

# **Guías para comunicar un diseño**

**Diagrama de actividades**

por

*Fernando Dodino*

**Versión 1.2**

**Marzo 2014**

# ***Índice***

- [1 Diagrama de actividades](#)
  - [1.1 Elementos](#)
- [2 Ejemplos](#)
  - [2.1 “Cuenta corriente”](#)
  - [2.2 “Fast-food” \(conurrencia\)](#)
- [3 Resumen](#)

# 1 Diagrama de actividades

## 1.1 Elementos



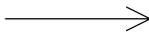
**Actividad/acción (activity/action state):** representa algún tipo de procesamiento (“Vender producto”, “Reservar libro”, “Seleccionar tipo de combustible”, etc.). Una actividad es susceptible de descomponerse en varias sub-actividades (se las puede detallar aparte en otros diagramas de actividad). Algunos autores distinguen entre *actividades* y *acciones*, donde las segundas son atómicas (no pueden descomponerse en sub-acciones).



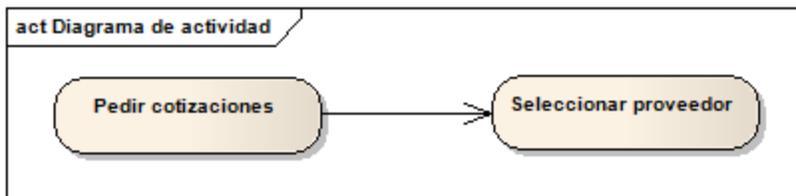
**Estado inicial:** permite identificar el origen de las transiciones por las que va a pasar el caso de uso. Un diagrama de actividades tiene un solo estado inicial.



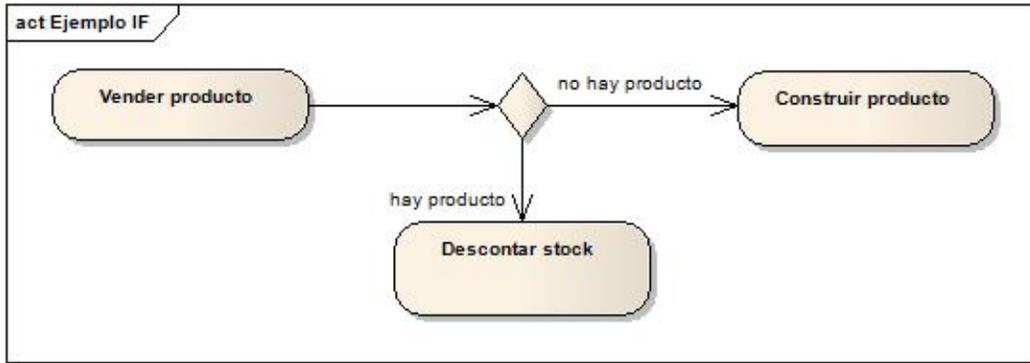
**Estado final:** es el último eslabón del flujo de transiciones del diagrama de actividades.



**Transición (control-flow):** refleja el paso de una actividad a otra; permite así indicar precedencias entre dos actividades. *Ejemplo:*

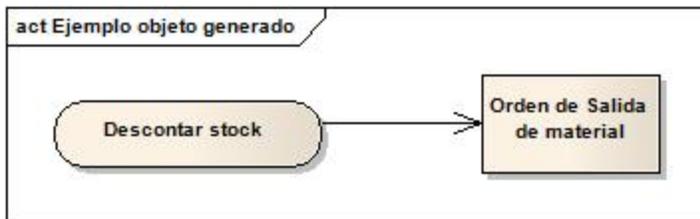


**Decisión (if):** en base a una condición se procesará una acción u otra. *Ejemplo:*



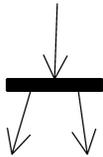
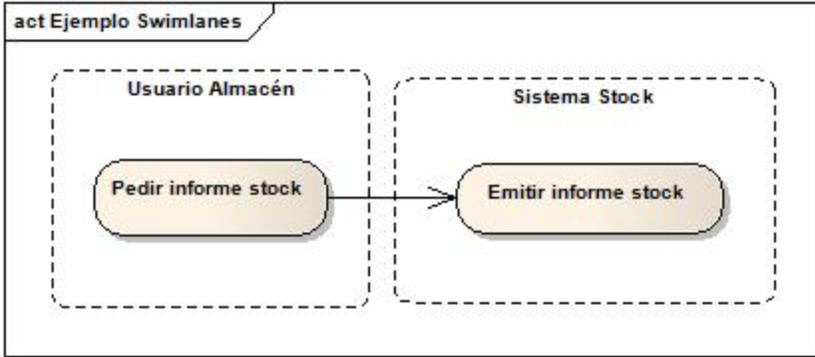
ObjectFlowState1

**Objeto:** se producen como resultado de las actividades. En el ejemplo anterior, la venta de un producto hace que la acción “Descontar stock” genere una orden de salida del material que está en el Depósito. Gráficamente:

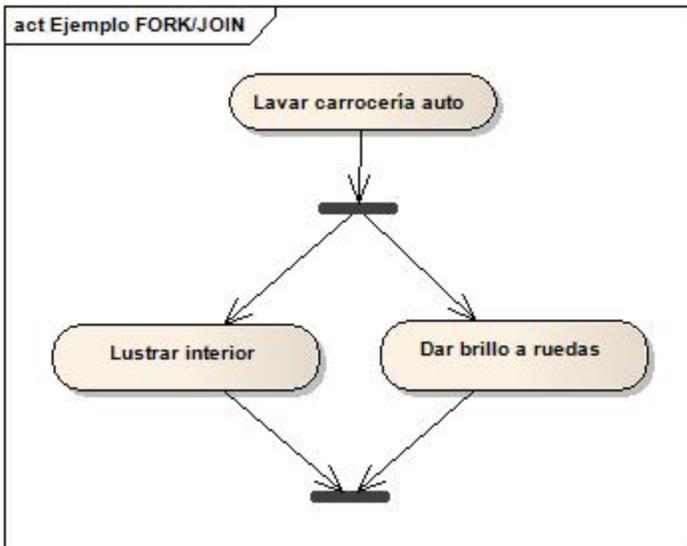


Partition1

**Regiones (Activity Region/Swimlane):** permite identificar las responsabilidades de los diferentes actores en un caso de uso. Las actividades y el flujo de transiciones se van particionando de manera que queda claro qué acción debe ejecutar cada uno de los actores. *Ejemplo:*



**Acciones concurrentes (forks/joins):** son actividades que pueden ocurrir en paralelo. *Ejemplo:* en un lavadero de autos una vez que lavaron la carrocería hay personas encargadas de lustrar el interior del auto y de dar brillo a las ruedas. Gráficamente:



El JOIN implica que todas las actividades posteriores deben sincronizar las tareas ejecutadas en paralelo. En el ejemplo de arriba, lustrar el interior y lavar las ruedas son acciones que deben completarse juntas para que la siguiente acción se lleve a cabo.

## 2 Ejemplos

### 2.1 “Cuenta corriente”

Vamos a modelar el pago de un cliente a través de un cajero por mostrador. El cajero identifica al cliente en la cuenta corriente del sistema, seleccionando el criterio de búsqueda:



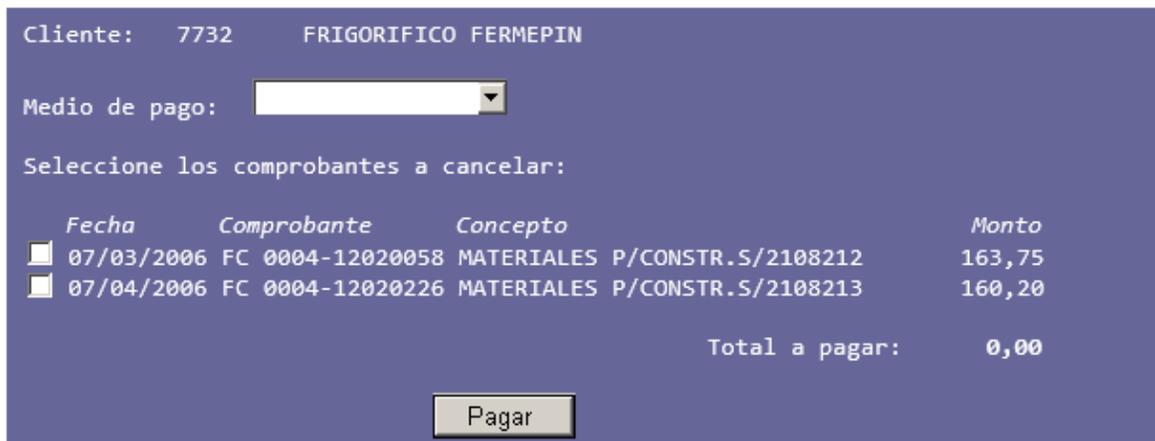
Consulta de cuenta corriente

Nº cliente:  Tipo Comprobante:

Razón Social:  Número:

Domicilio:

El cajero le indica los comprobantes pendientes de pago, entonces el cliente elige el medio de pago y los comprobantes a cancelar:



Cliente: 7732 FRIGORIFICO FERMEPIN

Medio de pago:

Seleccione los comprobantes a cancelar:

<input type="checkbox"/>	Fecha	Comprobante	Concepto	Monto
<input type="checkbox"/>	07/03/2006	FC 0004-12020058	MATERIALES P/CONSTR.S/2108212	163,75
<input type="checkbox"/>	07/04/2006	FC 0004-12020226	MATERIALES P/CONSTR.S/2108213	160,20

Total a pagar: 0,00

Si el medio de pago es Tarjeta de Crédito, el Sistema debe informar al sistema “Tarjeting Veriphone”. Además el sistema:

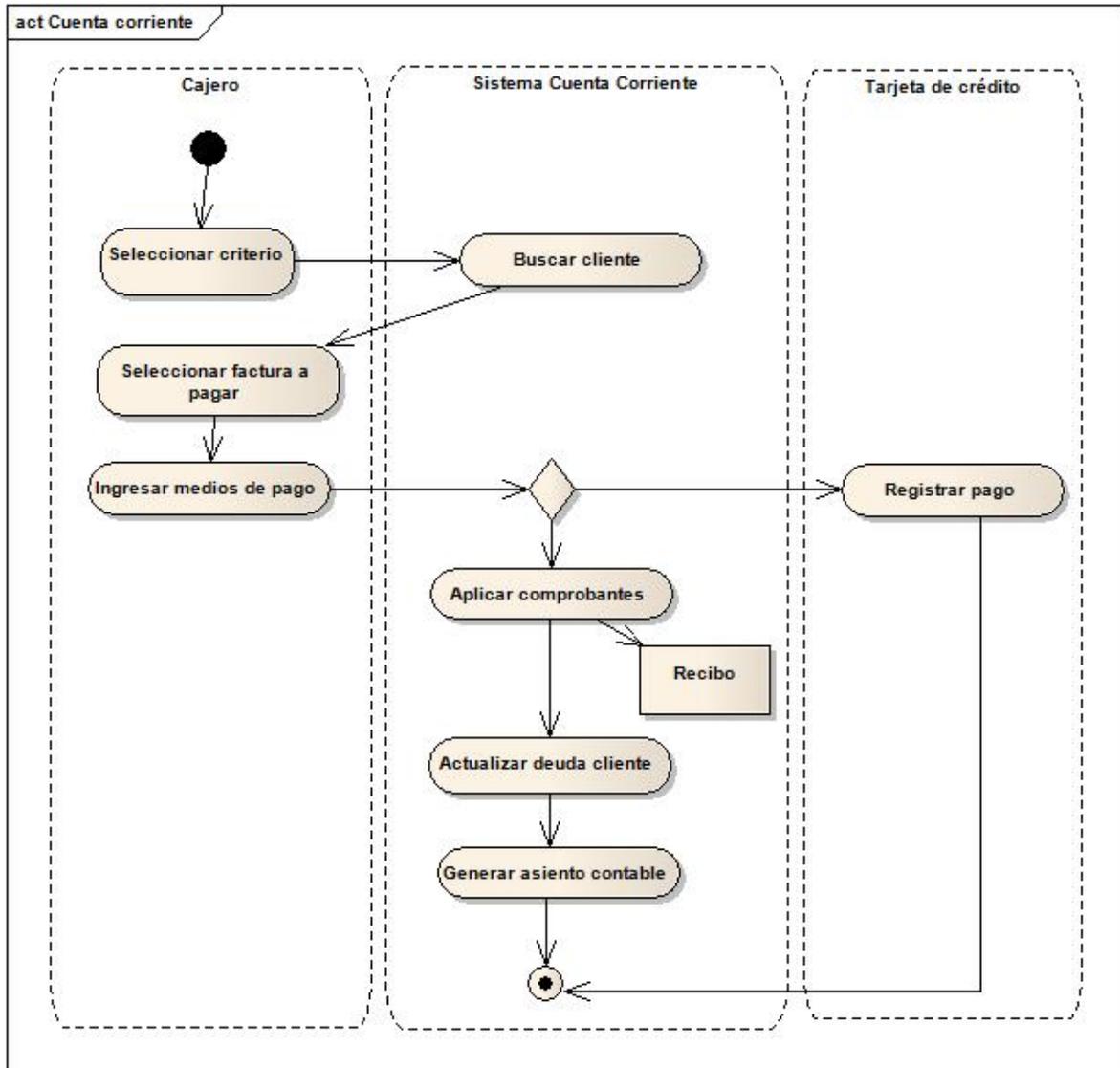
- Aplicará los comprobantes pagados (actualizará los montos pendientes de cada comprobante) generando el recibo correspondiente
- Actualizará la deuda del cliente
- Notificará a contabilidad para generar el asiento que refleje en la contabilidad el pago.

Para generar el diagrama de actividades debemos:

- Encontrar los actores de este caso de uso y qué responsabilidades cumplen

- Identificar las actividades/acciones que componen el caso de uso, si hay concurrencia (forks), decisiones (ifs) y cuál es el orden que sigue cada una de las acciones (flujos de control).
- Identificar si alguna de las acciones recibe o produce como resultado un objeto. Posiblemente todas las acciones generan algún tipo de objeto (un recibo, el criterio por el cual voy a seleccionar la cuenta corriente, un asiento contable, etc.) pero *sólo voy a mostrar los objetos que sea importante mostrar*.

Una posible solución:



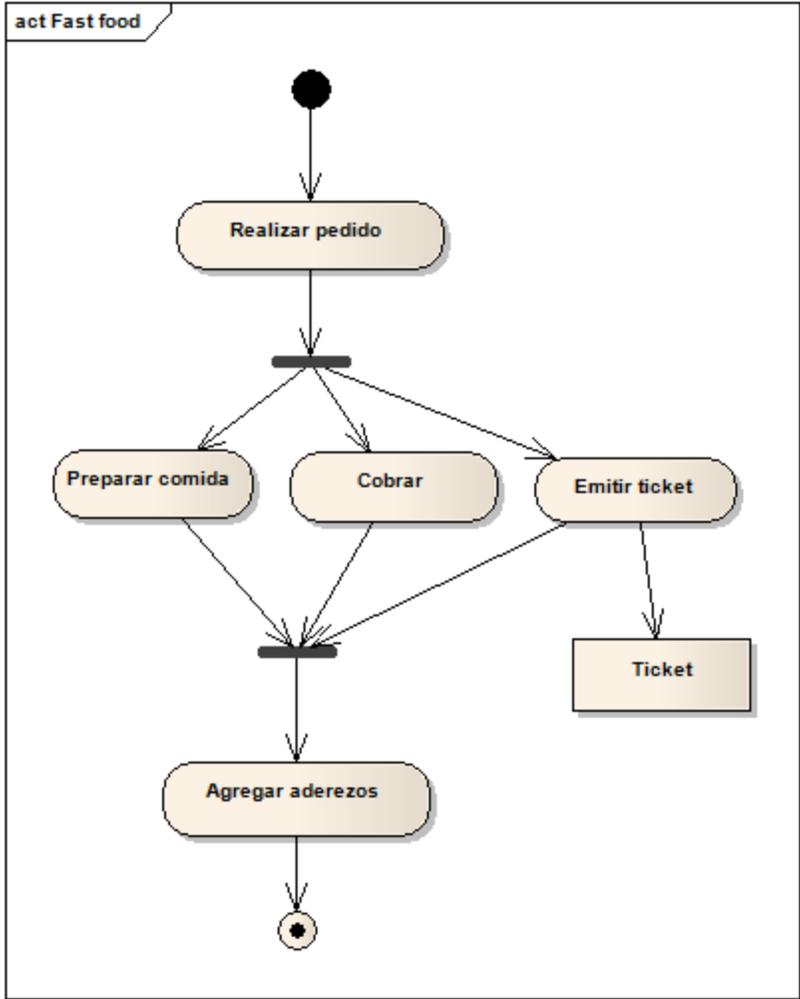
## 2.2 “Fast-food” (conurrencia)



Como veremos a continuación el diagrama de actividades también resulta útil para mostrar transiciones simultáneas que se dan en un caso de uso. Pensemos cómo es la operación de un local de comida rápida:

- Hacemos el pedido al cajero en base a las 213 promociones posibles de hamburguesas con papas fritas
- El cajero hace el pedido por altavoz, o bien se lanza en una carrera meteórica para llenar la gaseosa, recibir la hamburguesa y capturar las papas fritas, emitir el ticket y al mismo tiempo, cobrarnos (¡eso es lo que se dice conurrencia!)
- Por último, nos pregunta si queremos agregar aderezos (mostaza, mayonesa, ketchup, sal) y nos da la bandeja con sonrisa angelical.

Tomaremos como una acción atómica la preparación la comida. Lo que nos interesa es cómo mostrar las actividades que se superponen en el tiempo:



## 3 Resumen

**¿Quién hace el diagrama de actividad?** El analista funcional.

**¿Para quién es el diagrama de actividad?** Puede servir para el usuario final, ya que no estamos explicando técnicamente cómo resolver cada problema, y es útil para mostrar los actores que intervienen en cada caso de uso y su responsabilidad.

Al programador le sirve como visión general de un proceso, como explicación funcional del código que va a desarrollar, pero no mucho más que eso.

Este diagrama tiene reminiscencias del cursograma que se utiliza para definir circuitos administrativos, donde se mostraba el flujo de documentos que compartían los diferentes departamentos/actores externos de un circuito en una organización.

**¿Cuándo queremos hacer un diagrama de actividad?** Cuando tengamos ganas de analizar un proceso (para entenderlo o para mejorarlo), sin necesidad de bajar tanto a detalle cómo implementarlo.