



Usuario

## UA6. Elaboración de diagramas de comportamiento

Entornos de Desarrollo – 1ºDAM

Luis del Moral Martínez

versión 21.03

Bajo licencia CC BY-NC-SA 4.0



# Contenidos del tema

## 1. Introducción

- 1.1 Mapa conceptual del tema
- 1.2 Comportamiento dinámico del sistema
- 1.3 Algunos conceptos importantes

## 2. Diagramas de casos de uso

- 2.1 Elementos del diagrama
- 2.2 Asociaciones y relaciones

## 3. Diagramas de secuencia

- 3.1 Elementos del diagrama
- 3.2 Línea de vida de un objeto
- 3.3 Envío de mensajes

## 4. Diagramas de colaboración

- 4.1 Objetos y actores
- 4.2 Envío de mensajes

## 5. Diagramas de estados

- 5.1 Sucesos y acciones del sistema
- 5.2 Estado en reposo
- 5.3 Subestados

## 6. Diagramas de actividades

- 6.1 Decisiones
- 6.2 Concurrencia

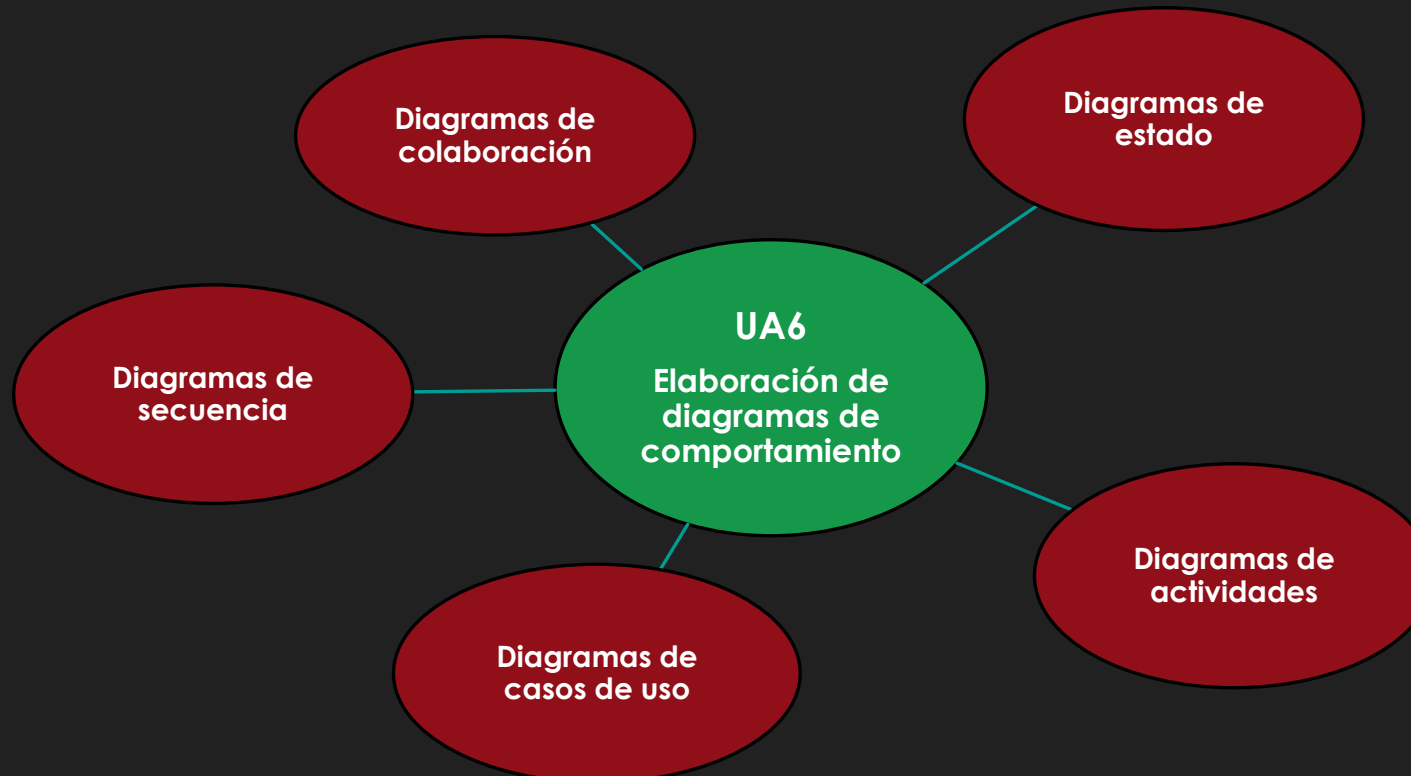
# Contenidos de la sección

## 1. Introducción

- 1.1 Mapa conceptual del tema
- 1.2 Comportamiento dinámico del sistema
- 1.3 Algunos conceptos importantes

# 1. Introducción

## 1.1 Mapa conceptual del tema



# 1. Introducción

## 1.2 Comportamiento dinámico del sistema

- En la UA anterior especificamos el **diseño estático** del sistema (diagrama de clases)
- El **diseño dinámico** se especifica con **diagramas de comportamiento**:
  - Diagrama de casos de uso
  - Diagrama de secuencia
  - Diagrama de colaboración
  - Diagrama de estados
  - Diagrama de actividades

# 1. Introducción

## 1.3 Algunos conceptos importantes

- **Actor:** persona o rol que interactúa con un sistema
- **Concurrencia:** consiste en que varios procesos se ejecuten a la vez (y puedan interactuar)
- **Mensaje:** cuando un objeto ejecuta un método de sí mismo u otro objeto (paso de mensaje)
- **Método recursivo:** método que realiza una llamada a sí mismo (más pesados que los iterativos)
- **Síncrono:** coordinado en el tiempo (hay que esperar a que finalice para continuar)
- **Standby:** en este estado se reduce el consumo y el sistema tarda más tiempo en responder

# Contenidos de la sección

## 2. Diagramas de casos de uso

- 2.1 Elementos del diagrama
- 2.2 Asociaciones y relaciones

## 2. Diagramas de casos de uso

### ¿Para qué sirven?

- Permiten especificar **cómo interactúa el usuario con el sistema**
- En estos diagramas se muestra la **interacción** entre el **sistema** y el **mundo exterior** (actores)



## 2. Diagramas de casos de uso

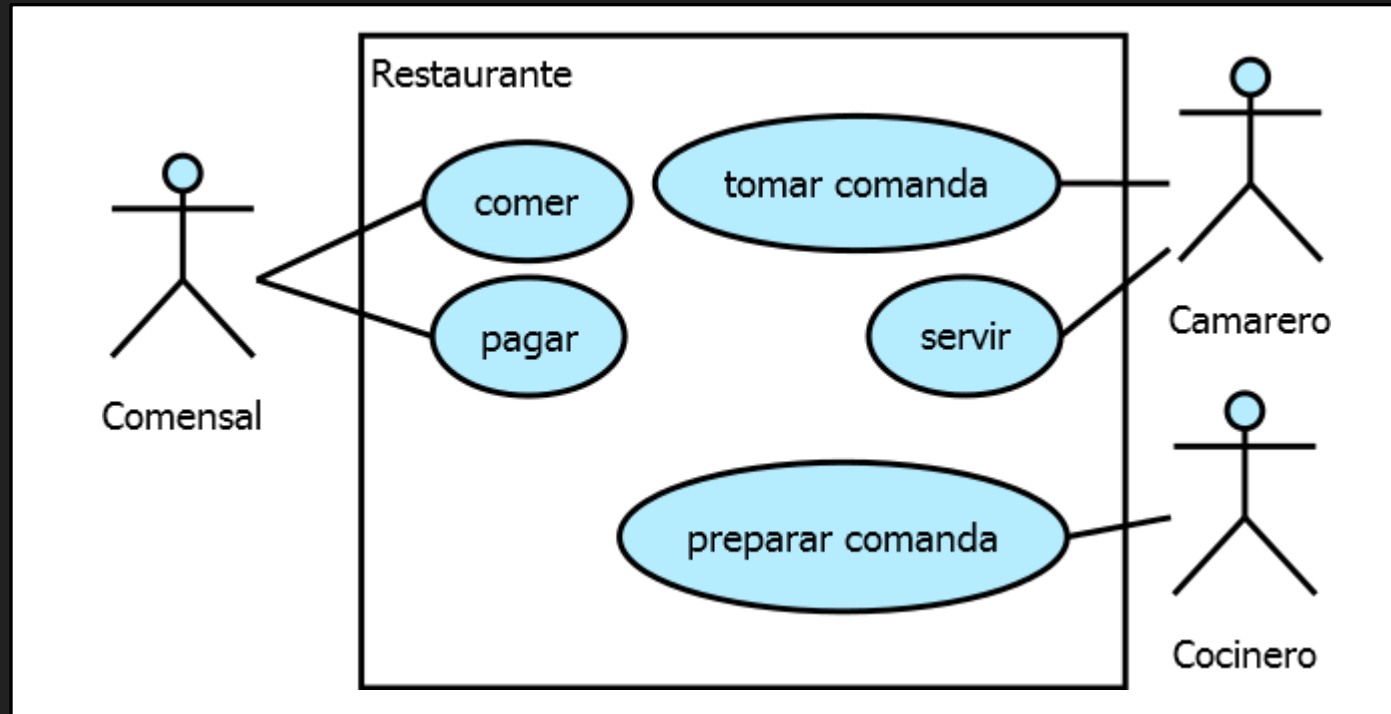
### 2.1 Elementos del diagrama

- **Sistema:** engloba la funcionalidad que ha sido codificada en la aplicación
- **Caso de uso:** es lo que hace el sistema, su funcionalidad
- **Actor:** es la persona o rol que usa el sistema (interactúa con uno o varios casos de uso)
- Veamos algunos **ejemplos**

## 2. Diagramas de casos de uso

### 2.1 Elementos del diagrama

- **Ejemplo 1:** restaurante



## 2. Diagramas de casos de uso

### 2.1 Elementos del diagrama

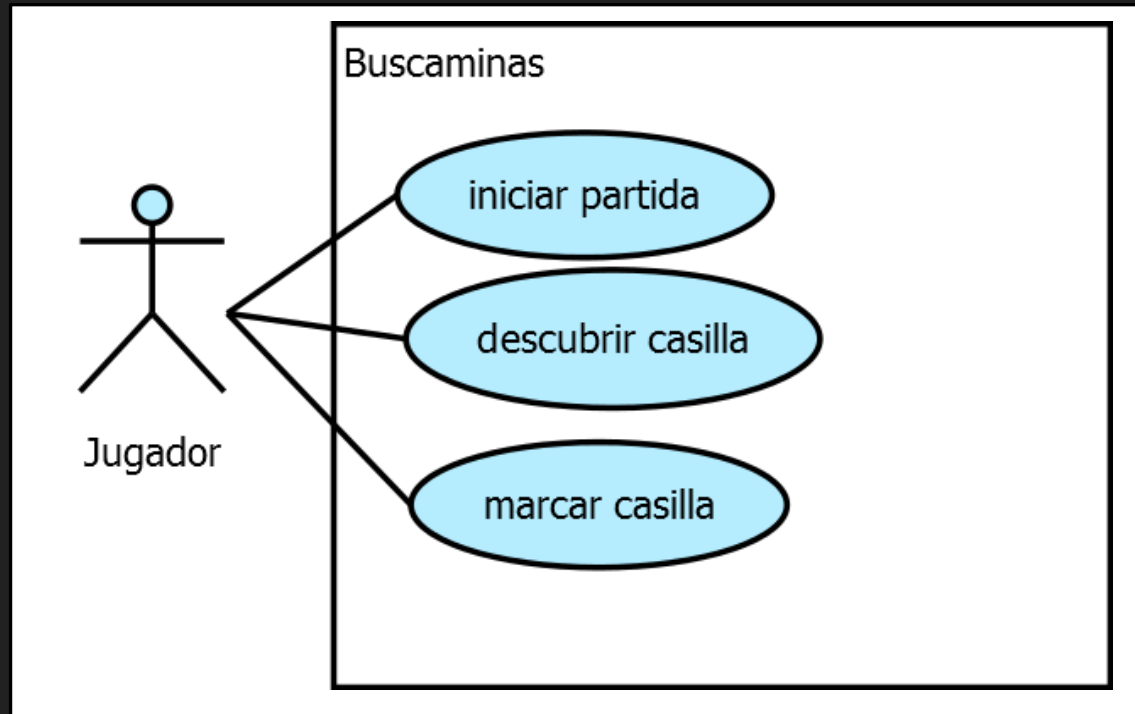
- **Ejemplo 2:** buscaminas



## 2. Diagramas de casos de uso

### 2.1 Elementos del diagrama

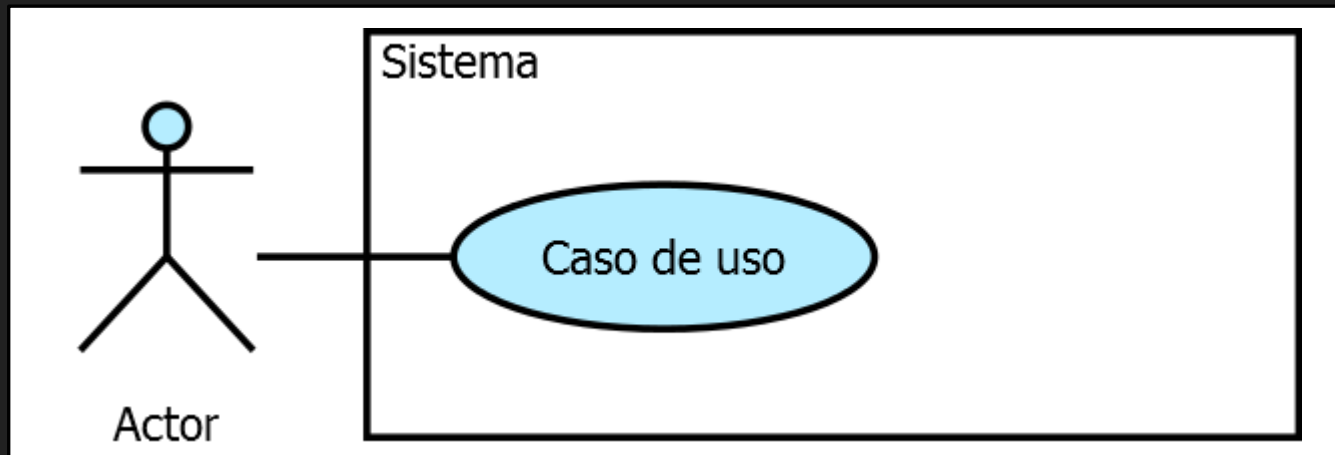
- **Ejemplo 2:** buscaminas



## 2. Diagramas de casos de uso

### 2.2 Asociaciones y relaciones

- Las asociaciones conectan el **actor** con los **casos de uso**
- Una asociación implica la **participación** de un **actor** en el sistema
- Las asociaciones son **obligatorias** porque representan la **interacción** del actor en el sistema

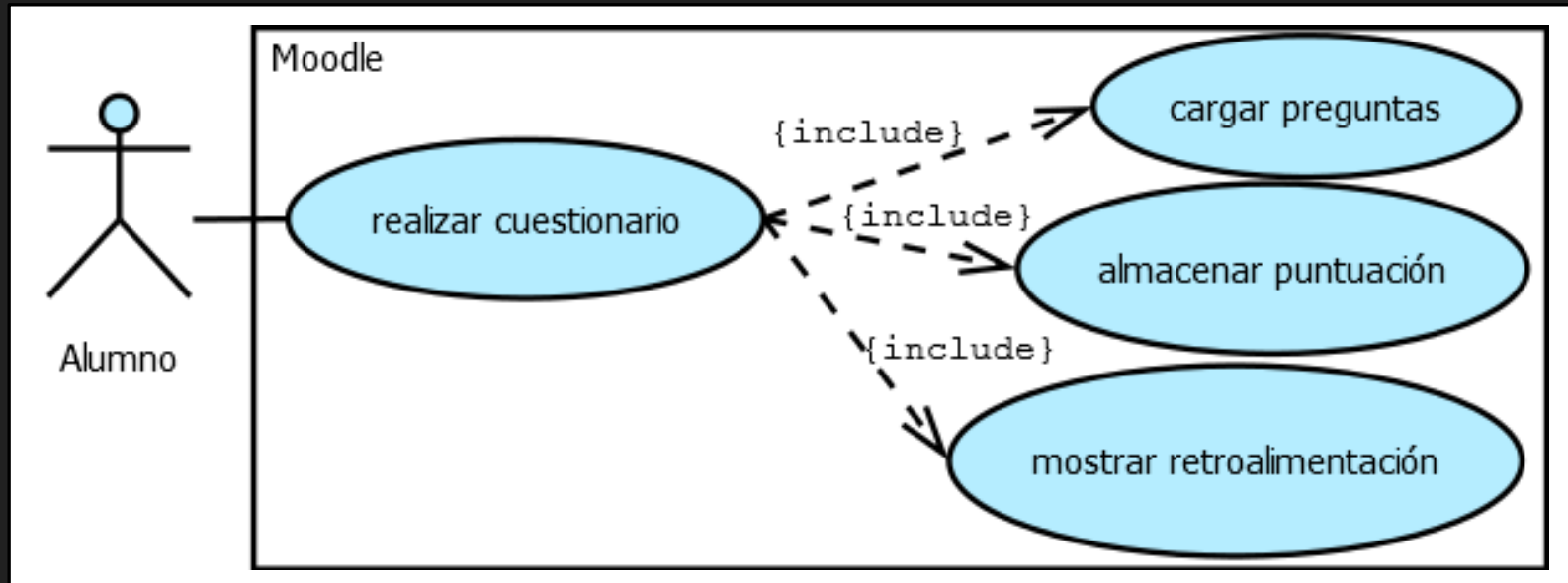


## 2. Diagramas de casos de uso

### 2.2 Asociaciones y relaciones

- En los casos de uso existen tres tipo de **relaciones**:

1. **Inclusión (include)**: se incluye la funcionalidad de un caso de uso en otro

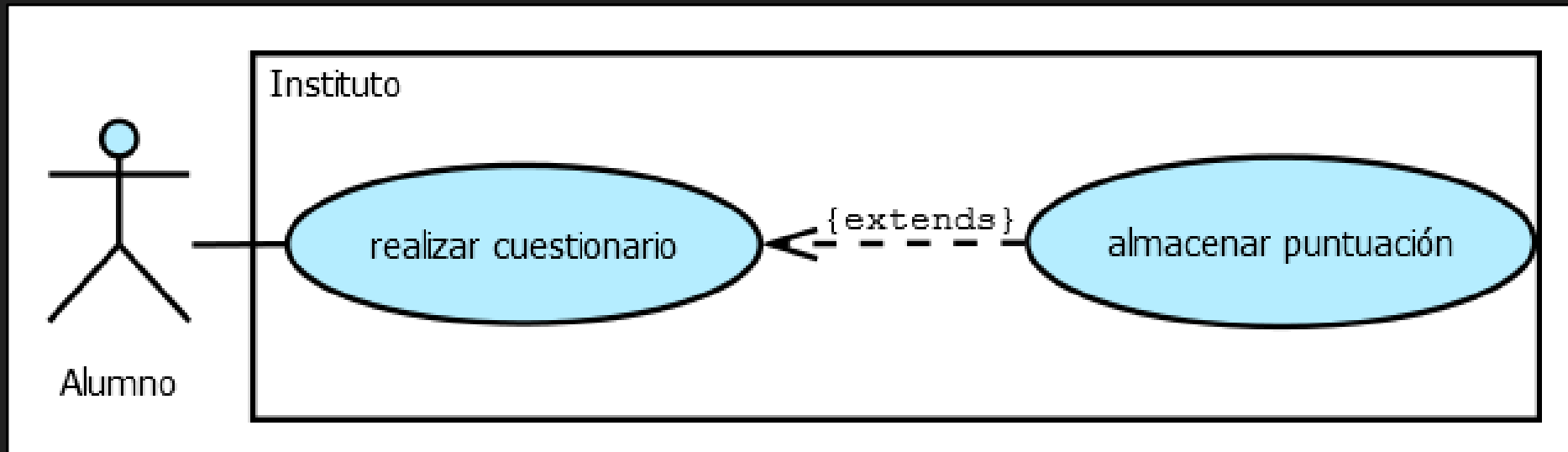


## 2. Diagramas de casos de uso

### 2.2 Asociaciones y relaciones

- En los casos de uso existen tres tipo de **relaciones**:

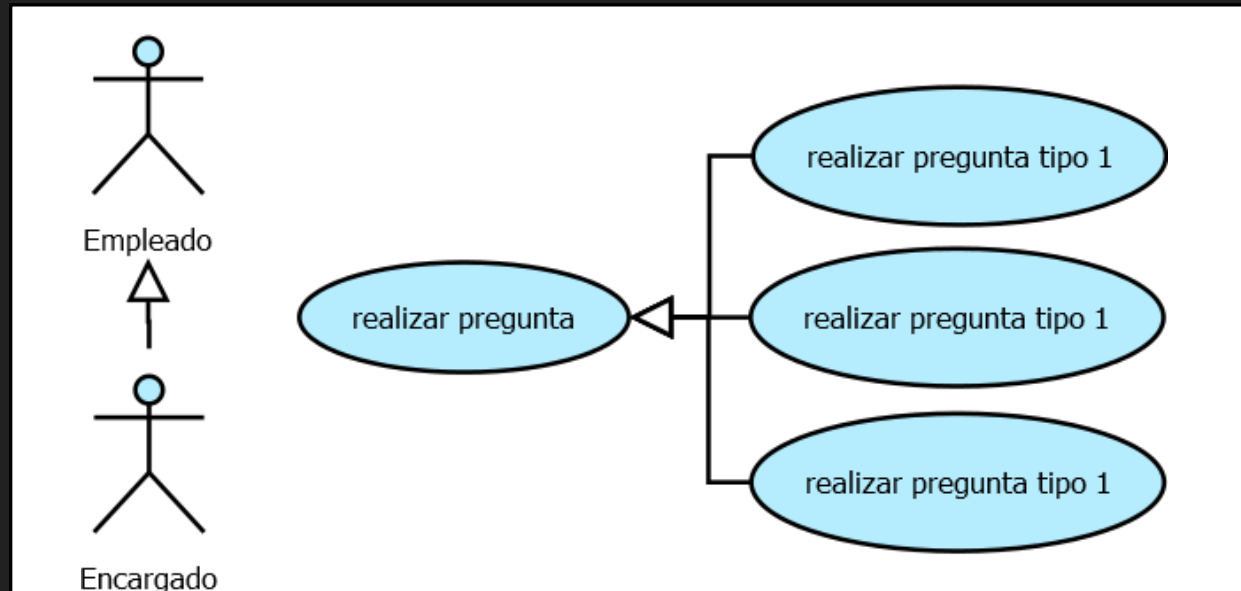
2. **Extensión (extends)**: variación de un caso de uso



## 2. Diagramas de casos de uso

### 2.2 Asociaciones y relaciones

- En los casos de uso existen tres tipo de **relaciones**:
  3. **Generalización**: permite identificar elementos comunes entre los distintos conceptos (actores o casos)





# Contenidos de la sección

## 3. Diagramas de secuencia

- 3.1 Elementos del diagrama
- 3.2 Línea de vida de un objeto
- 3.3 Envío de mensajes

# 3. Diagramas de secuencia

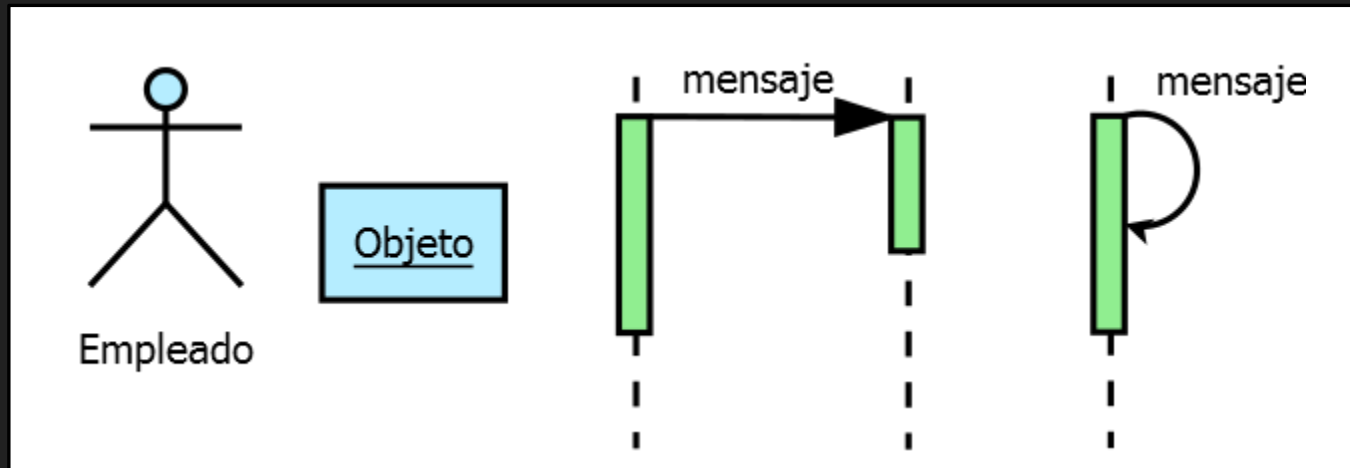
## ¿Para qué sirven?

- Permiten añadir la **variable tiempo** a las **interacciones** del sistema
- Estos diagramas permiten **secuenciar** y **temporizar** las interacciones en el sistema

# 3. Diagramas de secuencia

## 3.1 Elementos del diagrama

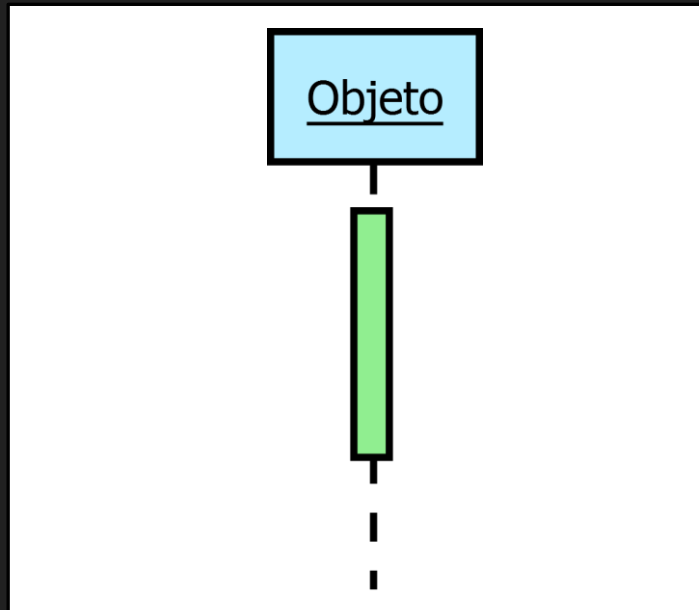
- **Objetos y actores:** son los que comienzan la secuencia de mensajes (origen)
- **Mensaje a otro objeto:** cuando un objeto interactúa con otro le pasa un mensaje
- **Mensaje al mismo objeto:** mensaje que se tiene como **origen** y **destino** el mismo objeto



# 3. Diagramas de secuencia

## 3.2 Línea de vida de un objeto

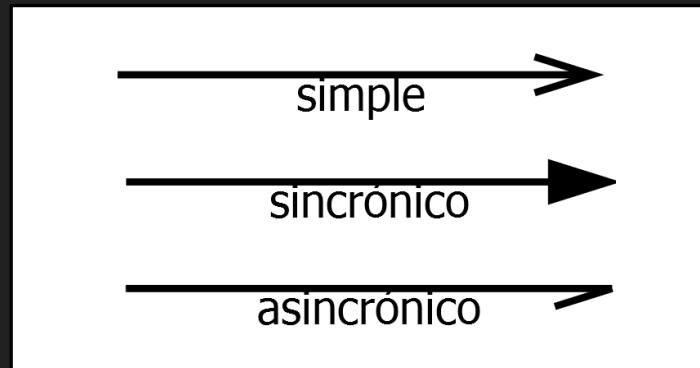
- Representa el **tiempo de vida** de un objeto
- A lo largo de su ciclo vital, el objeto **cambiará de estado** e **interactuará** con otros objetos



# 3. Diagramas de secuencia

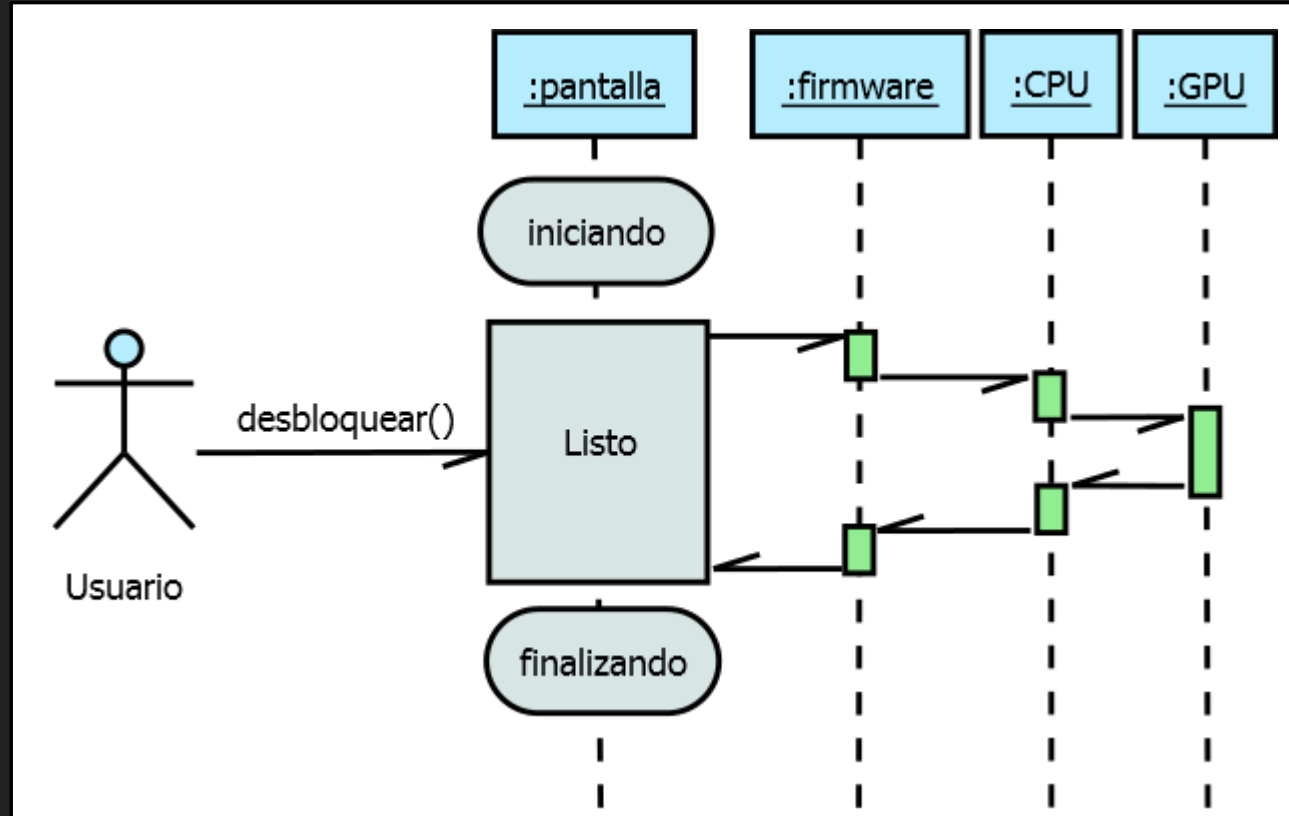
## 3.3 Envío de mensajes

- Los mensajes que intercambian los objetos pueden ser de tres tipos:
  - Simple:** pasan el hilo de ejecución de un objeto a otro (el código del objeto no se vuelve a ejecutar)
  - Sincrónicos:** esperan a que el método del otro objeto invocado termine para seguir la ejecución
  - Asincrónico:** no impiden que el método original siga su ejecución normal



# 3. Diagramas de secuencia

## Ejemplo de uso



# Contenidos de la sección

## 4. Diagramas de colaboración

- 4.1 Objetos y actores
- 4.2 Envío de mensajes

# 4. Diagramas de colaboración

## ¿Para qué sirven?

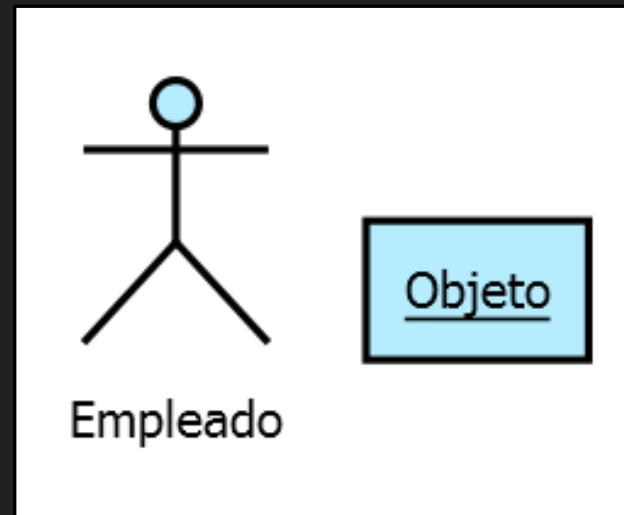
- Representan los **objetos**, las **interacciones**, el **orden** y los **mensajes enviados**
- Son similares a los **diagramas de actividad**, pero sin variable **temporal**



# 4. Diagramas de colaboración

## 4.1 Objetos y actores

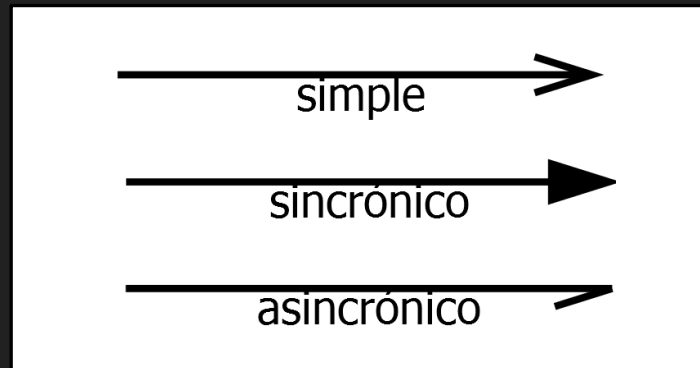
- **Objetos y actores:** son los que comienzan la secuencia de mensajes (origen)



# 4. Diagramas de colaboración

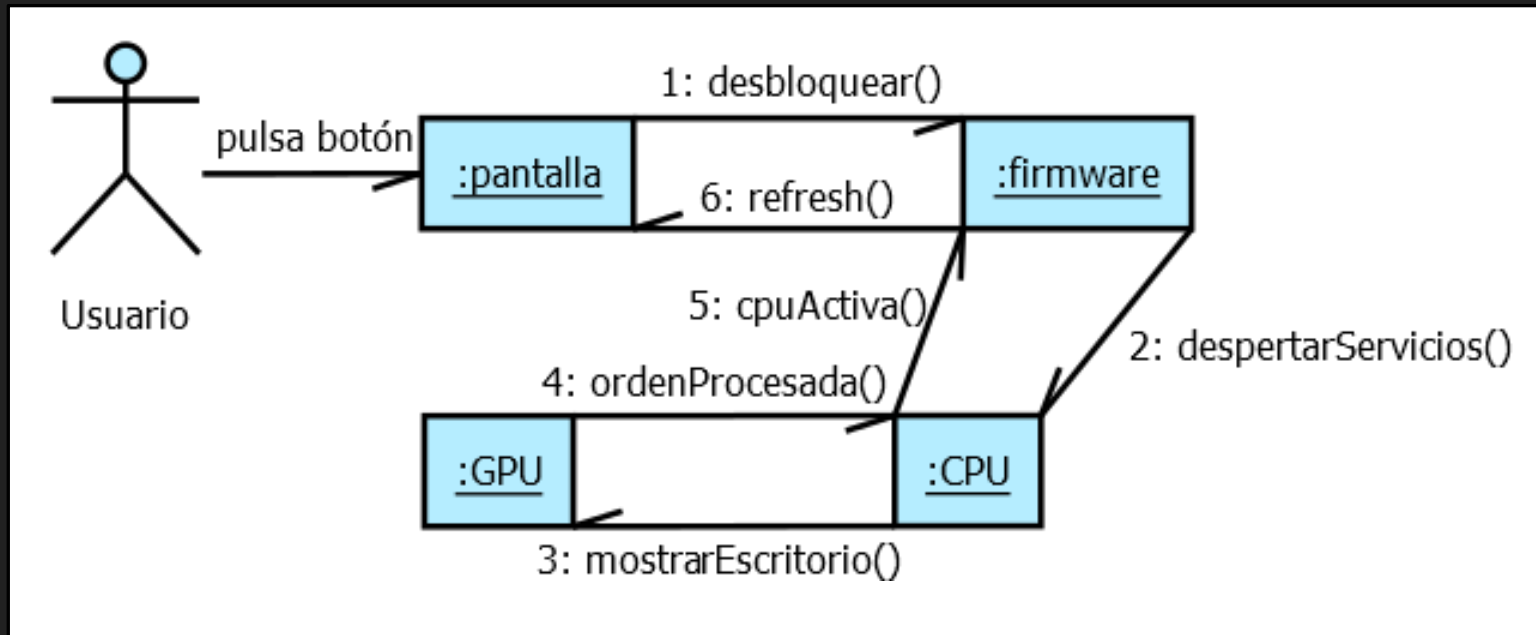
## 4.2 Envío de mensajes

- Los mensajes que intercambian los objetos pueden ser de tres tipos:
  - Simple:** pasan el hilo de ejecución de un objeto a otro (el código del objeto no se vuelve a ejecutar)
  - Sincrónicos:** esperan a que el método del otro objeto invocado termine para seguir la ejecución
  - Asincrónico:** no impiden que el método original siga su ejecución normal



# 4. Diagramas de colaboración

## Ejemplo de uso



# Contenidos de la sección

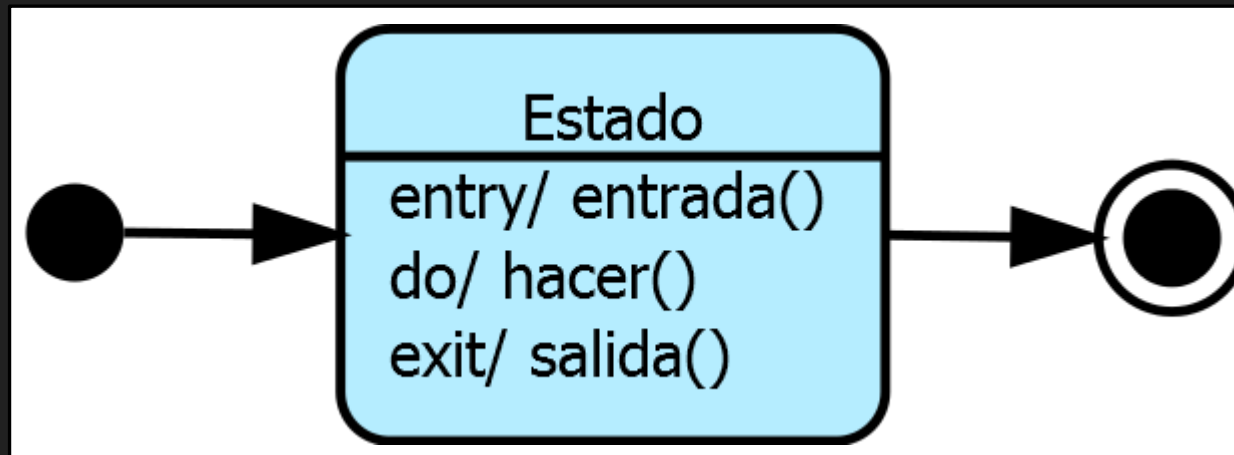
## 5. Diagramas de estados

- 5.1 Sucesos y acciones del sistema
- 5.2 Estado en reposo
- 5.3 Subestados

# 5. Diagramas de estados

## ¿Para qué sirven?

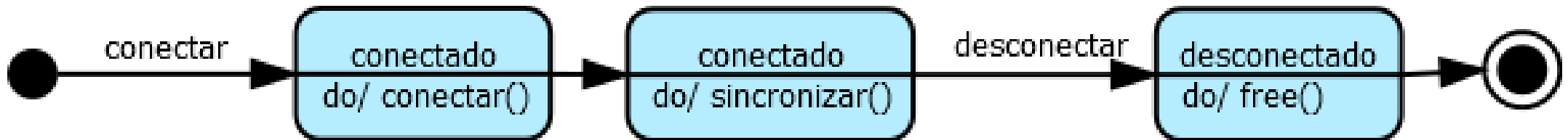
- Permiten representar los diferentes **eventos** que provocan un **cambio de estado** en el sistema
- Los iconos de estado recogen el nombre, las **variables de estado** y las **actividades**
- Existen tres actividades muy usadas: **entrada**, **salida** y **hacer**



# 5. Diagramas de estados

## 5.1 Sucesos y acciones del sistema

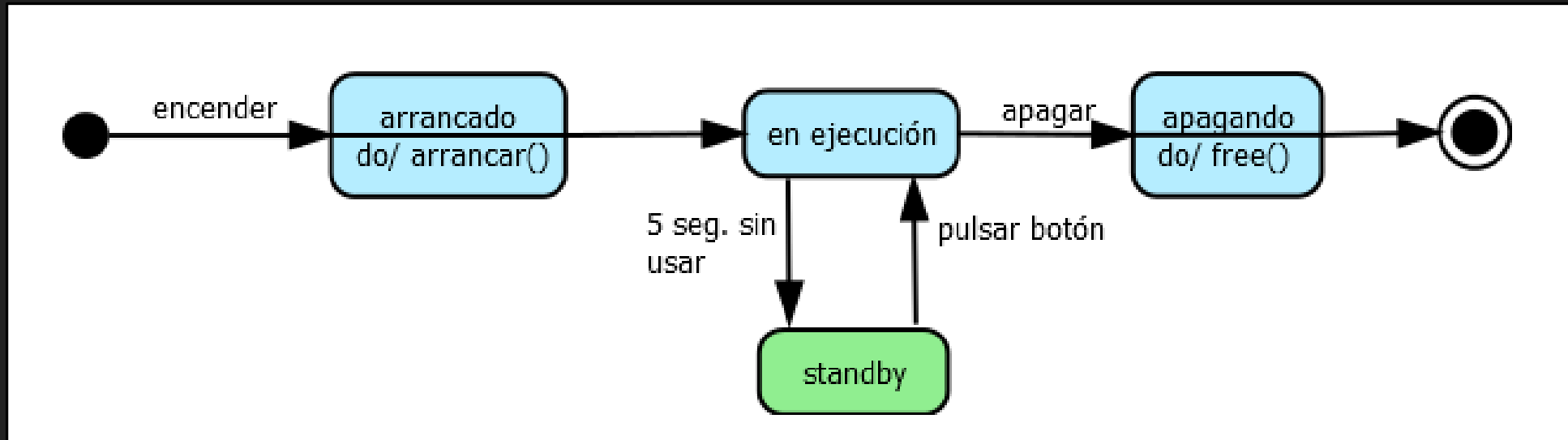
- Un **suceso** provoca que el estado del sistema cambie



# 5. Diagramas de estados

## 5.2 Estado en reposo

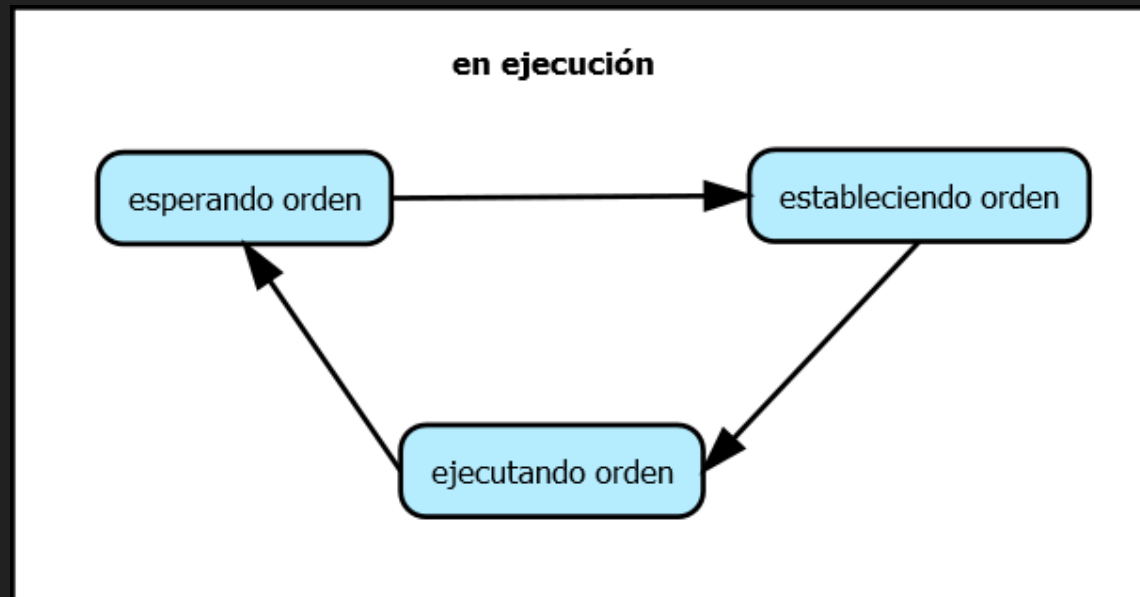
- El estado en reposo indica que el sistema está a la **espera** de que ocurra un **evento**



# 5. Diagramas de estados

## 5.3 Subestados

- A veces, un estado puede ser una **agrupación de subestados**
- Imaginemos el estado “en ejecución” del ejemplo anterior:





# Contenidos de la sección

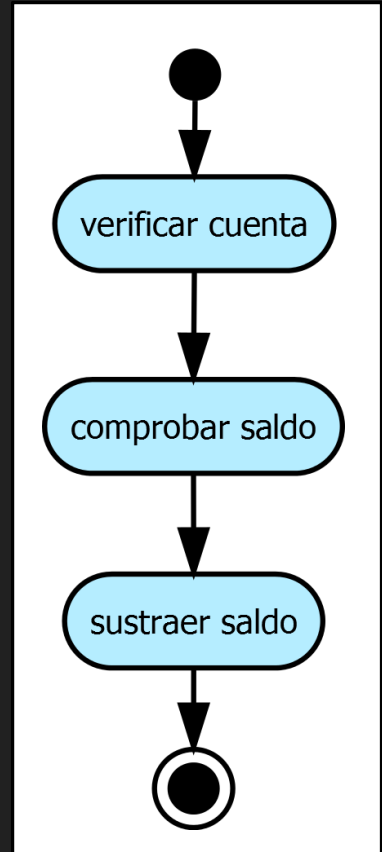
## 6. Diagramas de actividades

- 6.1 Decisiones
- 6.2 Concurrencia

# 6. Diagramas de actividades

## ¿Para qué sirven?

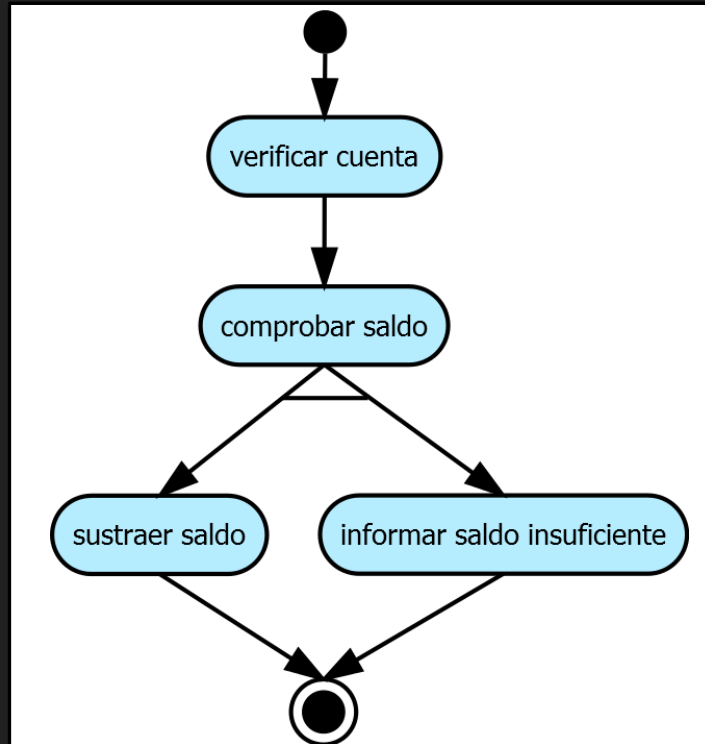
- Son parecidos a los **diagramas de flujo**
- Expresan el comportamiento de un proceso de forma esquemática



# 6. Diagramas de actividades

## 6.1 Decisiones

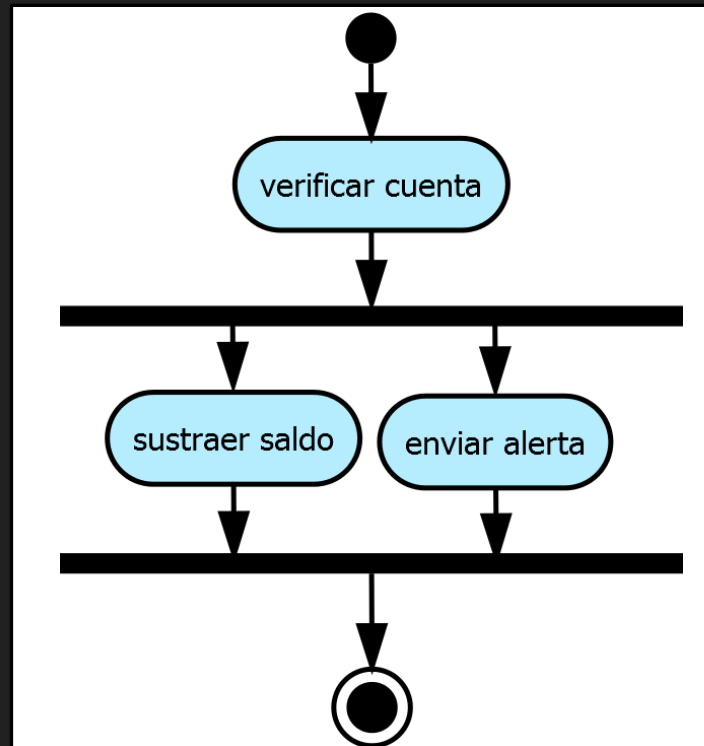
- Es posible representar la **toma de decisiones** en el diagrama



# 6. Diagramas de actividades

## 6.2 Concurrency

- La **concurrency** permite visualizar procesos que se deben sincronizar y se ejecutan a la vez



# Información complementaria

- **Curso completo diagramas UML (Playlist YouTube): [enlace](#)**
- **Diagramas de comportamiento (vídeo): [enlace](#)**
- **Descargar Dia Diagram Editor: [enlace](#)**
- **How to create UML diagram with Dia Diagram Editor: [enlace](#)**

# Créditos de las imágenes y figuras

## Diagramas, gráficas e imágenes

- Se han desarrollado en **PowerPoint** y se han incrustado en esta presentación
- Todos estos materiales se han desarrollado por el autor
- Para el resto de recursos se han especificado sus fabricantes, propietarios o enlaces
- Si no se especifica copyright, el recurso es de desarrollo propio