

MÁQUINAS DE ESTADOS EN UML

Una máquina de estado puede analizarse desde dos puntos de vistas a través de distintos diagramas de UML, el diagrama de actividades y el diagrama de clases. A continuación se detallarán elementos, aplicaciones y ejemplos de estos diagramas.



MÁQUINAS DE ESTADOS EN UML

DEFINICIONES

Una máquina de estados define el comportamiento de un objeto, que puede ser cualquier elemento de modelado, por ejemplo la instancia de una clase, un caso de uso o un sistema completo.

Este comportamiento se detalla enumerando la secuencia de **estados** por los que pasa el objeto que se analiza durante su vida útil, detallando **eventos** que puede recibir y las **acciones** que se realizarán como respuesta a los eventos recibidos.

Un estado es una situación en la cual el objeto de análisis se encuentra realizando alguna operación o esperando que se presente algún evento.

Un evento es un estímulo externo que hace que el objeto de análisis reaccione de alguna manera. Puede ser la lectura de un sensor, la acción de otro sistema o simplemente el paso del tiempo.

Cuando el objeto recibe un evento puede reaccionar ejecutando alguna operación, cambiando de estado o bien realizando ambas: ejecuta una operación y cambia de estado.

Una transición es la relación entre dos estados. Indica que el objeto estando en un estado “A” pasa al estado “B” en respuesta a un evento.

Una actividad es una operación no atómica que se realizará dentro de un estado.

Una acción es una operación atómica que se realiza dentro de un estado.

Una máquina de estados puede analizarse desde dos puntos de vistas:

- Destacando la secuencia de estados, eventos y transiciones a través de un **Diagrama de Estado**
- Destacando el flujo de actividades a través de un **Diagrama de Actividades**.

DIAGRAMA DE ESTADOS

El diagrama de estados es uno de los diagramas de UML que permite modelar los aspectos dinámicos del sistema.

Un diagrama de estados es una **máquina de estados** donde se destaca el flujo de control entre los estados por los que pasa un objeto durante su vida útil.

APLICACIÓN

Este diagrama se puede utilizar para modelar cualquier elemento de UML, normalmente se utilizará en:

- Instancias de clases: “Objetos”.
- Casos de Uso.
- Sistema completo.
- Subsistemas.
- Algoritmos.
- Sistemas de tiempo real.
- Diseño de interfaz de usuario (Menú).

ELEMENTOS

Estado: es una situación en la que el objeto se encuentra realizando alguna actividad o esperando que se presente algún evento.

Se representa con un rectángulo con puntas redondeadas, con su nombre en el interior.

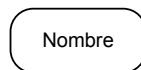


Figura 1: Estado.

Estado inicial: es un tipo especial de estado. Representa el estado en el que se sitúa el objeto al comienzo del análisis.

Se representa con un círculo negro.



Figura 2: Estado Inicial.

Estado final: es un tipo especial de estado. Representa que la ejecución de la máquina de estados ha finalizado

Se representa con un círculo blanco que tiene en su interior un círculo negro de menor tamaño.



Figura 3: Estado Final.

Transición: es la relación entre dos estados. Se representa con una línea dirigida desde el estado original hacia el estado destino. Si el estado origen es igual al estado destino se la denomina: **auto transición**.

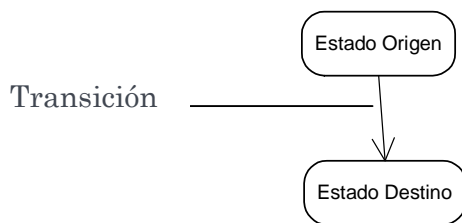


Figura 5: Transición.

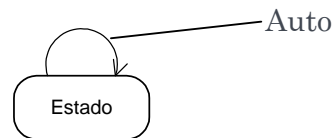


Figura 4: Auto transición.

Una transición está formada por varias partes:

- Estado Origen: estado en el que se encuentra el objeto antes de la transición.
- Evento: estímulo que provoca que objeto reaccione.
- Condición de guarda: es un requisito que se evalúa al momento de captarse el evento. (Expresión Booleana) Si se cumple con el requisito se dispara la transición (True), si no se cumple (False) no se dispara y el objeto no cambia de estado.
- Efecto: comportamiento que se ejecuta al dispararse la transición.
- Estado Destino: estado en el que queda el objeto luego de dispararse la transición.

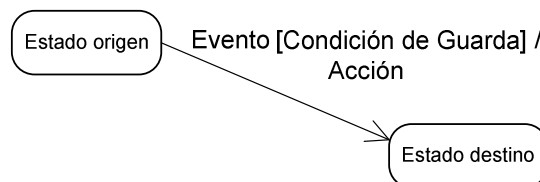


Figura 6: Evento, Condición de Guarda, Acción.

Veamos un ejemplo de diagrama de estados. Es este caso es el modelo de un sistema completo: un semáforo. Este semáforo va pasando del estado ALTA, PRECAUCIÓN a SIGA según un período de tiempo fijo. Con cada cambio de estado se van prendiendo o apagando luces según corresponda.

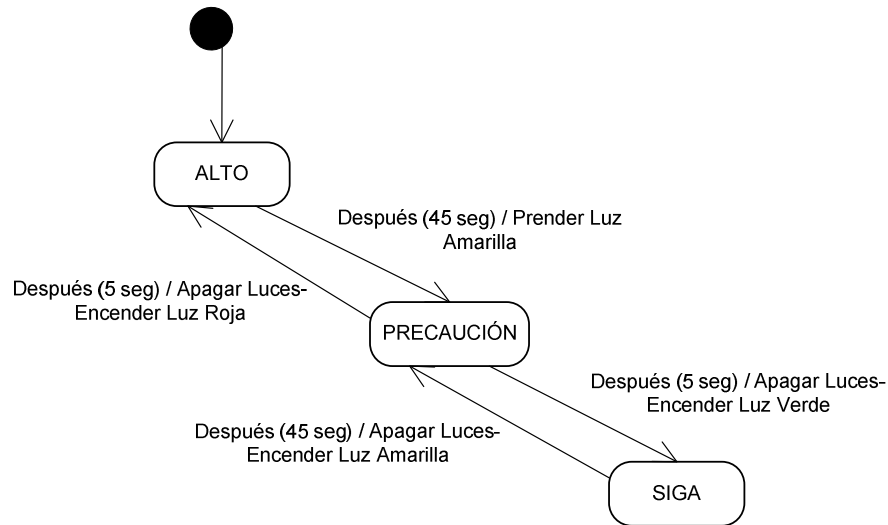


Figura 7: Diagrama de estados de un semáforo, primer punto de vista.

SUB ESTADOS

Un subestado es un estado ubicado dentro de otro. Permite estudiar el comportamiento del objeto cuando se encuentra en el estado principal.

Un estado simple es aquel que no posee una sub-estructura en su interior.

Un estado compuesto es un estado que posee subestados en su interior.

Sobre el ejemplo anterior, podríamos agregar que el semáforo a las 10 PM deja de funcionar, mostrando desde ese momento la luz amarilla en forma intermitente hasta las 6 AM donde el semáforo se enciende y vuelve a trabajar tal como se detalló en la figura 7.

Máquinas de Estados en UML. Diagrama de Estados.

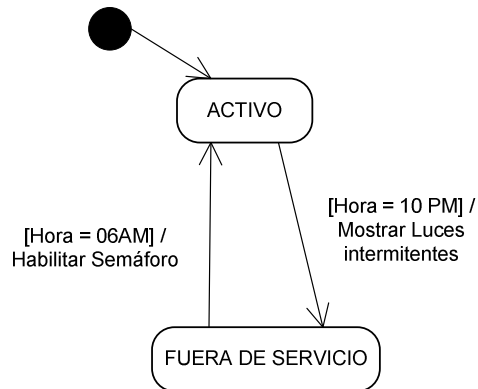


Figura 8: Diagrama de Estados de un semáforo, segundo punto de vista.

Utilizando el concepto de Subestados podríamos armar un diagrama como el de la figura 9, donde se puede ver en un solo diagrama el funcionamiento completo del semáforo.

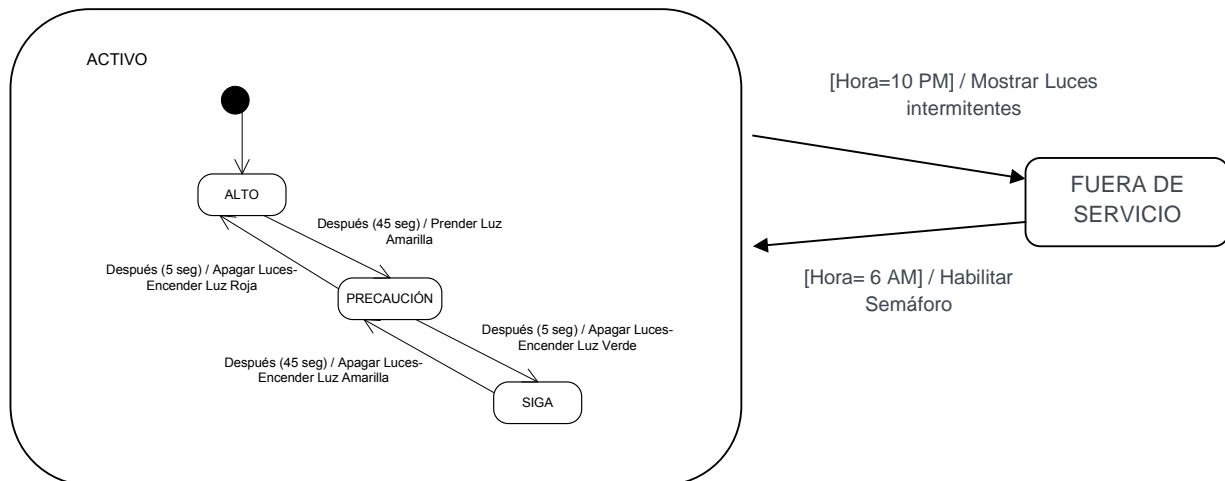


Figura 9: Diagrama de Estados de un semáforo con subestados.

DIAGRAMA DE ACTIVIDADES

Un diagrama de actividades es uno de los diagramas de UML que permite describir el comportamiento dinámico del sistema.

En este caso se destacará el flujo de control entre las actividades. En este diagrama se tiene en cuenta, el orden de ejecución de las actividades, actividades que pueden ser llevadas adelante en forma paralela y no solamente se modelarán aspectos del sistema sino también en este diagrama se podrán observar las tareas que serán realizadas por las personas y/o usuarios involucrados.

APLICACIÓN

Este diagrama, al igual que el diagrama de estados, se puede utilizar para modelar cualquier elemento de modelado de UML, normalmente se utilizará en:

- Sistema completo.
- Subsistemas.
- Casos de Uso.
- Una operación.
- Flujo de Trabajo (Workflow).
- Modelo de Negocio.
- Ingeniería directa (creación de código en lenguaje de programación a partir de un diagrama).
- Ingeniería inversa (creación del diagrama a partir de código escrito en algún lenguaje de programación).

ELEMENTOS

- **Nodo inicial:** indica el comienzo del flujo de actividades. Se representa con un círculo de color negro.



Figura 10: Nodo Inicial

- **Nodo final:** indica el fin del flujo de actividades. Se representa con un círculo blanco que tiene en su interior un círculo negro.



Figura 11: Nodo Final

- **Actividad:** es una acción o bien un conjunto de operaciones que pueden ser ejecutadas en forma automatizada o manualmente.

Se representan con un rectángulo de puntas redondeadas con su nombre en el interior.

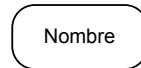


Figura 12: Actividad.

Se pueden reconocer dos tipos de actividades

- **Acción:** es una operación simple que no puede descomponerse. Es atómicas.

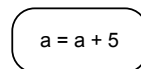


Figura 13: Acción.

- **Nodo de actividad:** es una agrupación de acciones.



Figura 14: Nodo de Actividad.

- **Transición:** es una relación entre dos actividades. Se representa con una flecha que se dirige desde la actividad predecesora hasta la sucesora. No se debe indicar ningún nombre, evento ni etiqueta.

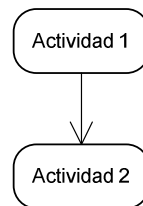


Figura 15: Transición

Cuando una actividad termina, el flujo de control continúa en la siguiente actividad indicada por la transición.

- **Bifurcación:** se utiliza para definir caminos alternativos. Se representa con un rombo. Pueden tener un flujo de entrada y dos o más de salida. En cada flujo debe incorporarse una expresión booleana que se evaluará al ingresar a la bifurcación. En este caso la bifurcación se conoce como: “Decisión”.

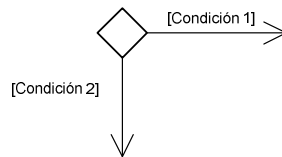


Figura 16: Bifurcación

Cuando dos caminos vuelven a unirse también puede utilizarse una bifurcación con dos entradas y una salida. En este caso la bifurcación se conoce como: “Fusión”.

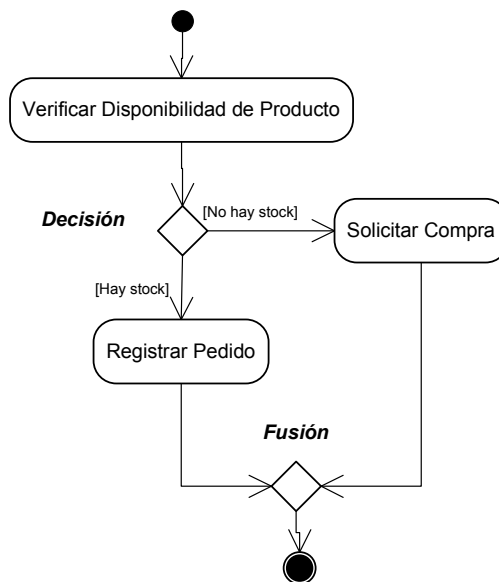


Figura 17: Ejemplo de Bifurcación. Decisión. Fusión.

- **Barra de sincronización:** describe la coordinación entre actividades. (Stevens & Pooley, 2007). Se utiliza para mostrar las divisiones y uniones de los distintos flujos de control que pueden ejecutarse en forma paralela.

Se representa con una barra gruesa horizontal.

Figura 18: Barra de sincronización

Una división representa la separación de un flujo de control simple en dos o más flujos paralelos. Luego de la división las actividades de cada flujo continúan su ejecución por separado respetando el orden de las transiciones de cada flujo hasta que se vuelven a encontrar en un solo camino: la unión.

La unión representa la sincronización de dos o más flujos de control.

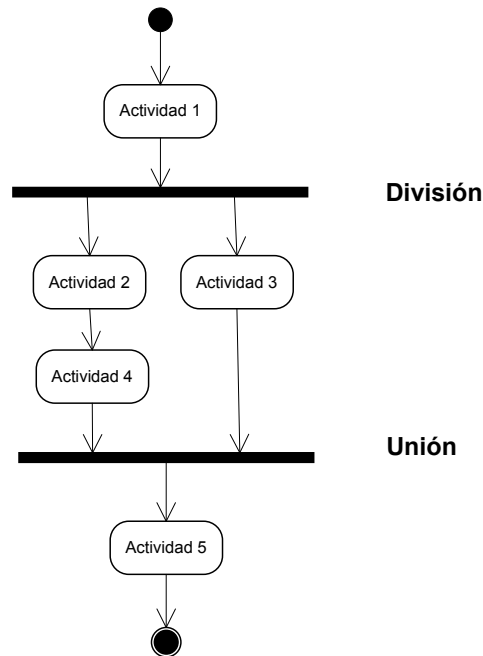


Figura 19: Ejemplo Barra de Sincronización. División. Unión.

A continuación se mostrará un ejemplo completo de Diagrama de Actividades con todos los elementos que se han detallado. En este caso se representarán las actividades básicas necesarias para enviar un mensaje de texto desde un celular.

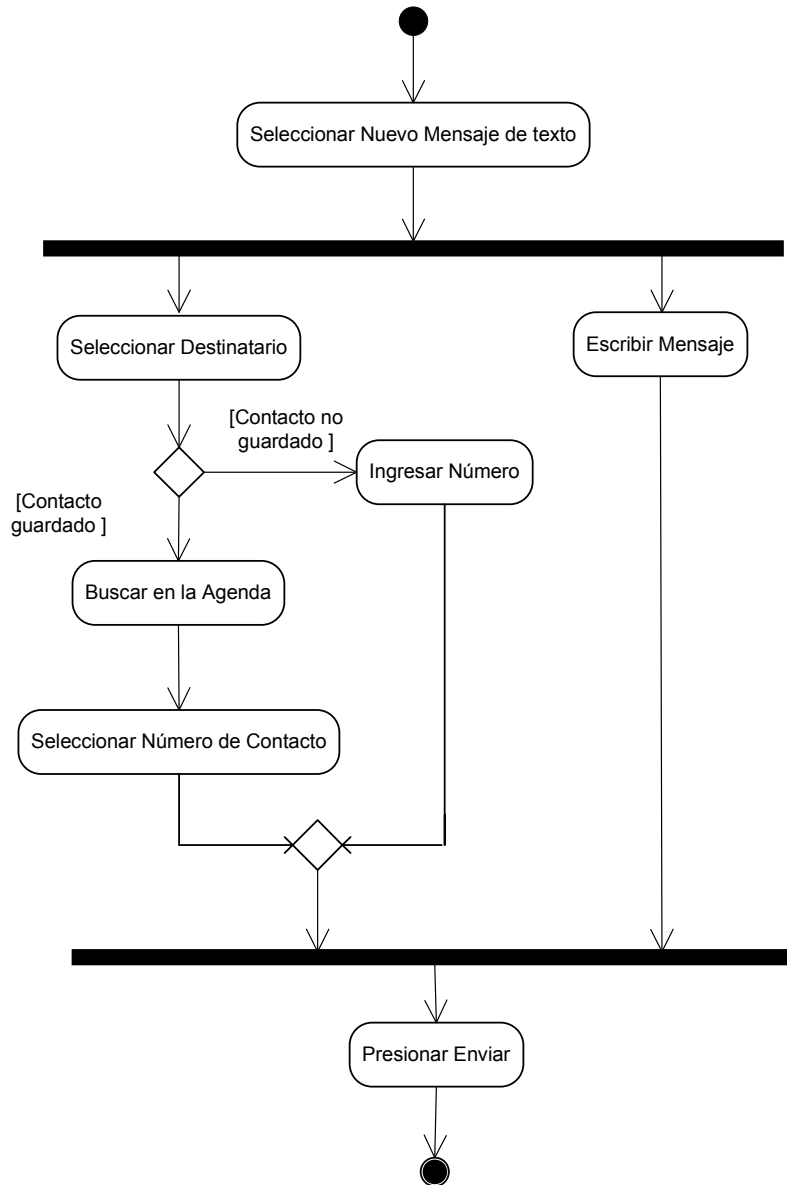


Figura 20: Diagrama de Actividad. Enviar Mensaje de Texto.

CALLES

También conocidas como carriles andariveles o swimlanes. Consiste en agrupar las actividades en grupos, donde cada grupo representa el responsable de ejecutar la o las actividades asociadas. Los grupos se separan por líneas verticales. Cada calle tiene un nombre único y cada actividad debe pertenecer a una única calle.

Son especialmente utilizadas para representar flujos de trabajos (Workflow)

Veamos un ejemplo de Diagrama de Actividades con “Calles”. En este diagrama se analizarán las acciones realizadas durante el préstamo de un libro en la biblioteca de una universidad.

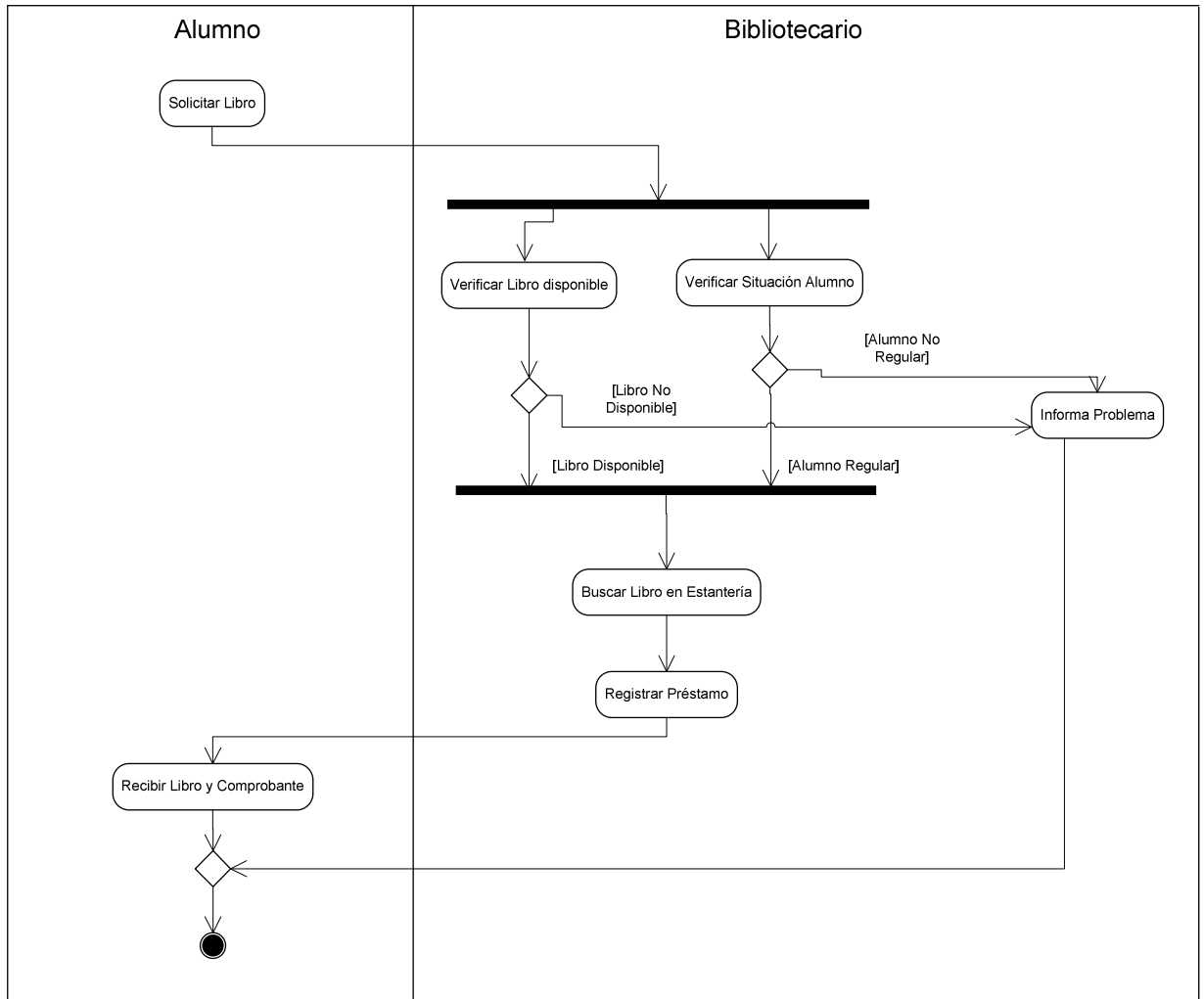


Figura 21: Diagrama de Actividades con Calles.

A continuación un ejemplo adicional con el uso de Calles, en este caso modelando un flujo de trabajo, un circuito de ventas que parte desde la solicitud de un cliente hasta la entrega del pedido.

Máquinas de Estados en UML.
Diagrama de Actividades.

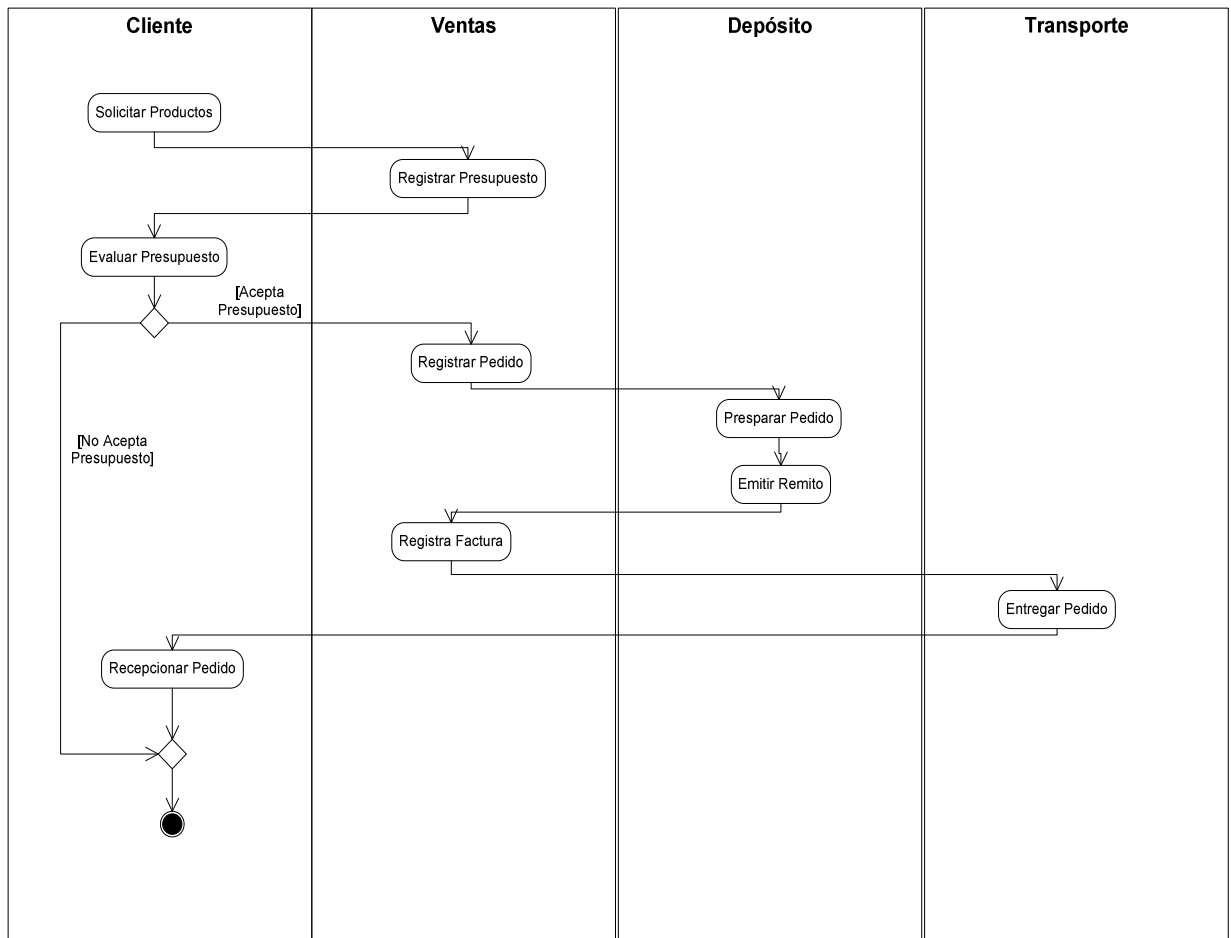


Figura 22: Ejemplo de Diagrama de Actividades con Calles. Flujo de Trabajo.

BIBLIOGRAFÍA

Booch, G., Rumbaugh, J., & Jacobson, I. (2006). *El Lenguaje Unificado de Modelado Guía del Usuario. Segunda Edición*. Addison - Wesley.

Fowler, M. (1999). *UML gota a gota*. México: Addison Wesley.

Kendall, K. E., & Kendall, J. E. (2011). *Análisis y Diseño de Sistemas*. México: Pearson Educación.

Pfleeger, S. L. (2002). *Ingeniería de Software. Teoría y Práctica*. Buenos Aires, Argentina.: Prentice Hall.

Pressman, R. S. (2006). *Ingeniería del Software. Un enfoque práctico*. México: Mc Graw Hill.

Sommerville, I. (2005). *Ingeniería del Software*. Madrid: Pearson. Addison Wesley.

Stevens, P., & Pooley, R. (2007). *Utilización de UML en Ingeniería del Software con Objetos y Componentes*. España: Pearson Addison Wesley.