

How to

# Q-flow

# Patrones básicos de

# Workflow



*Versión: 2.0*

*Fecha de publicación 28-03-2011*

*Aplica a: Q-flow 3.0 y Q-flow 3.1*

## Índice

Introducción.....	3
Patrones de control.....	4
Patrón: Secuencia .....	4
Patrón: Separación en paralelo (también llamado AND/split) .....	4
Patrón: Sincronización .....	5
Patrón: Opción exclusiva (XOR-Split) .....	6
Patrón: Fusión simple .....	7
Patrones avanzados de paralelismo y sincronización.....	9
Patrón: Múltiple Opción (Or-Split).....	9
Patrón: Fusión Sincronizada.....	10
Patrón: Fusión Múltiple .....	12
Patrón: Discriminador .....	13
Patrón: Unión, N-de-los-M.....	16
Patrones Estructurales.....	19
Patrón: Ciclos Arbitrarios .....	19
Patrón: Terminación Implícita .....	20
Patrones de tareas con múltiples instancias .....	22
Patrón: Múltiples instancias sin sincronización.....	22
Patrón: Múltiples instancias con conocimiento a priori en tiempo de diseño .....	24
Patrón: Múltiples instancias con conocimiento a priori en tiempo de ejecución. ....	26
Patrón: Múltiples instancias sin conocimiento a priori en tiempo de ejecución.....	28
Patrones basados en estado.....	30
Patrón: Selección Diferida.....	30
Patrón: Enrutamiento en paralelo entrelazado.....	31
Patrón: Hito.....	33
Patrones de Cancelación.....	36
Patrón: Cancelar Actividad.....	36
Patrón: Cancelar Caso.....	37

## Introducción

El departamento de “Technology Management” de la “Technische Universiteit Eindhoven” ha identificado y publicado un conjunto de patrones básicos que se encuentran en forma recurrente dentro de la implementación de sistemas de Workflow (<http://is.tm.tue.nl/research/patterns/patterns.htm>).

Este tipo de patrones capturan los aspectos elementales para el control de procesos y son equivalentes a la definición de conceptos inicialmente definida por la Workflow Management Coalition (WfMC) [[Wor99](#)].

Estos patrones se han convertido en un estándar, no solo como formas normalizadas de encarar la solución de las situaciones que estos patrones atacan, sino también como una forma de evaluar la capacidad de las herramientas de Workflow, de acuerdo a su capacidad para implementar en forma natural dichos patrones.

El presente documento describe el cumplimiento de los patrones básicos de Workflow por parte de Q-flow. Se presentará la descripción de cada patrón junto con la forma en que el mismo puede ser realizado utilizando Q-flow.

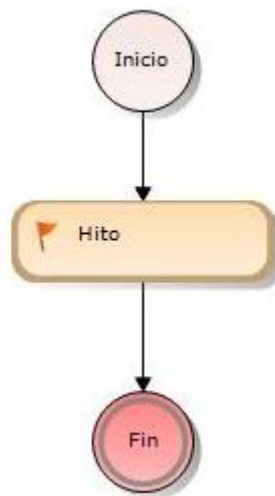
## Patrones de control

### Patrón: Secuencia

Una actividad en un proceso de Workflow es habilitada después de ser completada otra actividad en el mismo proceso.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_1.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_1.swf)

#### Implementación con Q-flow



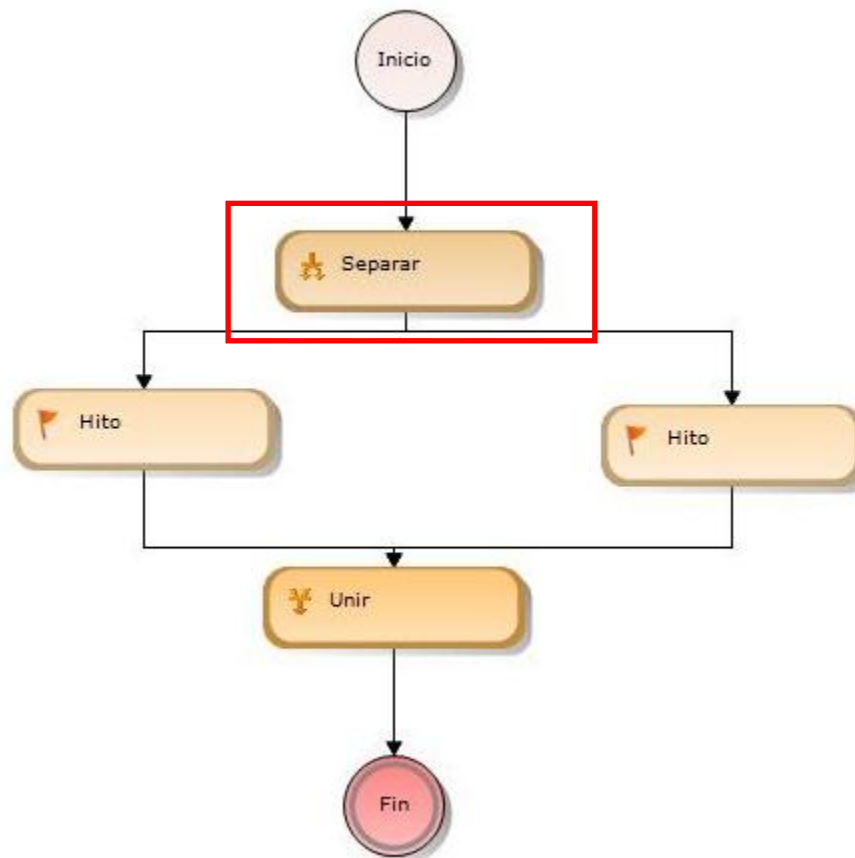
Los diagramas de proceso de Q-flow implementan este patrón por defecto en cada actividad. O sea, por defecto, los conectores entre tareas implican precedencia, salvo que se explicita lo contrario.

### Patrón: Separación en paralelo (también llamado AND/split)

Dos o más actividades de un proceso se ejecutan en paralelo. En un punto del proceso de Workflow, el hilo de control se divide en múltiples hilos de control, habilitando la ejecución de las tareas en paralelo y sin restricciones de orden entre ellas.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_2.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_2.swf)

#### Implementación con Q-flow



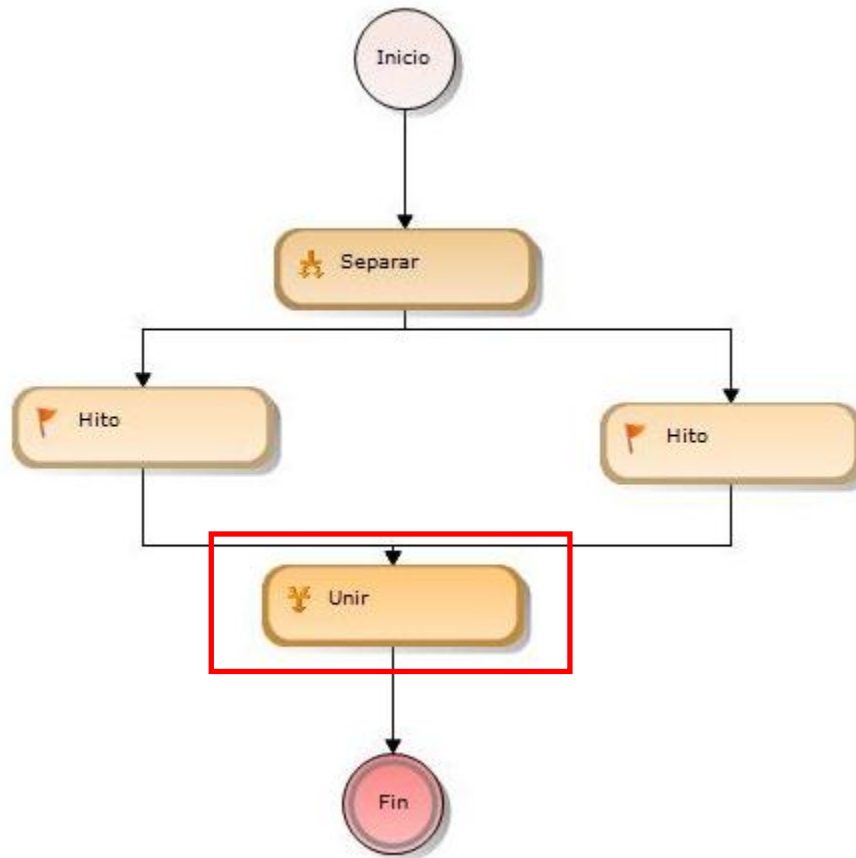
Este patrón se implementa en Q-flow utilizando el paso de separación, el cual separa el hilo de ejecución en múltiples caminos de ejecución paralelos. Q-flow permite la ejecución de múltiples caminos en paralelo (sin límite en cuanto al nivel de anidaciones o contenido de los hilos internos).

## Patrón: Sincronización

Una actividad es iniciada cuando dos o más hilos completan la ejecución de sus actividades. En un punto del proceso, dos o más hilos de control convergen en un solo hilo.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_3.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_3.swf)

### Implementación con Q-flow



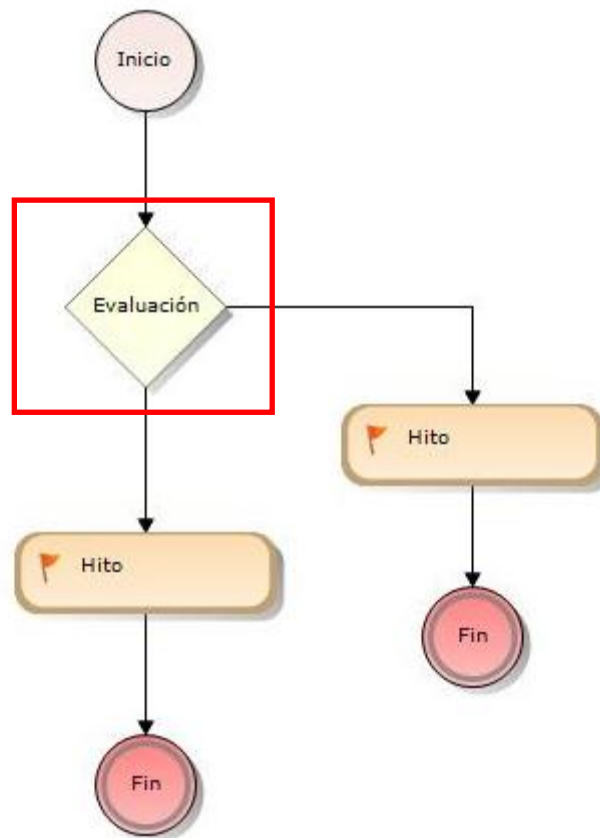
El paso de unión de Q-flow implementa entre otros este patrón. Al ser éste el patrón más común de sincronización, el paso de unión se comporta por defecto de acuerdo al mismo, sincronizando todos los hilos de entrada antes de activar la siguiente actividad.

### **Patrón: Opción exclusiva (XOR-Split)**

En un punto del proceso, una o más de sus ramas son seleccionadas en base a los datos de control del proceso.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_4.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_4.swf)

### **Implementación con Q-flow**



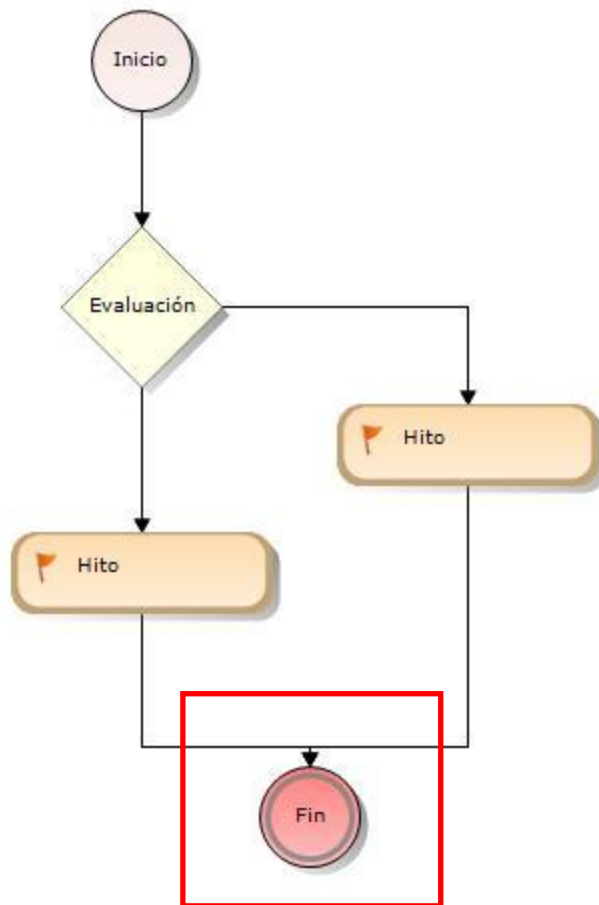
El patrón de selección exclusiva de camino es implementado en Q-flow por medio del paso de evaluación, donde es evaluada una condición (basada en respuestas dadas a preguntas o tareas, votaciones, evaluación de datos relevantes del workflow o valores de propiedades de los usuarios participantes del workflow) y de acuerdo al valor de la evaluación se selecciona uno u otro camino de ejecución.

## Patrón: Fusión simple

Se mezclan varios hilos de control en uno solo, en determinado punto del proceso.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_5.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_5.swf)

**Implementación con Q-flow**



La fusión simple en Q-flow no utiliza ningún constructor específico, sino que la simple unión de los caminos o hilos de ejecución sobre una tarea determina que cualquiera de ellos que llegue a la tarea la active.

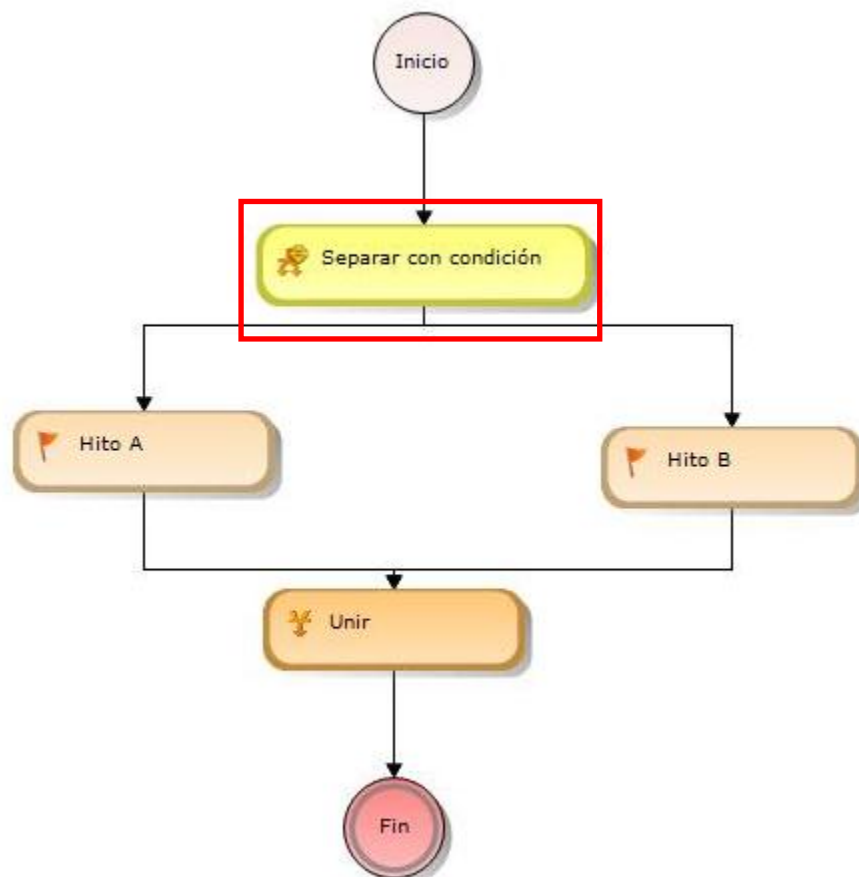
## Patrones avanzados de paralelismo y sincronización

### Patrón: Múltiple Opción (Or-Split)

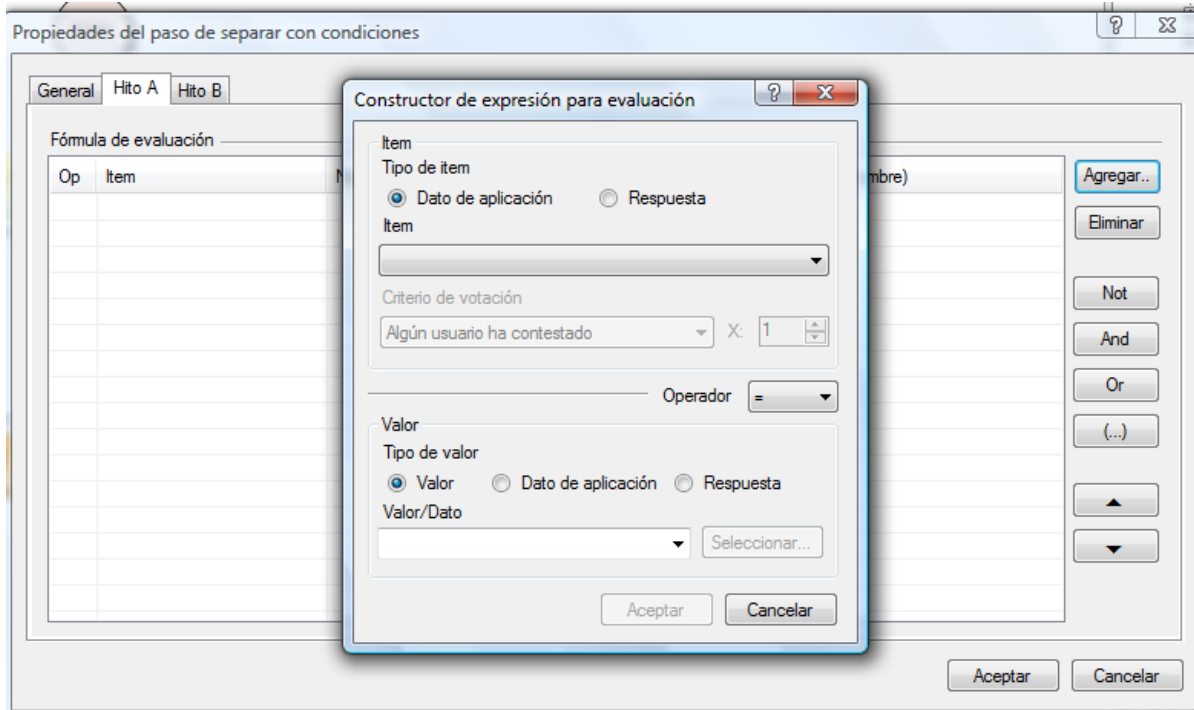
En un punto del proceso, basado en los datos de control del proceso, uno o más caminos o ramas son seleccionados.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_6.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_6.swf)

### Implementación con Q-flow



El paso de Separación de Q-flow implementa este patrón incluyendo una evaluación opcional en cada camino de ejecución en que abre el proceso, permitiendo que cada camino sea activado opcionalmente, de acuerdo con el resultado de la evaluación correspondiente (verdadero o falso).

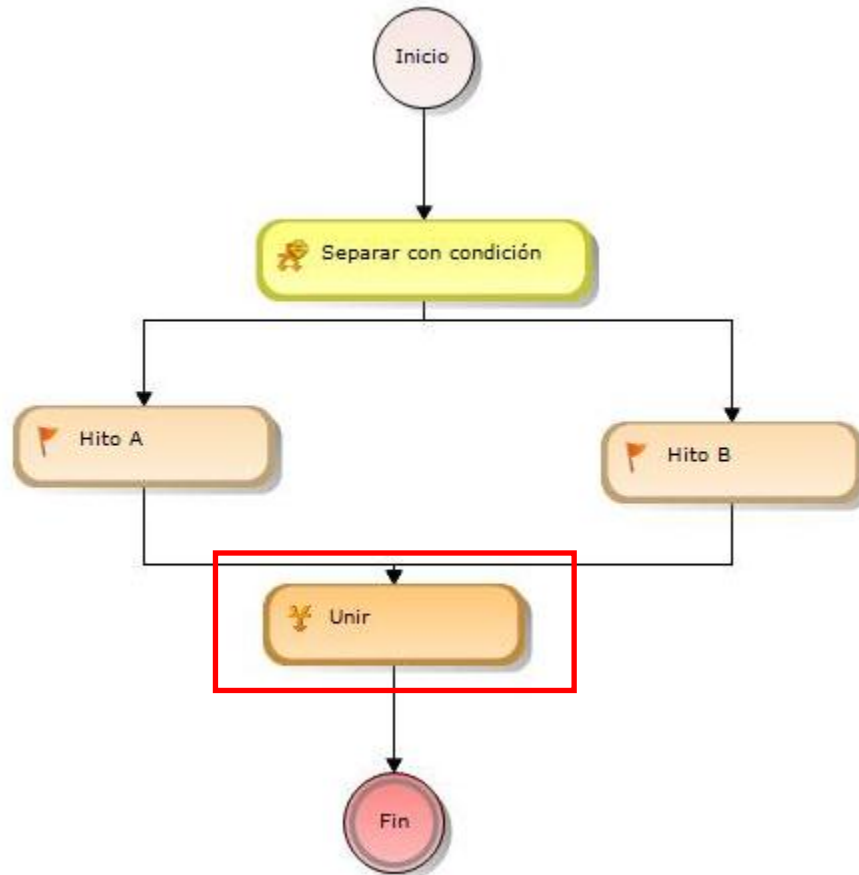


## Patrón: Fusión Sincronizada

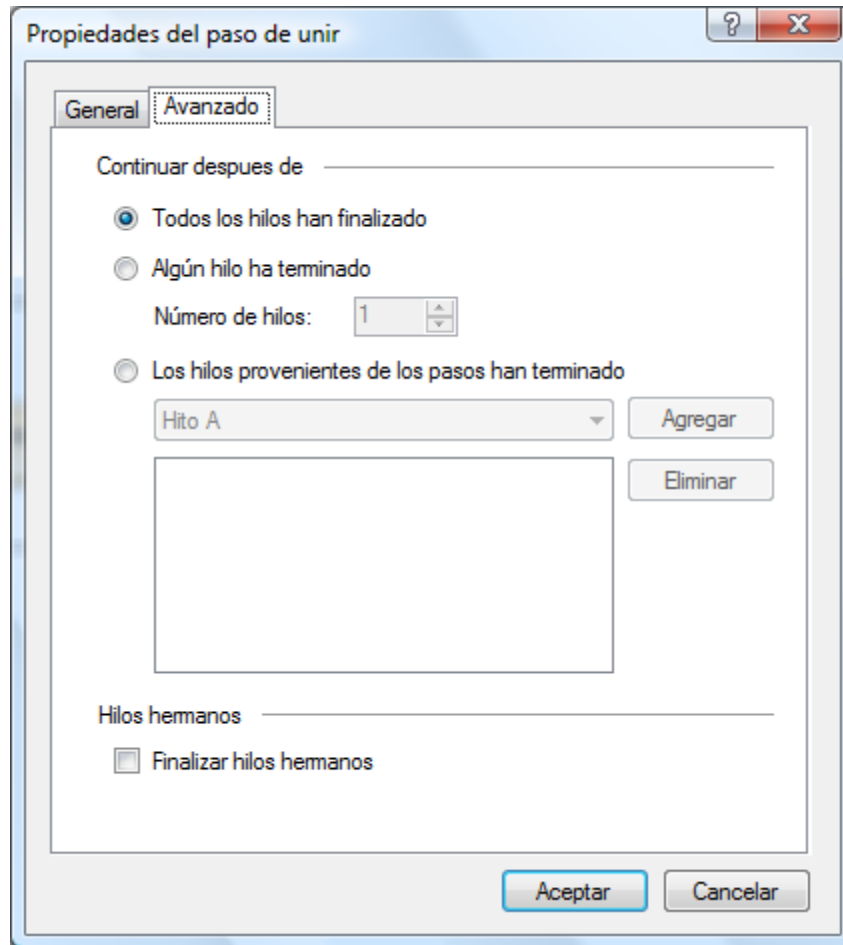
Se trata de un punto en el proceso de Workflow donde múltiples caminos convergen en un único hilo de control. Si más de un camino había sido tomado, entonces debe producirse una sincronización entre los hilos de control. Si había sido tomado sólo uno, los demás hilos deben converger.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_7.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_7.swf)

**Implementación con Q-flow**



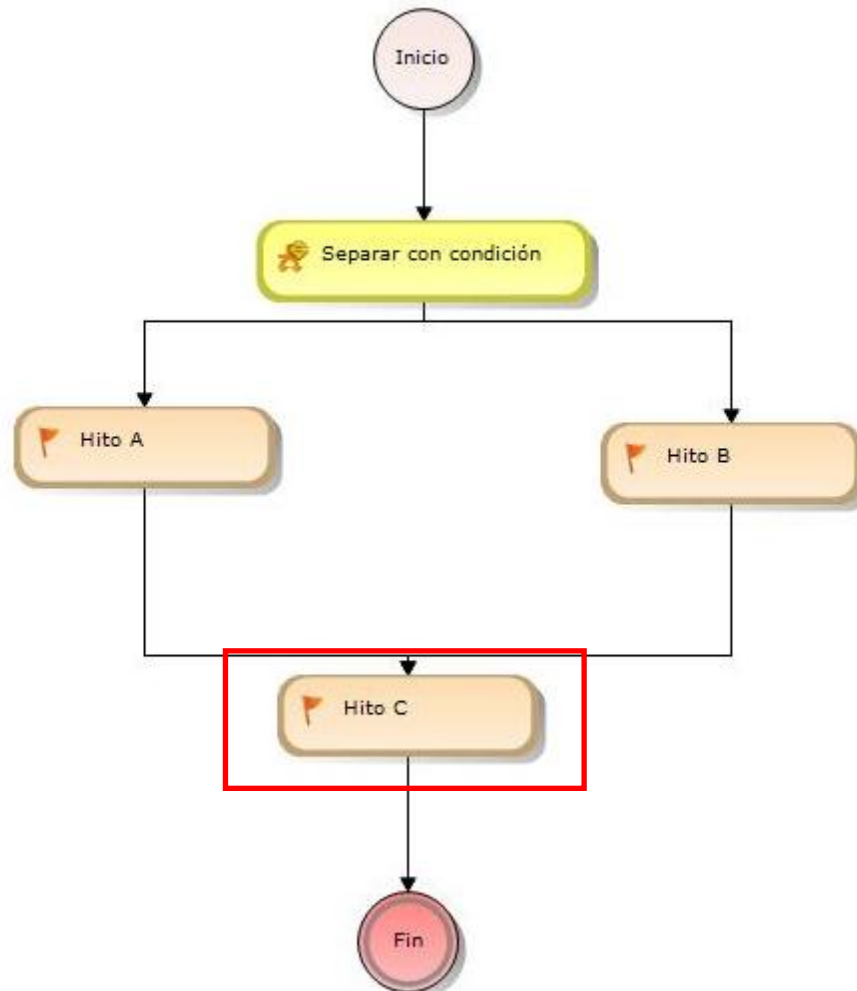
El paso de unión de Q-flow implementa este patrón. El mismo, en su configuración, permite que se seleccione la espera por todos los hilos iniciados en el fork/split correspondiente, de forma de que espere la finalización de todos los hilos antes de activar la siguiente tarea.



## Patrón: Fusión Múltiple

En un una actividad convergen distintas ramas o caminos en los que se abrió el hilo de control en una bifurcación anterior. La actividad donde convergen los hilos será activada una vez por cada hilo de la rama que se activó en la separación.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_8.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_8.swf)



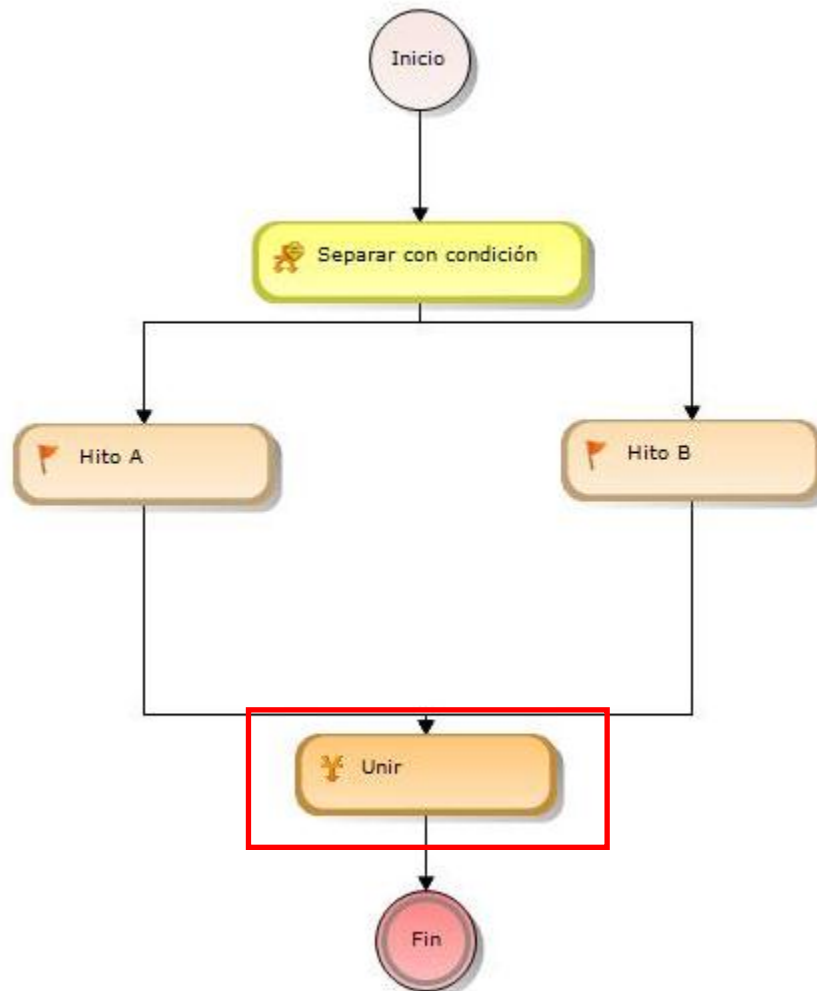
Este patrón está implementado en Q-flow por medio de una separación con condición en la cual los hilos se separan condicionalmente y continuando la ejecución de cada hilo en un paso en común. Se mantienen uno o dos hilos dependiendo de la separación y cada uno de los hilos puede estar en el mismo paso del proceso.

### Patrón: Discriminador

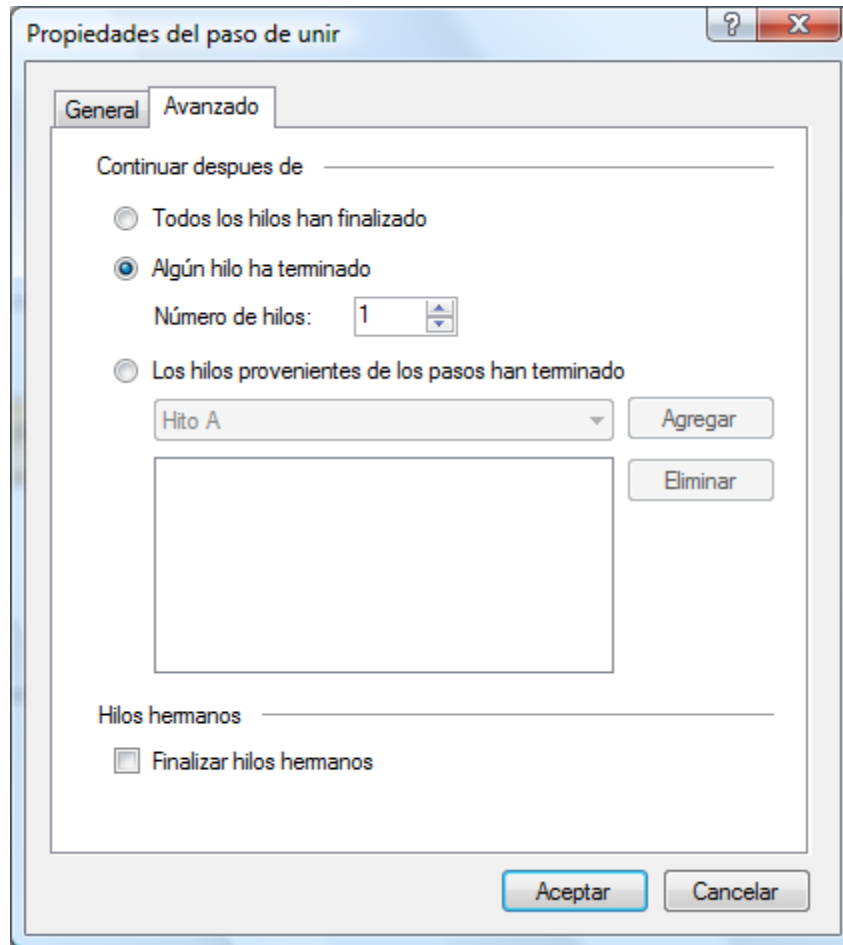
Es un punto del proceso que espera por una rama o hilo de control antes de activar la siguiente actividad. Los siguientes hilos de control que lleguen a ella luego de activada la tarea siguiente son ignorados, hasta que al llegar el último la tarea finaliza.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_9.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_9.swf)

## Implementación con Q-flow

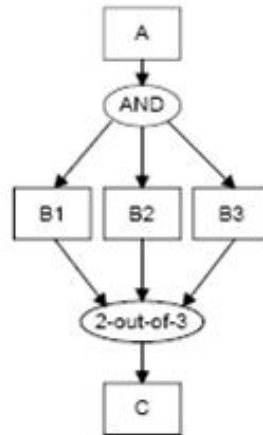


Este patrón es implementado en Q-flow por medio del paso de Unión, en el cual varios hilos de ejecución se fusionan, permitiendo que se active la siguiente actividad en base a la cantidad de hilos o caminos de ejecución que alcanzan el paso de unión. Para implementar este patrón, en las propiedades del paso de unión se selecciona la opción continuar cuando ha llegado el primer hilo.

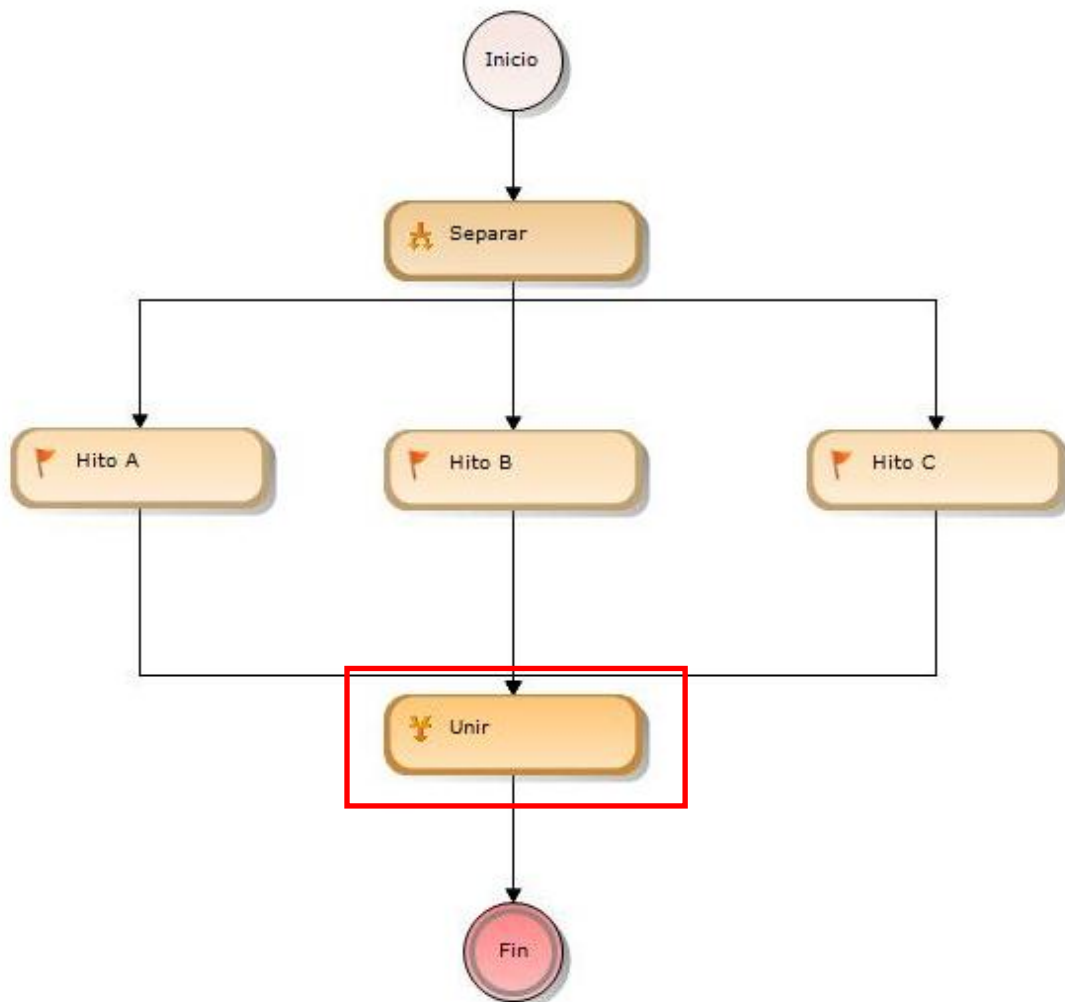


### Patrón: Unión, N-de-los-M

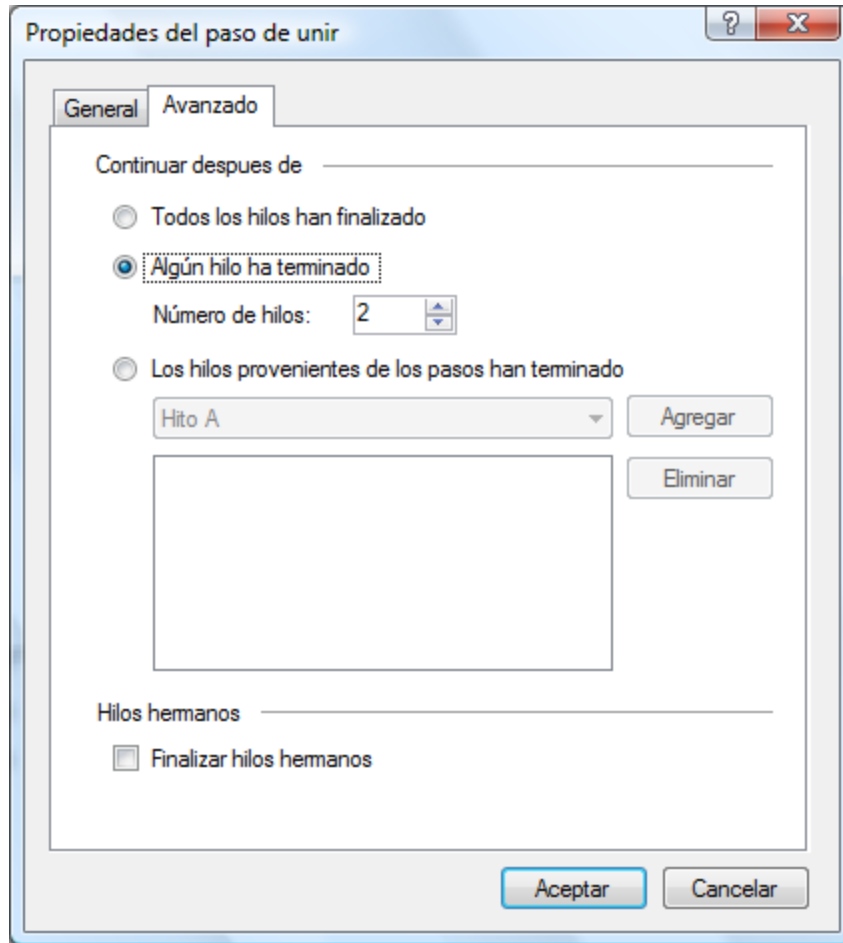
Unión N-de-los-M , es un punto en el proceso de workflow donde M caminos de ejecución paralelos convergen en uno. La siguiente actividad es activada sólo cuando N caminos se han completado. La completación de los siguientes M-N caminos es ignorada.



## Implementación con Q-flow



Este patrón es implementado en Q-flow por medio del paso de Unión, en el cual varios hilos de ejecución se fusionan, permitiendo que se active la siguiente actividad en base a la cantidad de hilos o caminos de ejecución que alcanzan el paso de unión. Para implementar este patrón, en las propiedades del paso de unión se selecciona la opción continuar cuando han llegado N hilos.

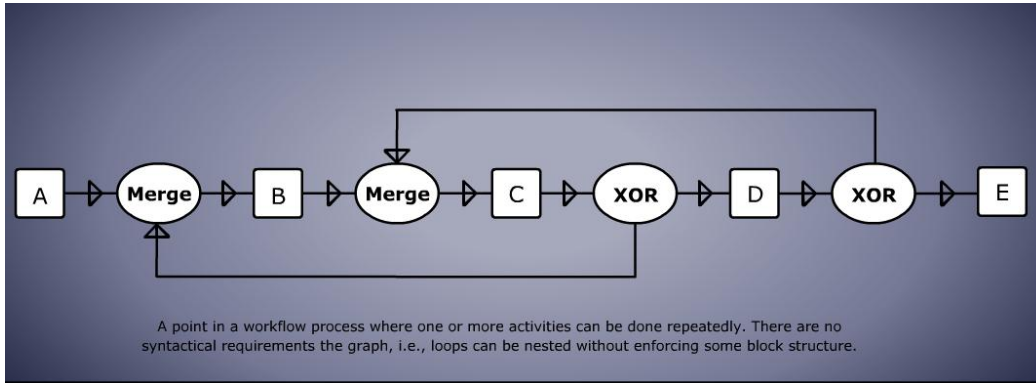


## Patrones Estructurales

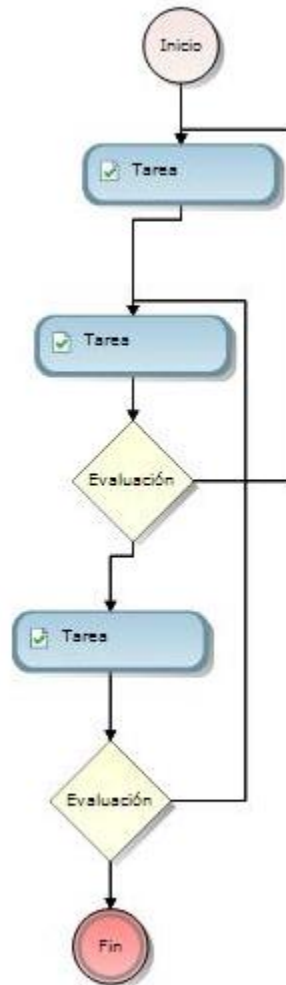
### Patrón: Ciclos Arbitrarios

Se trata de un punto en el proceso donde una o más actividades pueden ser realizadas en forma repetitiva. Este patrón existe para diferenciar los Workflows que necesitan de un constructor explícito de repetición (como lo sugiere el WPDL).

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_10.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_10.swf)



## Implementación con Q-flow



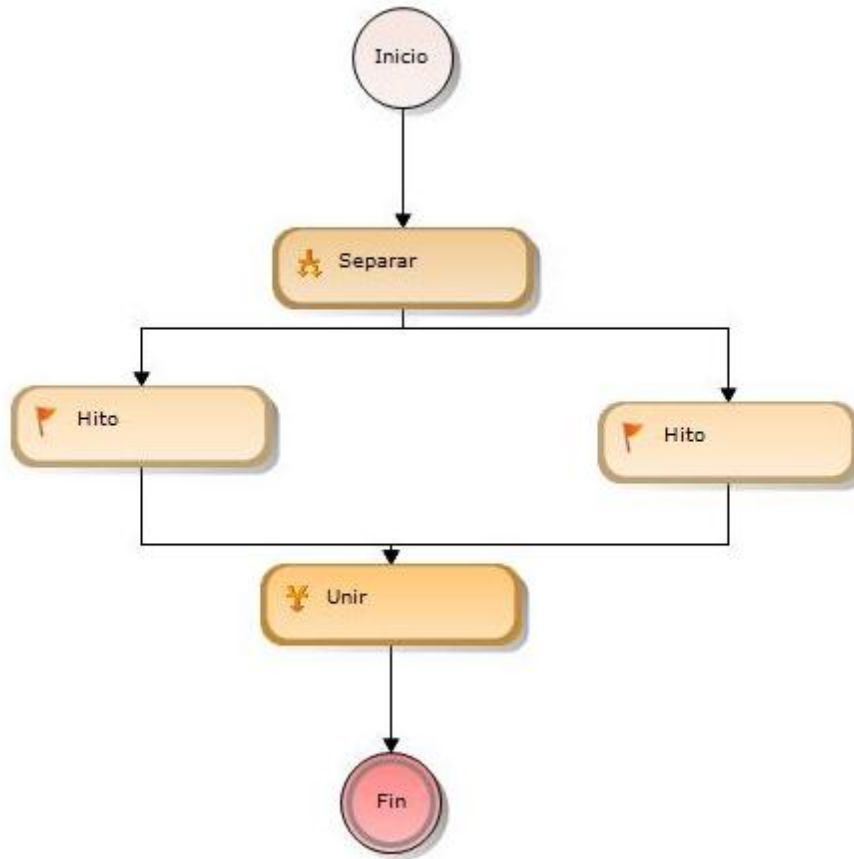
Al igual que lo que sucede con el patrón de “Fusión Simple”, el modelo de grafo que implementa Q-flow no necesita de un constructor específico para implementar la repetición o el patrón de “Ciclos Arbitrarios”. La simple unión de una bifurcación con el punto de entrada de una tarea o actividad de Q-flow es una implementación de este patrón.

### Patrón: Terminación Implícita

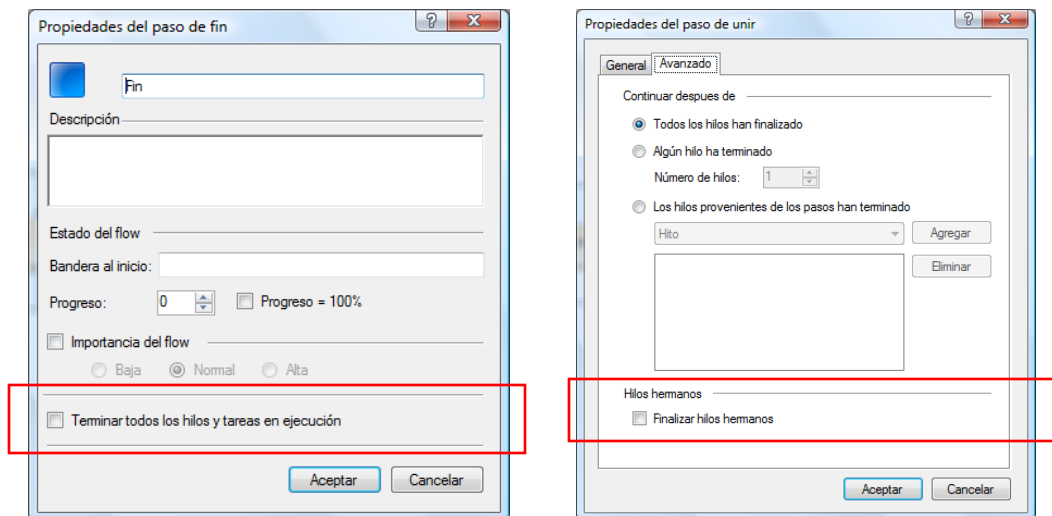
Un proceso debe terminar cuando no quedan actividades pendientes y cuando no está en deadlock. Se evita un terminador explícito que cancela las actividades pendientes.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_11.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_11.swf)

## Implementación con Q-flow



Las propiedades de los pasos de unión y de terminación muestran la capacidad del sistema para mantener activas las tareas en los hilos o caminos activos luego de haberse procesado los pasos de unión y/o de cancelación o de lo contrario finalizarlas implícitamente.



### Patrones de tareas con múltiples instancias

Existen múltiples formas de abordar estos patrones. Durante el presente análisis, se abordará la forma más común de implementarlos, que es con pasos de tarea, o sea, con actividades asignadas a usuarios.

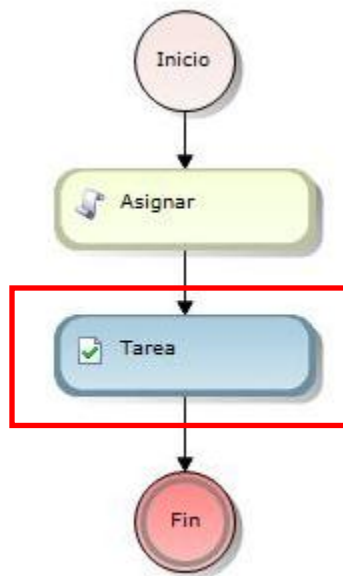
#### Patrón: Múltiples instancias sin sincronización.

Una actividad es instanciada múltiples veces, desconociéndose en tiempo de diseño el número de instancias a ser habilitadas. Cuando todas las instancias creadas finalizan, se ejecuta la siguiente actividad.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_12.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_12.swf)

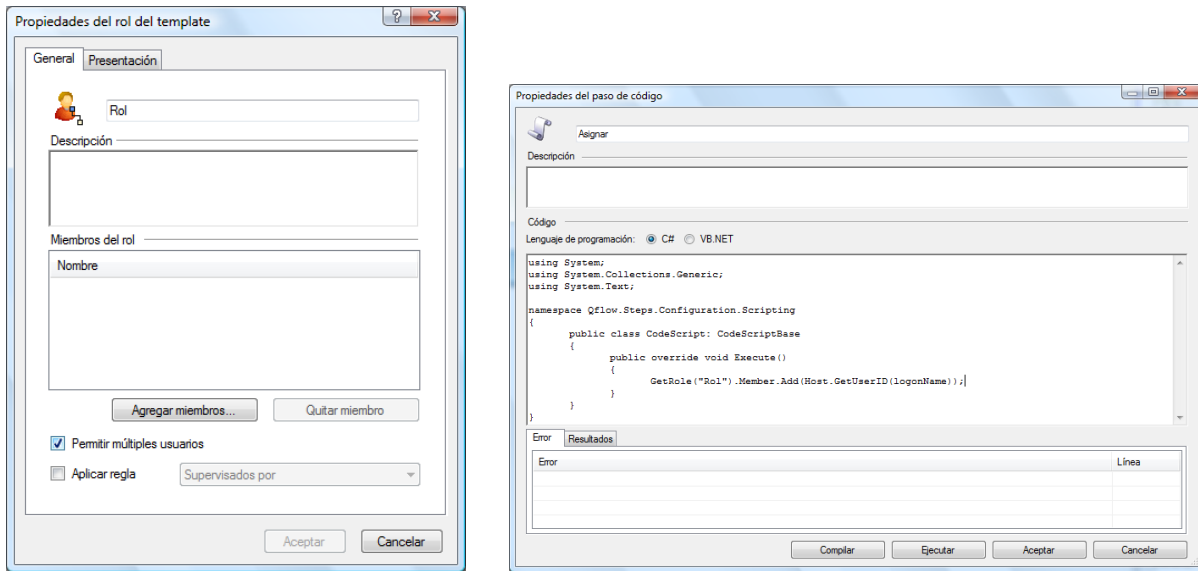
#### Implementación con Q-flow

Dicho patrón puede ser implementado con Q-flow utilizando una tarea con un destinatario multivaluado. El destinatario multivaluado puede ser cargado, generándose automáticamente las correspondientes instancias de la tarea en el momento de instanciarse la misma.

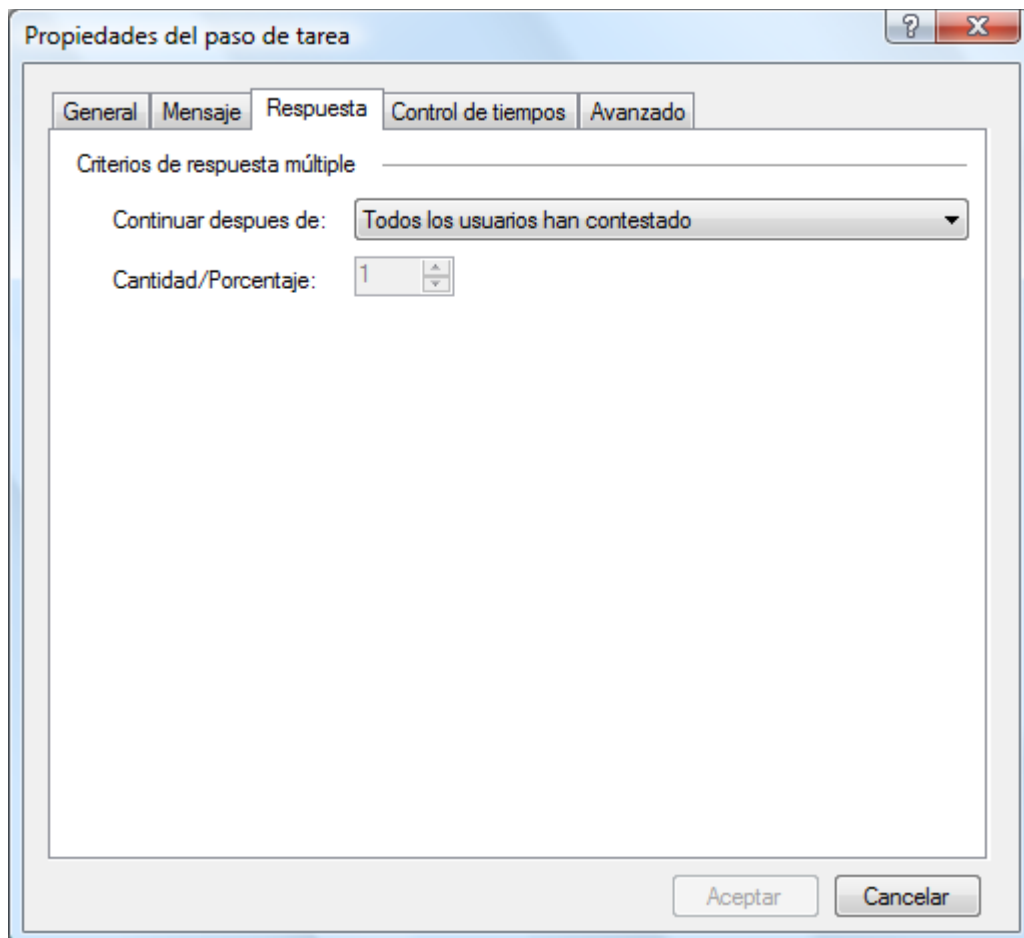


# Patterns & Practices

Asignación del rol en tiempo de ejecución:



Configuración del paso de tarea para que espere a que se completen todas las instancias antes de ejecutar la siguiente actividad:



### Patrón: Múltiples instancias con conocimiento a priori en tiempo de diseño

Se soporta que una actividad sea habilitada o invocada muchas veces, con la cantidad de veces conocida en tiempo de diseño.

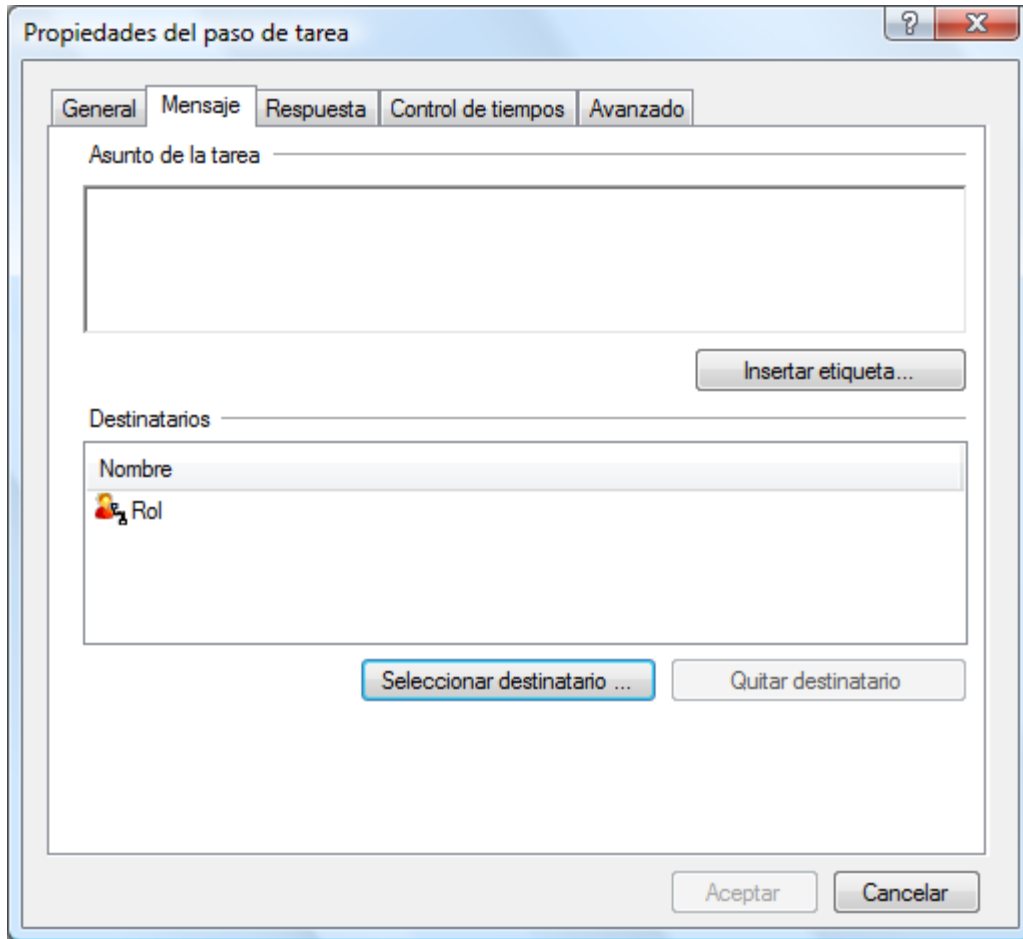
[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_13.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_13.swf)

#### Implementación con Q-flow

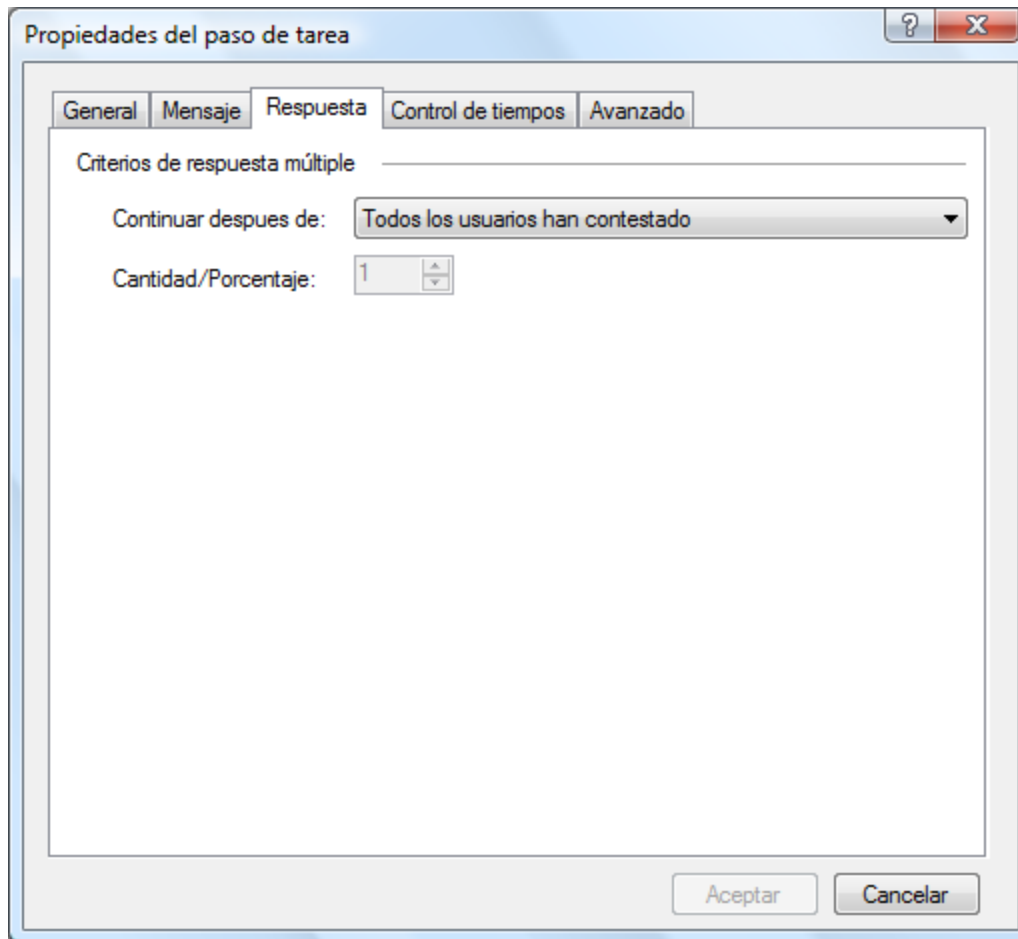
Este patrón también es de simple implementación en Q-flow, ya que la configuración natural de los pasos permite que se determinen en tiempo de diseño la cantidad de instancias de la actividad a ser instanciada en tiempo de ejecución.



El paso de pregunta es configurado con la lista de destinatarios (tomadas de diversas fuentes) a quienes se dirigirá la pregunta o tarea.



En la configuración del paso, se especifica que debe esperar a que se completen las actividades antes de continuar con la siguiente actividad.



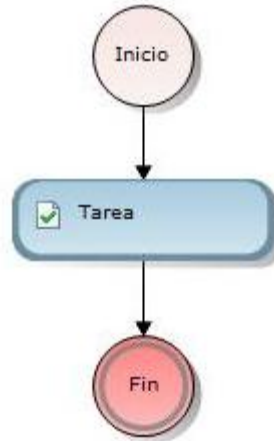
### **Patrón: Múltiples instancias con conocimiento a priori en tiempo de ejecución.**

Una actividad es instanciada más de una vez, dependiendo de información disponible en tiempo de diseño, antes de invocar la actividad. Esta información puede provenir de los datos de control del proceso o de los recursos del sistema, etc.

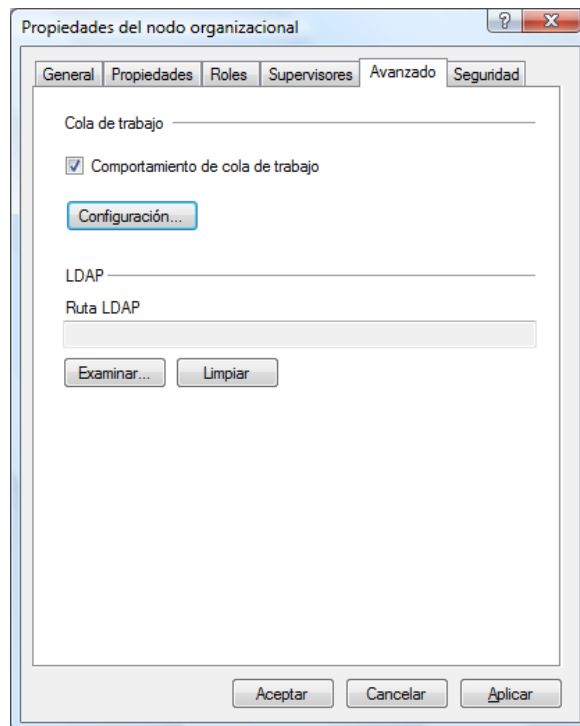
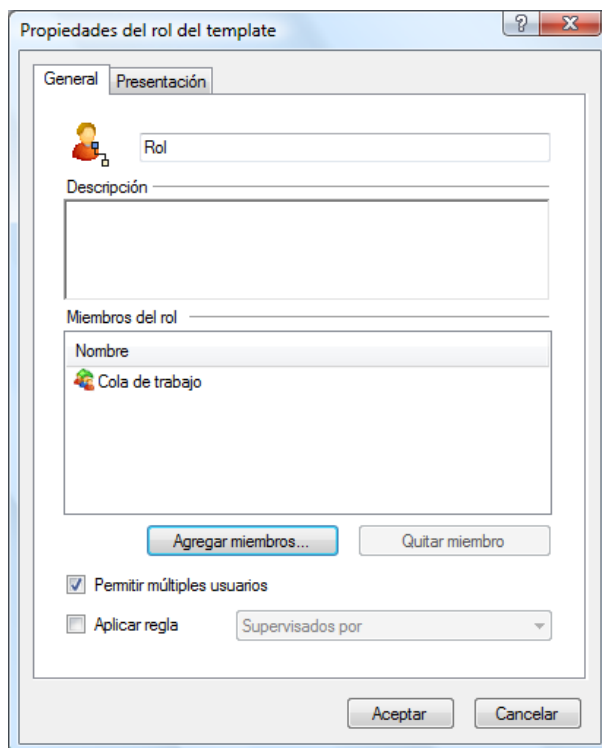
[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_14.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_14.swf)

#### **Implementación con Q-flow**

Dicho patrón puede ser implementado con Q-flow utilizando una tarea con un destinatario multivaluado, donde la tarea es asignada en tiempo de ejecución de acuerdo a la información disponible en ese momento.



En este patrón, antes de instanciarse la actividad, se analiza el valor de un dato de aplicación del proceso y mediante el script especificado en la siguiente imagen se generan tantas instancias del destinatario de la tarea como sean necesarias, asignándose cada instancia a la cola de trabajo.



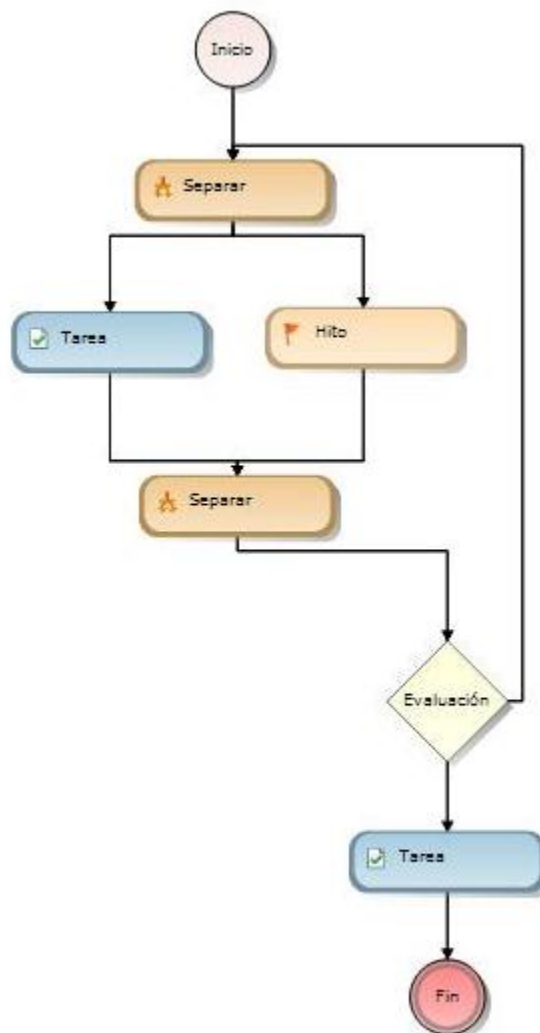
## Patrón: Múltiples instancias sin conocimiento a priori en tiempo de ejecución.

Una actividad es instanciada muchas veces en tiempo de ejecución, sin conocimiento de cuántas instancias se crearán ni en tiempo de diseño ni en tiempo de ejecución, hasta que se ha terminado de crear las instancias. Es el caso de un bucle en tiempo de ejecución que instancia una actividad.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_15.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_15.swf)

### Implementación con Q-flow

Este patrón se implementa abriendo en dos el hilo de ejecución en un paso de separación (Separar) para que la tarea se active y continuando la ejecución luego del paso de unión, configurando el mismo para que continúe al llegar el primer hilo de ejecución (patrón discriminador), de forma que la tarea continúa activa y cuando termina simplemente se detiene su hilo de ejecución al llegar al paso de unión. El evaluador implementa el patrón X-OR, permitiendo que se activen tantas instancias de la misma tarea como sean necesarias, hasta que la condición de evaluación del X-OR determine que se debe activar la tarea B.



## Patterns & Practices

Nota: Si se debe esperar por todas las instancias de A para activar B, entonces se debe utilizar un contador en el hilo de A, incrementándolo antes de entrar a A y disminuyéndolo al salir, condicionándose el pasaje a B sólo cuando ese contador llegue a cero. La evaluación del mismo puede dispararse con un paso de espera, para sincronizarse con las novedades de parte del workflow.

## Patrones basados en estado

### Patrón: Selección Diferida

Se trata de un punto en el proceso donde uno o múltiples caminos son seleccionados.

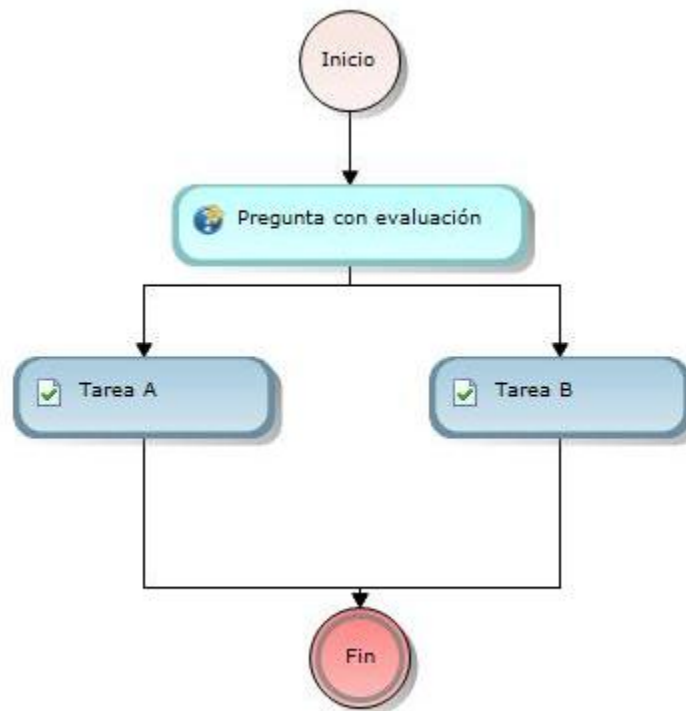
- En contraste con el XOR-Split, la selección no es explícita (basada en datos o decisiones) pero múltiples alternativas son ofrecidas por el entorno.
- En contraste con el AND-Split, sólo una de las opciones es seleccionada.

Supongamos que en un punto del proceso se debe completar una actividad u otra basada en la disponibilidad de los recursos, pero ninguno de los recursos utilizables está disponible, por lo que se debe esperar a que uno de los recursos esté disponible para decidir cuál alternativa seleccionar.

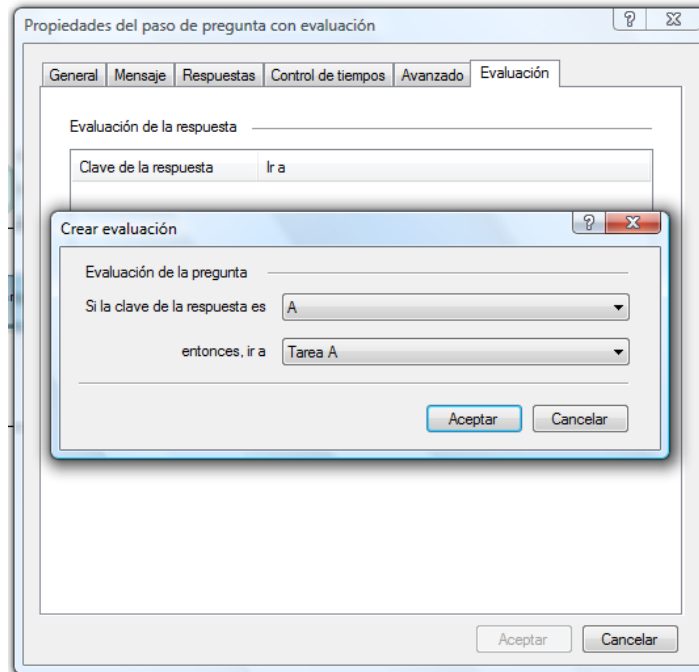
[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_16.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_16.swf)

### Implementación con Q-flow

El paso de pregunta con evaluación permite seleccionar la opción a ser tomada en tiempo de ejecución.



En base a la respuesta dada, selecciona el camino a tomar o la tarea a activar.



### **Patrón: Enrutamiento en paralelo entrelazado.**

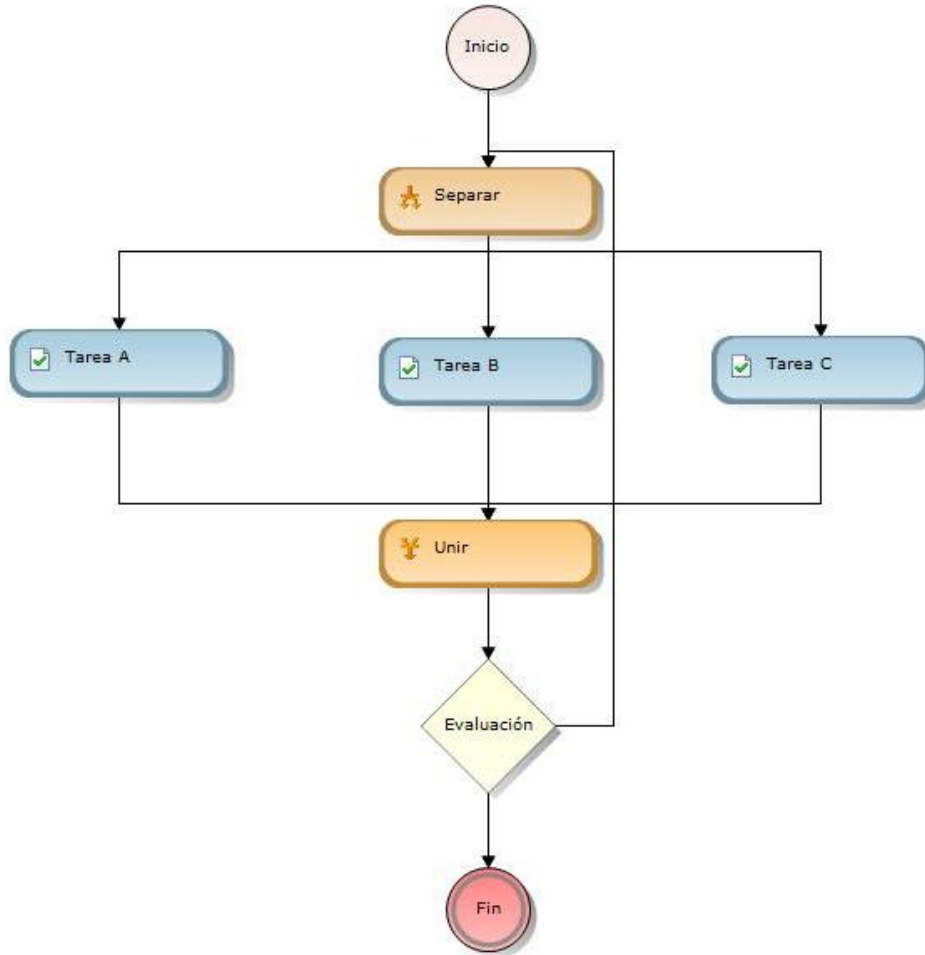
Un conjunto de actividades es ejecutado en un orden arbitrario, decidido en tiempo de ejecución.

Es el caso de actividades que pueden ejecutarse en cualquier orden pero no en paralelo.

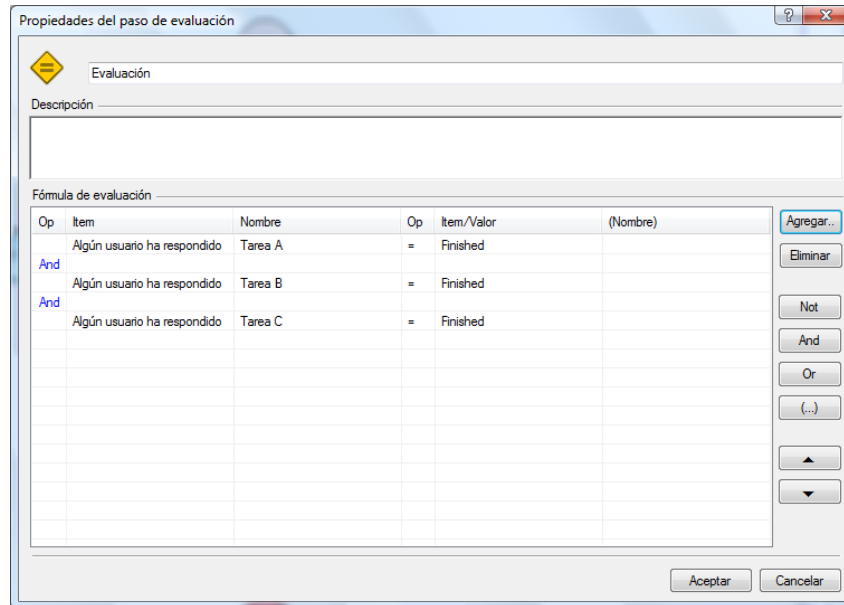
[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_17.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_17.swf)

### **Implementación con Q-flow**

El paso de separación de Q-flow, si bien no permite directamente implementar un modelo en el cual se seleccione por sí solo un camino de ejecución, sí permite que cada hilo disponga de un evaluador implícito, por lo cual a cada uno de los tres hilos que se muestran en el siguiente diagrama se le puede dar una condición de inicio. Una vez establecido un criterio de selección que permita que sólo uno de los hilos actúe en cada iteración, alcanza con implementar un bucle por medio del patrón x-or de forma que cierre el ciclo de ejecución repitiéndolo hasta que todas las tareas se hayan completado.



La siguiente figura muestra las propiedades a colocar en el paso de evaluación para que continúe ejecutando otra tarea mientras no se han completado o instanciado las tres tareas especificadas.



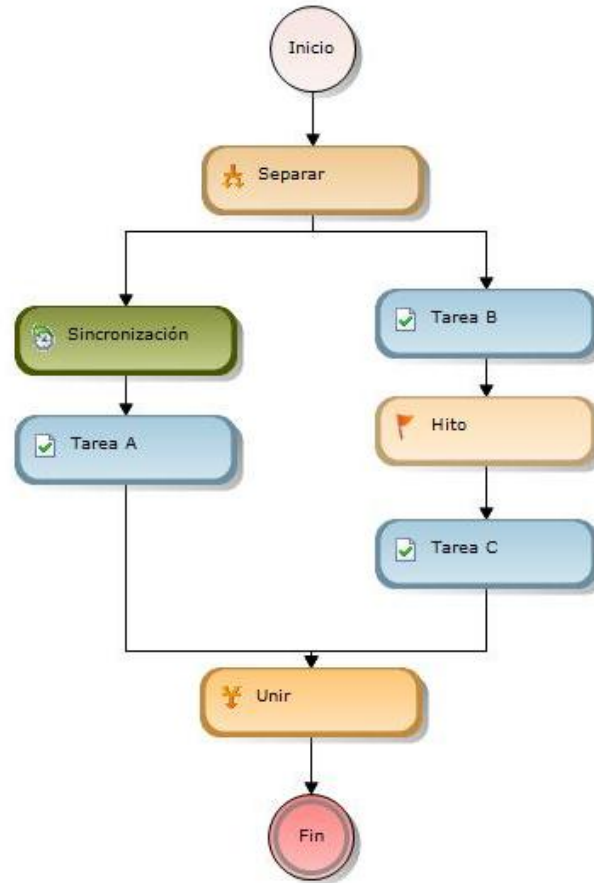
## Patrón: Hito

Permite testear si un proceso de Workflow ha alcanzado determinada fase o etapa.

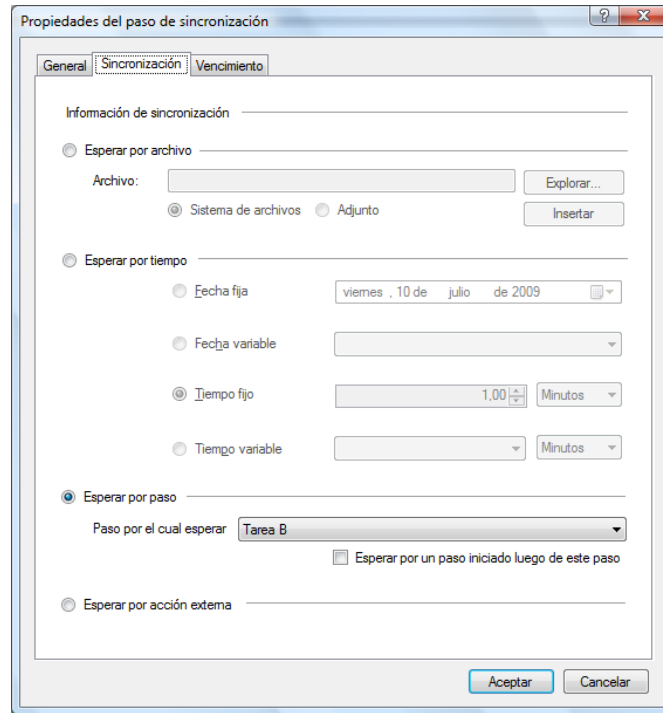
[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_18.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_18.swf)

## Implementación con Q-flow

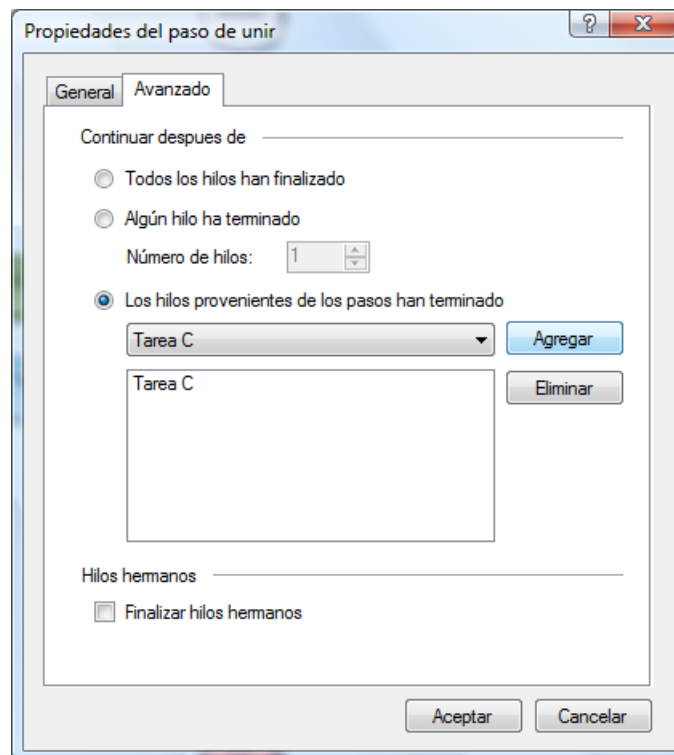
Este patrón es implementado en forma nativa por Q-flow. En el siguiente diagrama, se abren dos hilos de ejecución paralela y en el primero de los hilos se espera por la ejecución del hito (paso Sincronización) para activar la tarea A. Se continúa inmediatamente con la tarea C (con dos hitos y esperas se puede hacer una sincronización cruzada). Al llegar al paso de unión, la ejecución continúa.



En la configuración del paso de “sincronización” se muestra que se esperará por el hito contenido en el otro hilo.



El paso de unión puede ser configurado para que la ejecución continúe cuando se ha completado el segundo hilo de ejecución, independizándose la continuación del workflow de la terminación o de la ejecución de la tarea A, la cual, según diversos caminos que podría tomar el segundo hilo, podría no ejecutarse nunca.



## Patrones de Cancelación

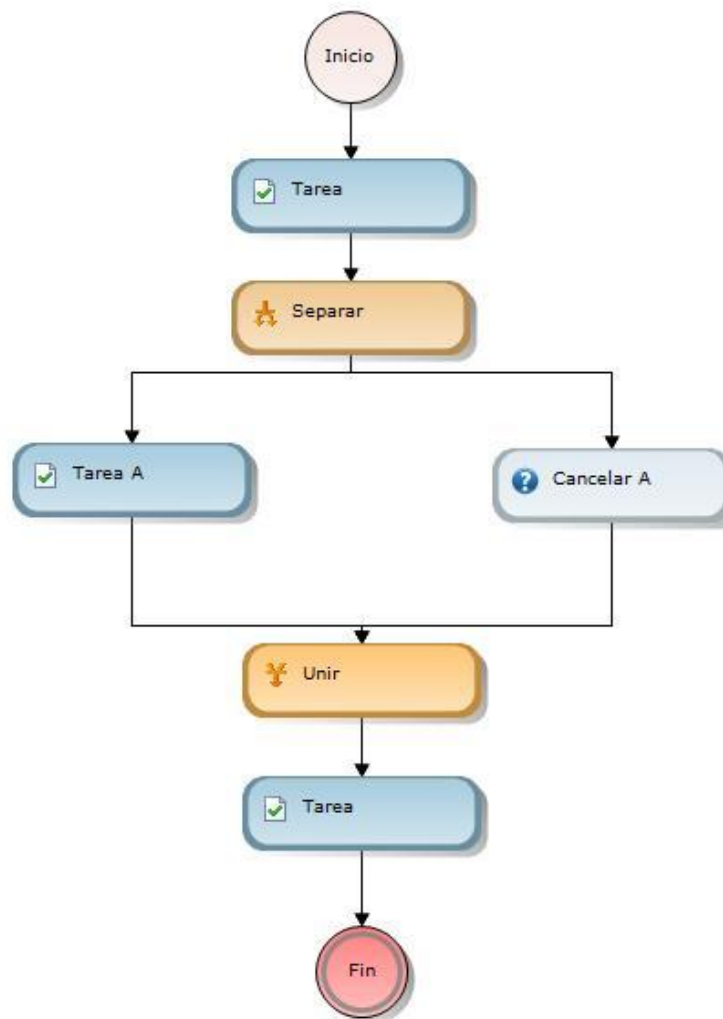
### Patrón: Cancelar Actividad

El sistema permite que al completarse una actividad otra actividad sea cancelada o deshabilitada (una actividad que una vez completada hace innecesaria otra que se encuentra activada).

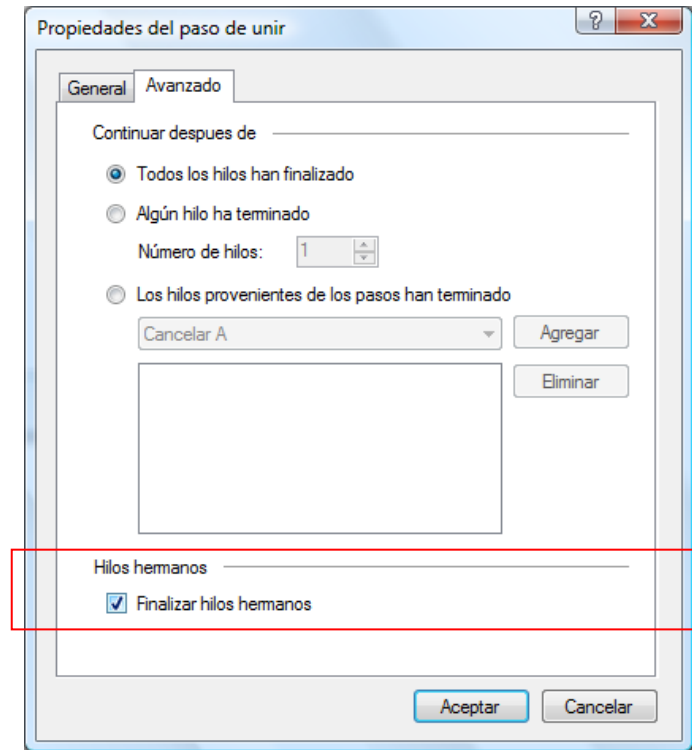
[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_19.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_19.swf)

### Implementación con Q-flow

Se inicia el proceso, hasta que en un punto nos encontramos con un paso de fork que divide el mismo en dos hilos: uno que representa el hilo de ejecución normal del proceso y el otro que representa el hilo de cancelación.



La unión de los hilos cancela las tareas definidas. Se muestra como se debe configurar el paso de unión.



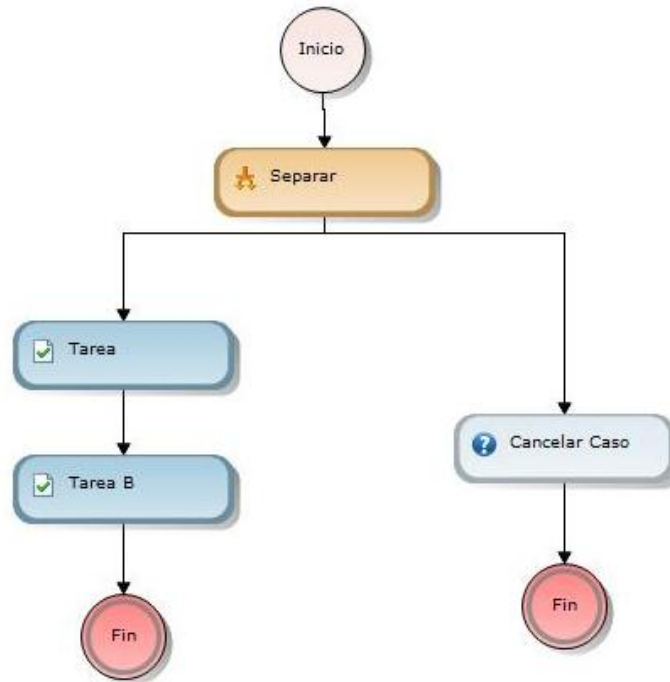
### Patrón: Cancelar Caso.

Una instancia completa del proceso es cancelada. Por ejemplo, en un proceso de evaluación y selección, uno de los participantes se retira, teniendo que cancelarse todas las actividades relacionadas con la evaluación de este participante.

[http://is.tm.tue.nl/research/patterns/download/swf/pat\\_20.swf](http://is.tm.tue.nl/research/patterns/download/swf/pat_20.swf)

### Implementación con Q-flow

Se inicia el proceso con un paso de fork que divide el mismo en dos hilos: uno que representa el hilo de ejecución normal del proceso y el otro que representa el hilo de cancelación.



El paso de finalización cancela el proceso en su conjunto. Se muestra la configuración del paso de fin.

