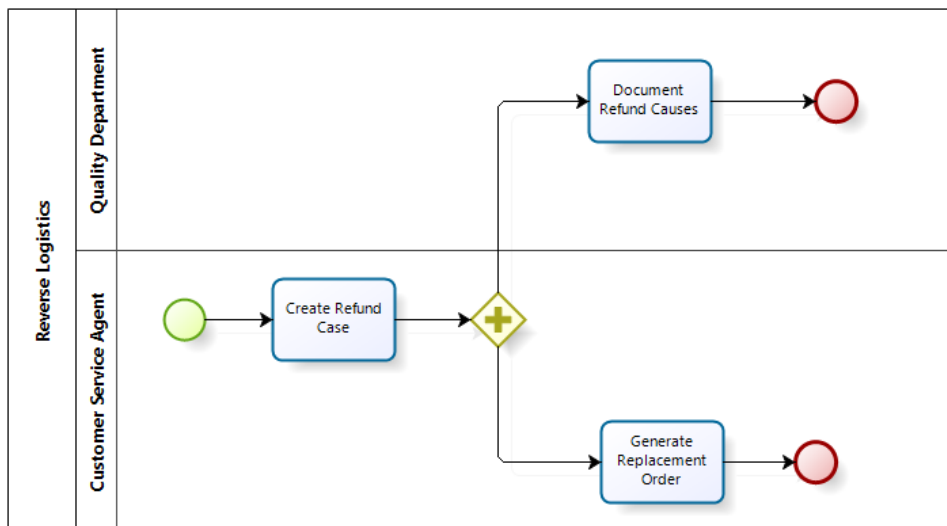


Diagrama 37. Ejemplo de Terminación Implícita

Note que el patrón utiliza evento al final de los caminos que deben ser completados. El proceso se considera terminado cuando todos los eventos de fin son alcanzados.

## WCP 43 - Terminación Explícita

### Descripción

La instancia de un proceso o de un sub proceso debe terminar si alcanza cierto estado. Generalmente se especifica utilizando un nodo de final específico. Cuando se alcanza el nodo, las tareas o trabajo faltante son cancelados y la instancia del proceso es completada de forma exitosa [1].

### Ejemplo

Suponga un proceso de *Solicitud de Crédito*. El proceso incluye un sub proceso llamado Solicitar Documentos donde un analista de crédito determina qué papeles deben ser enviados por el cliente para otorgar el crédito, luego, un correo electrónico es enviado al cliente solicitando los documentos.

Si luego de 3 días el cliente no ha entregado los documentos, la solicitud se cancela y el sub proceso debe terminar.

### Implementación

El patrón de terminación explícita es presentado a continuación con la ayuda de evento de fin terminal.

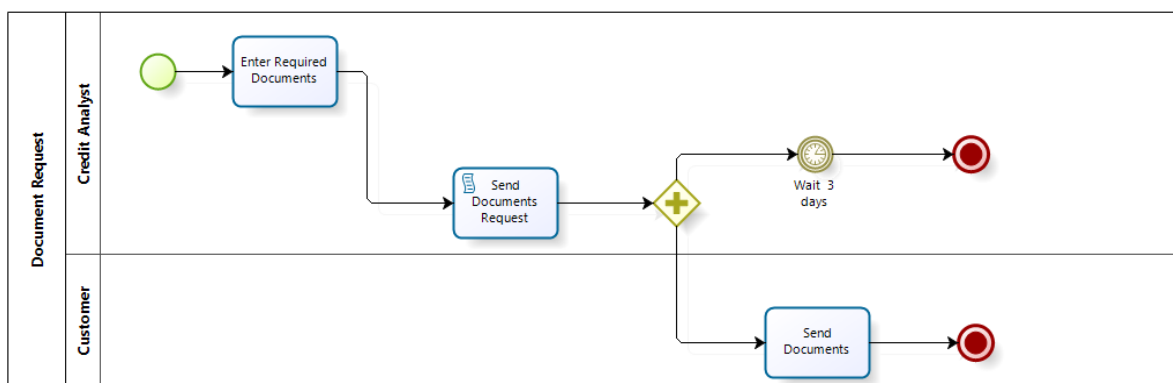


Diagrama 38. Ejemplo de Terminación Explícita

Note que el patrón utiliza Eventos de fin Terminal al final de cada una de las ramas habilitadas. El proceso se considera como terminado cuando alguno de los eventos es alcanzado.

## Patrones de Activación o de disparo

Los patrones de activación o de disparo modelan procesos que interactúan con otros a través del flujo de mensajes.

### WCP 23 – Activación Transitoria

#### Descripción

El patrón permite que una actividad sea activada por otra parte del proceso o por un proceso externo. Esta activación es transitoria ya que si la señal o mensaje para activar no es recibida por una actividad de inmediato, esta se pierde [1].

#### Ejemplo

En el proceso de solicitud de compras, el departamento de compras recibe requerimientos. Se debe seleccionar el mejor proveedor y generar la orden de compra.

El proceso no puede continuar con la actividad de *Pago de factura* hasta que los productos hayan sido entregados e inspeccionados por la empresa.

#### Implementación

Para implementar el patrón se utilizan mensajes de eventos o señales.

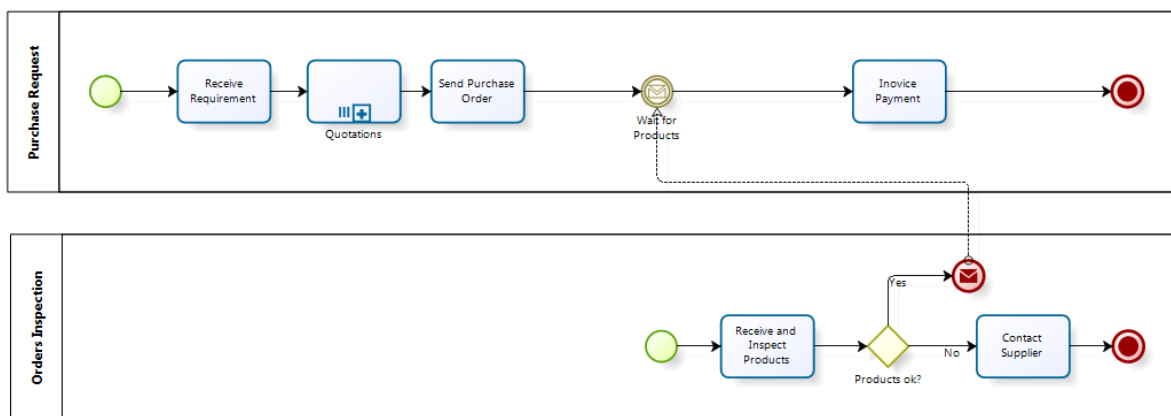


Diagrama 39. Ejemplo de Activación Transitoria.

En este caso el proceso espera el mensaje para poder continuar. Para más información sobre configuración de eventos de mensaje visite:

[http://help.bizagi.com/bpmsuite/es/index.html?bpmn\\_colaborativo.htm](http://help.bizagi.com/bpmsuite/es/index.html?bpmn_colaborativo.htm)

## WCP 24 – Activación Persistente

### Descripción

El patrón permite que una actividad sea activada por otra parte del proceso o por un proceso externo. La activación es persistente ya que es guardada hasta que el flujo que debe recibirla llegue al punto de recepción de la señal [1].

### Ejemplo

Una empresa que produce tapicería para los asientos de carros deportivos. La tapicería es producida de acuerdo a las especificaciones y preferencias de los clientes, cuando estas son definidas se envía una señal a la línea de montaje para continuar con el proceso de producción de los automóviles.

### Implementación

El patrón utiliza modelos de colaboración y retiene el mensaje enviado por el primer mensaje hasta que el Segundo proceso pueda recibirlo. En este caso, la empresa termina la tapicería de los asientos, sin importar donde se encuentre el proceso de producción del carro. El mensaje de terminación de la tapicería no se pierde, esta persiste hasta que el proceso de producción pueda escucharla.

Incluso cuando el coche está justo en el comienzo de la línea de montaje, el mensaje se envía una vez pero no se pierde si la línea de ensamble no está lista para instalar los asientos.

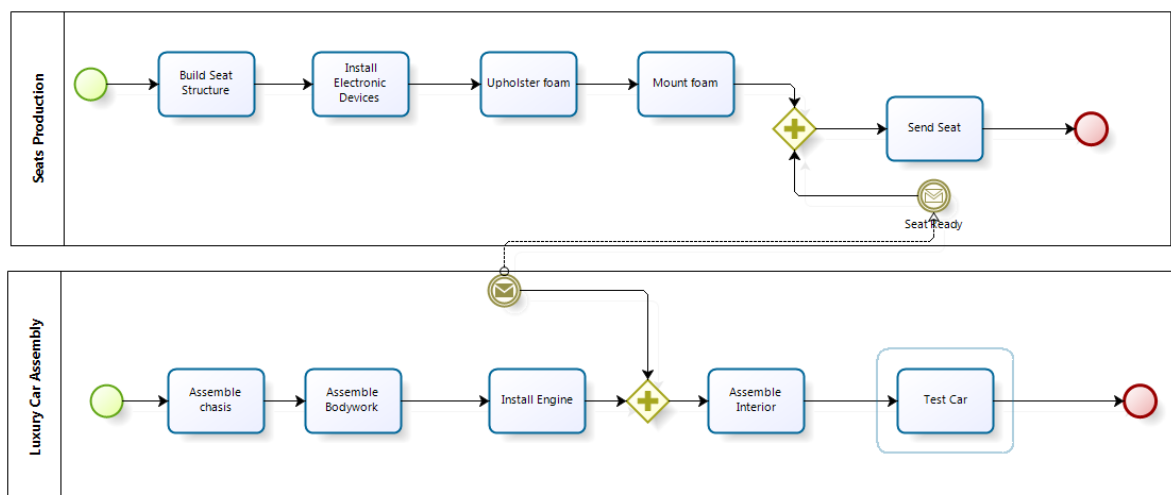


Diagrama 39. Ejemplo de Activación Persistente

## REFERENCIAS

- [1] Nick Russell, Arthur H.M. Ter Hofstede, "Workflow Control-Flow Patterns, A Revised View".BPM Group, Queensland University Of Technology, Australia. 2006.
- [2] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski and A.P. Barros, "Workflow Patterns", Department of Technology Management, Eindhoven University of Technology, Australia, 2002.
- [3] Marcus Goetz, "Modeling Workflow Patterns through a Control-flow perspective using BPMN and the BPM Modeler Bizagi", Institute of Applied Informatics and Formal DescripciónMethods, University Karlsruhe,