

# INGENIERÍA DE APLICACIONES

---

Modelado de Sistemas

Profesor : Omar Alejandro Bazar

# Temario

Modelado de Sistemas

Lenguaje de Modelado Unificado (UML)

Modelos de Contexto

Modelos de Interacción

- Modelos de casos
- Diagramas de secuencia

Modelos Estructurales

Modelos de Comportamiento

# Modelado de Sistemas

Es el proceso de desarrollo de **modelos abstractos** de un sistema.

Cada modelo presenta una **vista** o **perspectiva** diferente de ese sistema.

Por lo general, consiste en representar el sistema utilizando algún tipo de notación gráfica, que casi siempre se basa en anotaciones en el **Lenguaje de modelado unificado (UML)**.

También es posible desarrollar modelos formales (matemáticos) de un sistema, como una especificación detallada del sistema.

# Modelado de Sistemas

Los modelos se utilizan:

- durante el proceso de ingeniería de **requisitos** para **ayudar a derivar los requisitos** de un sistema,
- durante el proceso de **diseño** para **describir el sistema** a los ingenieros que implementan el sistema
- y **después de la implementación** para **documentar** la estructura y operación del sistema.

# Modelado de Sistemas

- Un modelo es una **abstracción** del sistema que se estudia, en lugar de una representación alternativa de ese sistema.
- Idealmente, una representación de un sistema debe mantener toda la información sobre la entidad que se representa.
- Una abstracción simplifica y selecciona deliberadamente las características más sobresalientes.

# Modelado de Sistemas

Se pueden desarrollar diferentes modelos para representar el sistema desde diferentes perspectivas. Por ejemplo:

- **Una perspectiva externa**, donde se modela el contexto o el entorno del sistema.
- **Una perspectiva de interacción**, donde se modelan las interacciones entre un sistema y su entorno o entre los componentes.
- **Una perspectiva estructural**, donde se modela la organización de un sistema o la estructura de los datos que procesa.
- **Una perspectiva de comportamiento**, donde se modela el comportamiento dinámico del sistema y cómo responde a los eventos.

# Lenguaje de Modelado Unificado (UML)

- El Lenguaje de Modelado Unificado (UML) es un conjunto de 13 tipos de diagramas diferentes que se pueden usar para modelar sistemas de software.
- El UML es universalmente aceptado como el enfoque estándar para desarrollar modelos de sistemas de software.
- Se han propuesto variantes para modelos de sistemas más generales.

# Lenguaje de Modelado Unificado (UML)

- **Diagramas de actividades:** muestran las actividades involucradas en un proceso o en el procesamiento de datos.
- **Diagramas de casos,** que muestran las interacciones entre un sistema y su entorno.
- **Diagramas de secuencia,** que muestran las interacciones entre los actores y el sistema y entre los componentes.
- **Diagramas de clases,** que muestran las clases de objetos en el sistema y las asociaciones entre estas clases.
- **Diagramas de estado,** que muestran cómo el sistema reacciona ante eventos internos y externos.



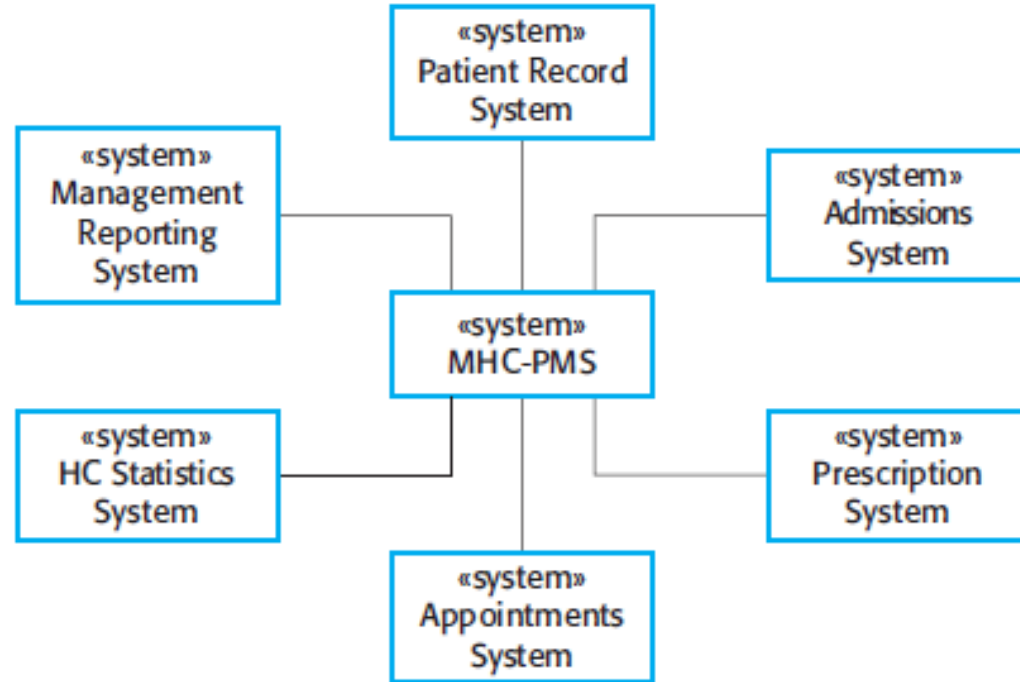
# Modelos de Contexto

- En una etapa temprana de la especificación de un sistema, debe decidir sobre **los límites del sistema**.
- Esto implica trabajar con las partes interesadas para decidir **qué funcionalidad** se debe **incluir** en el sistema y **qué se proporciona en el entorno del sistema**.
- En algunos casos, el límite entre un sistema y su entorno es relativamente claro.
- En otros casos, hay más flexibilidad...

# Modelos de Contexto

- Una vez que se han tomado algunas decisiones sobre los límites del sistema, parte de la actividad de análisis es la **definición de ese contexto y las dependencias que un sistema tiene en su entorno.**
- Normalmente, producir un modelo arquitectónico simple es el primer paso en esta actividad.

# Modelos de Contexto



Contexto de un Sistema de Manejo de Pacientes con Salud Mental.

# Modelos de Contexto

- Los modelos de contexto normalmente muestran que el entorno incluye varios otros sistemas automatizados, sin embargo, **no muestran los tipos de relaciones** entre los sistemas en el entorno y el sistema que se está especificando.
- Los sistemas externos pueden:
  - producir datos o consumir datos del sistema,
  - compartir datos con el sistema,
  - pueden estar conectados directamente, a través de una red o no estar conectados en absoluto.
  - podrían estar ubicados conjuntamente o ubicados en edificios separados.

# Modelos de Contexto

- Los modelos de contexto normalmente muestran que el entorno incluye varios otros sistemas automatizados, sin embargo, **no muestran los tipos de relaciones** entre los sistemas en el entorno y el sistema que se está especificando.
- Los sistemas externos pueden:
  - producir datos o consumir datos del sistema,
  - compartir datos con el sistema,
  - pueden estar conectados al sistema o no estar conectados en absoluto.
  - podrían estar ubicados separados.

Todas estas relaciones pueden afectar los requisitos y el diseño del sistema que se define y deben tenerse en cuenta.

# Modelos de Contexto

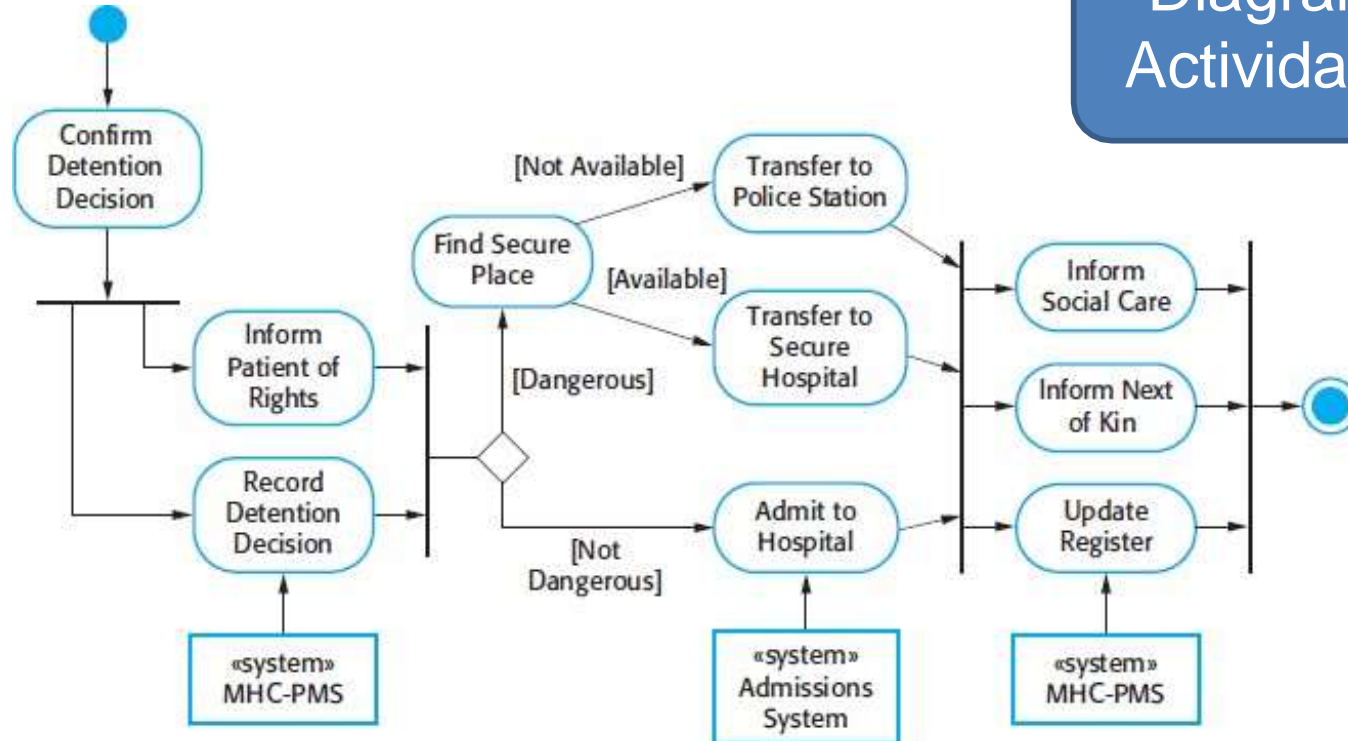
- Los modelos de contexto normalmente muestran que el entorno incluye varios otros sistemas automatizados, sin embargo, **no muestran los tipos de relaciones** entre los sistemas en el entorno y el sistema que se está especificando.
  - Los sistemas externos pueden:
    - producir datos o consumir datos del sistema,
    - compartir datos con el sistema,
    - pueden estar conectados directamente, a través de una red o no estar conectado, se puede o no.
- Por lo tanto, los modelos de contexto simples se usan junto con otros modelos, como los modelos de procesos.

# Modelos de Contexto

- Los **modelos de procesos** describen los procesos humanos y automatizados en los que se utilizan sistemas de software en particular.

# Modelos de Contexto

Diagrama de  
Actividad UML



Modelo de proceso de detención involuntaria



# Modelos de Contexto

- Los **diagramas de actividades** están destinados a mostrar las actividades que componen un proceso de sistema y el flujo de control de una actividad a otra.

# Modelos de Contexto

## Diagrama de Actividad UML:

- El **inicio** de un proceso se indica con un **círculo relleno**; el **final** por un **círculo lleno dentro de otro círculo**.
- Los **rectángulos con esquinas redondeadas** representan **actividades**, es decir, los subprocesos específicos que deben llevarse a cabo.
- Puede incluir **objetos** en **cuadros de actividad**.
- Las **flechas** representan **el flujo de trabajo** de una actividad a otra.

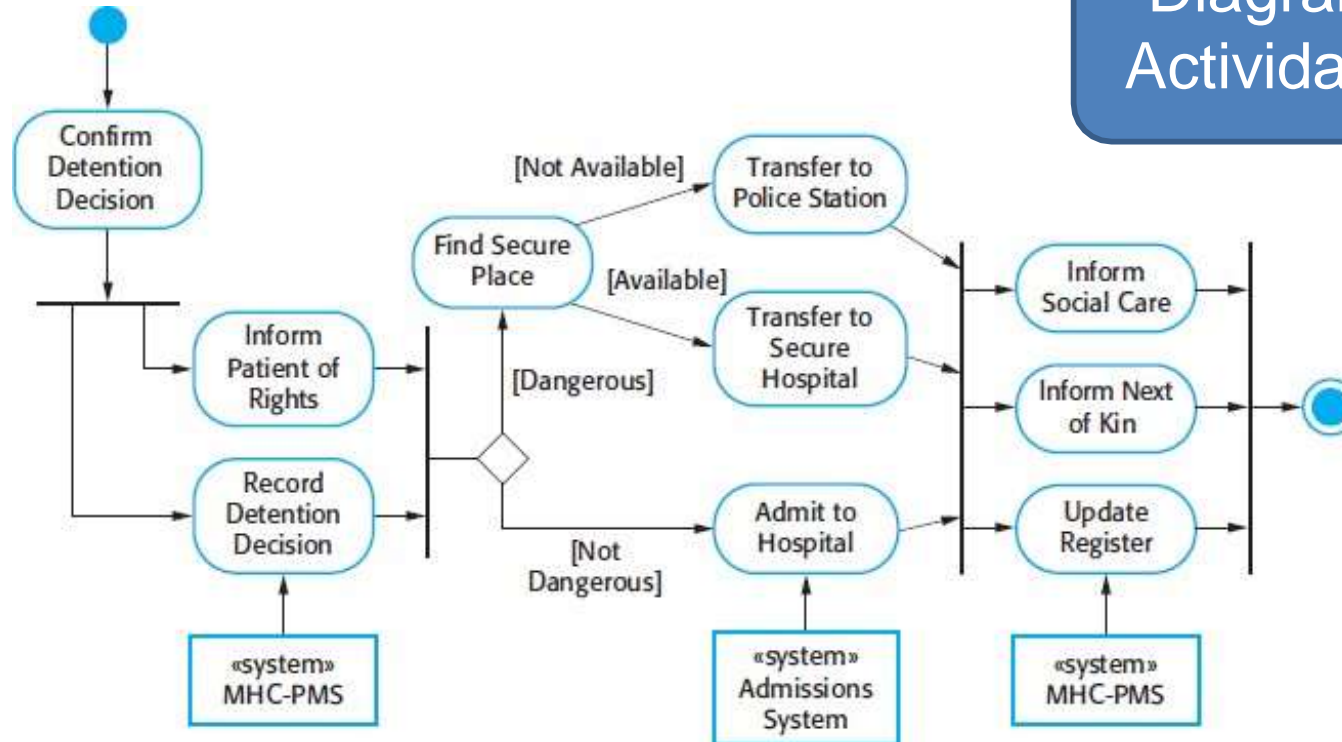
# Modelos de Contexto

## Diagrama de Actividad UML:

- Una **barra sólida** se utiliza para indicar la **coordinación** de la actividad.
- Cuando el flujo de más de una actividad conduce a una barra sólida, todas estas actividades deben completarse antes de que el progreso sea posible.
- Cuando el flujo de una barra sólida conduce a una serie de actividades, estas se pueden ejecutar en paralelo.

# Modelos de Contexto

Diagrama de  
Actividad UML



Modelo de proceso de detención involuntaria

# Modelos de Interacción

- Todos los sistemas implican interacción de algún tipo.
  - Esta puede ser la interacción del usuario,
  - la interacción entre el sistema y otros sistemas o entre los componentes del mismo.

Modelar la **interacción del usuario** es importante ya que ayuda a identificar los requisitos del usuario.

La **interacción entre el sistema** y el entorno resalta los problemas de comunicación que pueden surgir.

# Modelos de Interacción

Hay dos enfoques relacionados con el modelado de interacción:

- **Modelado de casos**, que se utiliza principalmente para modelar las interacciones entre un sistema y actores externos (usuarios u otros sistemas).
- **Diagramas de secuencia**, que se utilizan para modelar las interacciones entre los componentes del sistema, aunque también se pueden incluir agentes externos.

# Modelado de Casos de Uso (Interacción)

- Un **caso de uso** se puede describir como un escenario simple que describe lo que un usuario espera de un sistema.
- En su forma más simple, un caso de uso se muestra como una elipse con los actores involucrados representados con figuras de palo.



# Modelado de Casos de Uso (Interacción)

- La notación de **figura de palo** se desarrolló originalmente para cubrir la interacción humana, pero también se usa para representar otros sistemas y hardware externos.
- Formalmente, los diagramas de casos de uso deben usar líneas sin flechas, ya que las flechas en el UML indican la dirección del flujo de mensajes.





# Modelado de Casos de Uso (Interacción)

- Los diagramas de casos de uso ofrecen una visión general bastante simple de una interacción, por lo que se debe proporcionar más detalles para comprender lo que está ocurriendo.
- Este detalle se puede incorporar mediante una **descripción textual** simple, una **descripción estructurada en una tabla** o un **diagrama de secuencia**.

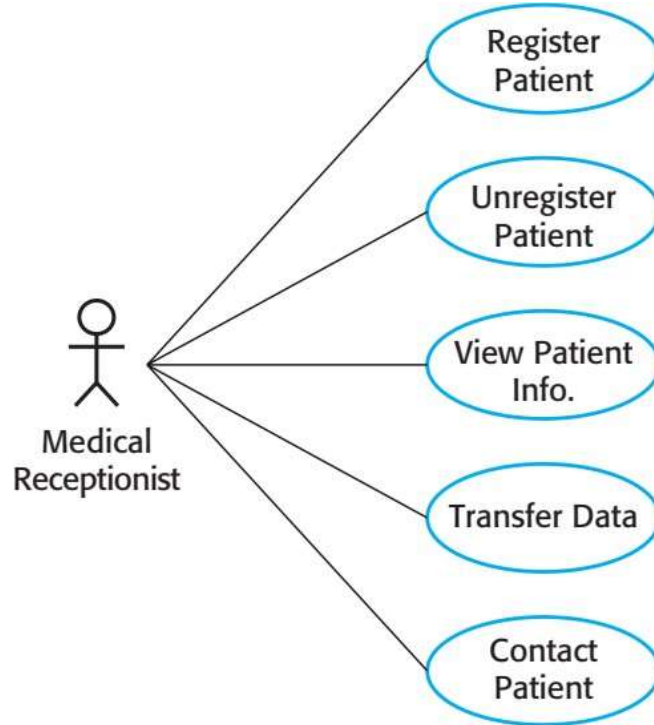
# Modelado de Casos de Uso (Interacción)

## MHC-PMS: Transferencia de datos

Actores	Recepcionista, sistema de registros de pacientes (PRS)
Descripción	Una recepcionista puede transferir datos del MHC-PMS a una base de datos general de registros de pacientes. La información transferida puede ser información personal actualizada (dirección, número de teléfono, etc.) o un resumen del diagnóstico y tratamiento del paciente.
Datos	Información personal del paciente, resumen del diagnóstico y tratamiento
Estímulos	Comando de usuario emitido por la recepcionista
Respuesta	Confirmación que el sistema PRS ha sido actualizado
Comentarios	La recepcionista debe tener los permisos de seguridad apropiados para acceder a la información del paciente y al PRS.



# Modelado de Casos de Uso (Interacción)



# Diagramas de Secuencias (Interacción)

- Los **diagramas de secuencia** en UML se usan principalmente para modelar las **interacciones** entre los **actores** y los **objetos** en un sistema y las interacciones entre los objetos mismos.
- Muestra la **secuencia de interacciones** que tiene lugar durante un caso de uso o instancia de caso de uso.
- El UML tiene una rica sintaxis para diagramas de secuencia, que permite modelar muchos tipos diferentes de interacción.

# Diagramas de Secuencias (Interacción)

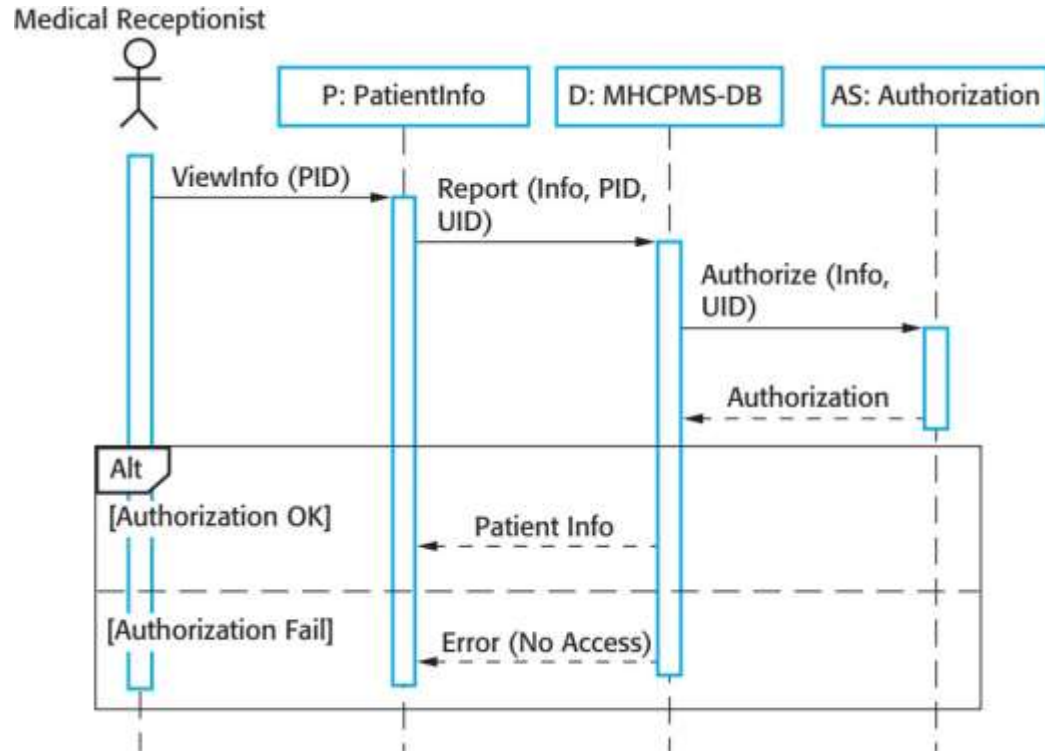


Diagrama de Secuencia para “ver información del paciente”

# Diagramas de Secuencias (Interacción)

- Los **objetos** y **actores** involucrados se enumeran a lo largo de la parte superior del diagrama, con una línea punteada dibujada verticalmente a partir de estos.
- Las **interacciones** entre objetos se indican con flechas comentadas.
- El **rectángulo** en las líneas punteadas indica la línea de vida del objeto en cuestión (es decir, el momento en que la instancia del objeto está involucrada).
- La secuencia de interacciones se lee de arriba a abajo.
- Los **comentarios** en las flechas indican las llamadas a los objetos, sus parámetros y los valores de retorno.
- También se pueden mostrar **alternativas**. Se usa un cuadro llamado *alt* con las condiciones indicadas entre corchetes.

# Diagramas de Secuencias (Interacción)

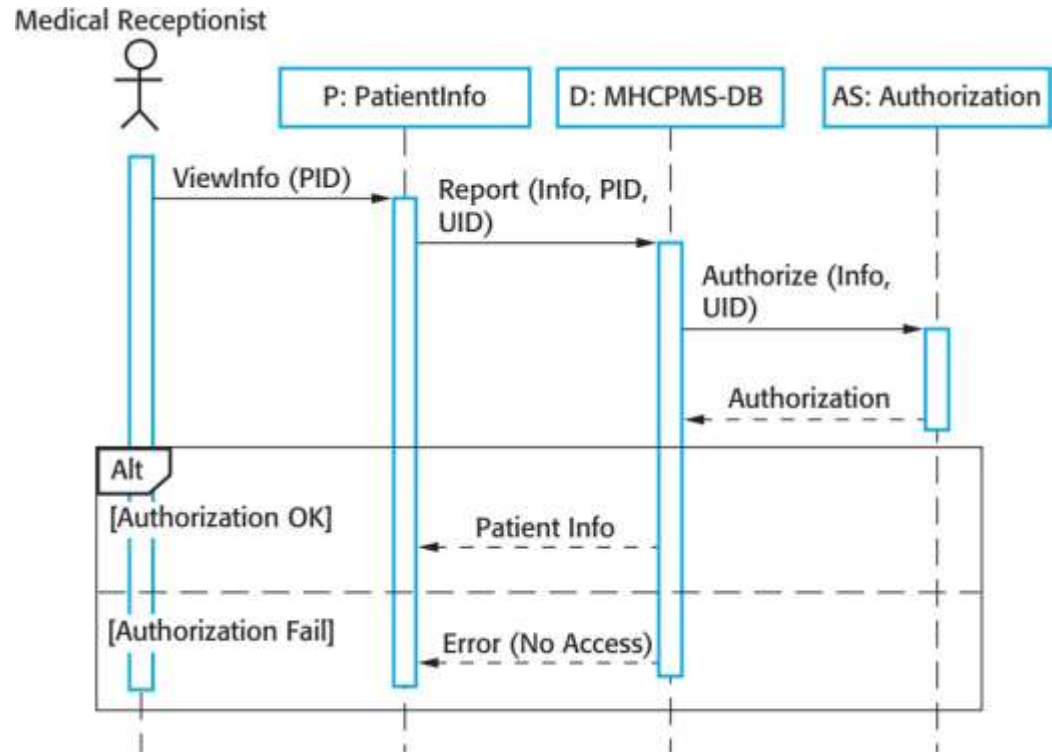


Diagrama de Secuencia para “ver información del paciente”

# Diagramas de Secuencias (Interacción)

- La recepcionista activa el método *ViewInfo* en una instancia P de la clase de objeto *PatientInfo*, proporcionando el identificador del paciente, PID.
- P es un objeto de interfaz de usuario, que muestra información del paciente en forma de formulario.

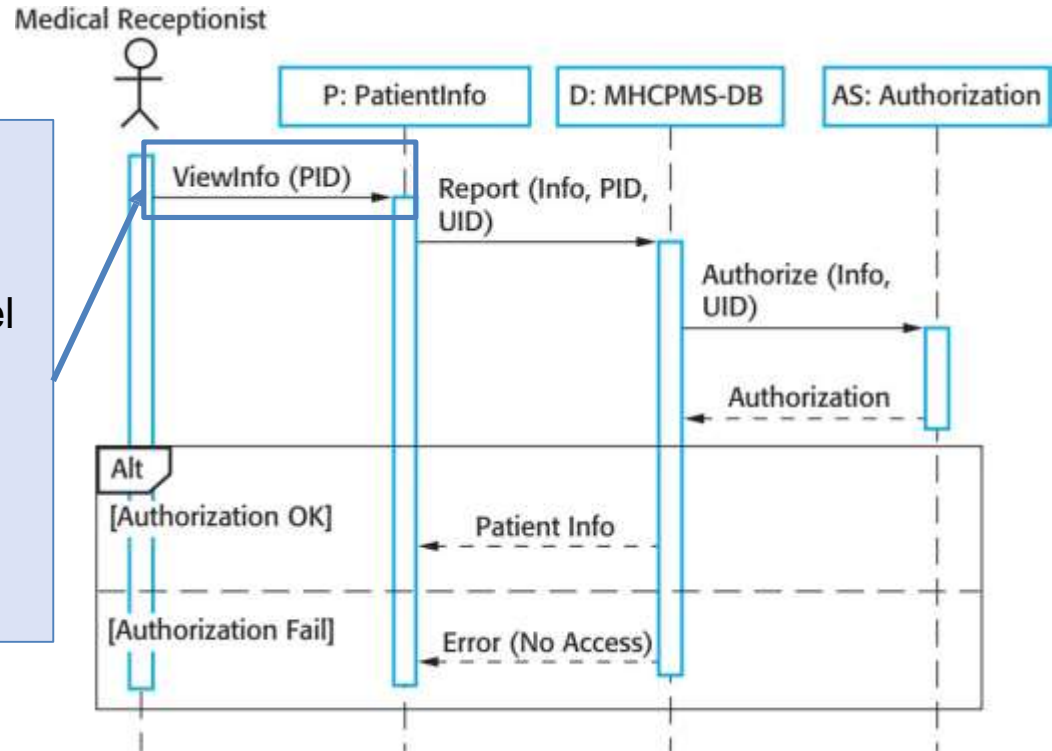


Diagrama de Secuencia para “ver información del paciente”



# Diagramas de Secuencias (Interacción)

P llama a la base de datos para solicitar la información requerida, proporcionando el identificador PID otorgado por la recepcionista.

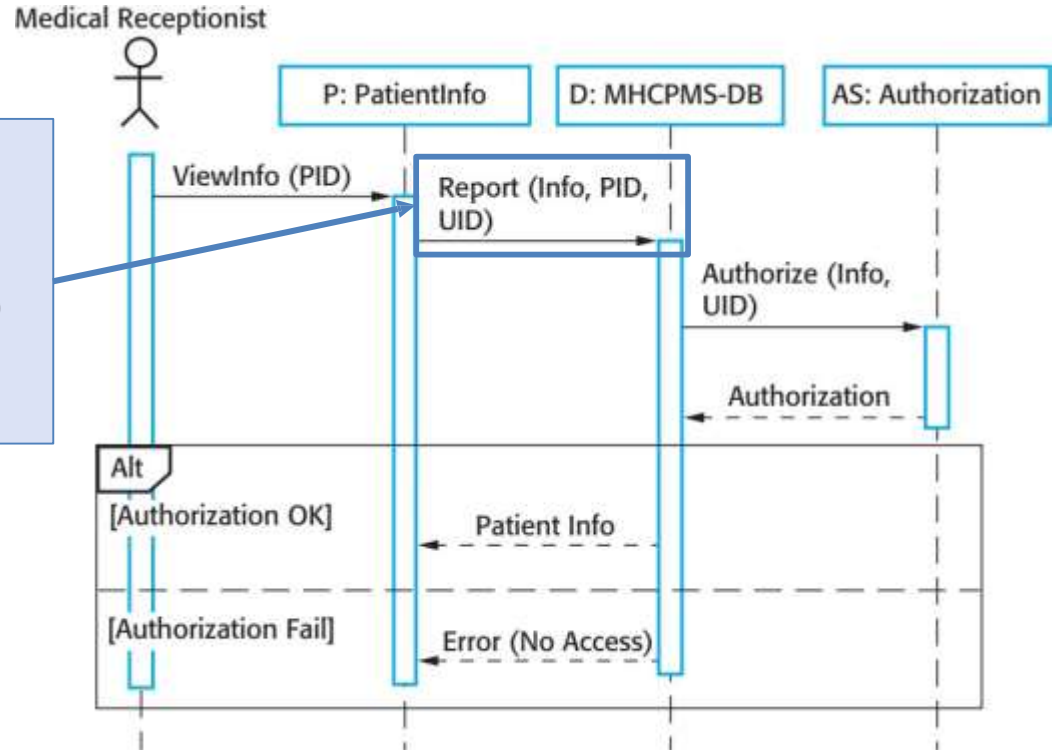


Diagrama de Secuencia para “ver información del paciente”

# Diagramas de Secuencias (Interacción)

La base de datos comprueba con un sistema de autorización que el usuario está autorizado para esta acción.

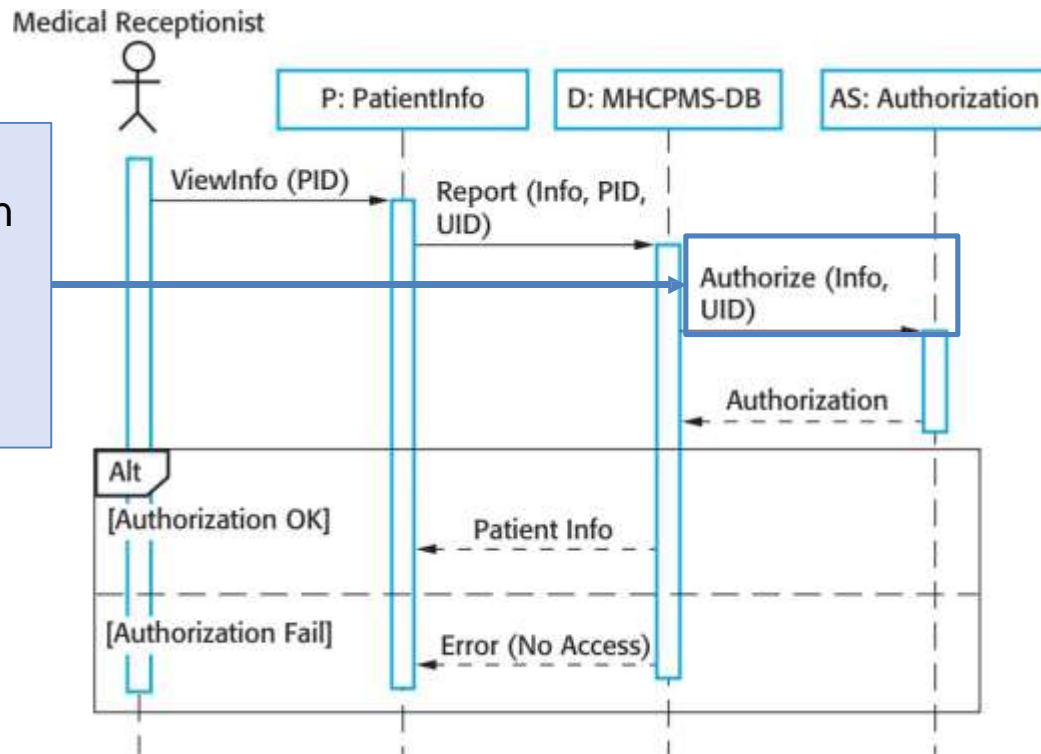


Diagrama de Secuencia para “ver información del paciente”

# Diagramas de Secuencias (Interacción)

Si se autoriza, se devuelve la información del paciente y se completa un formulario en la pantalla del usuario. Si la autorización falla, se devuelve un mensaje de error.

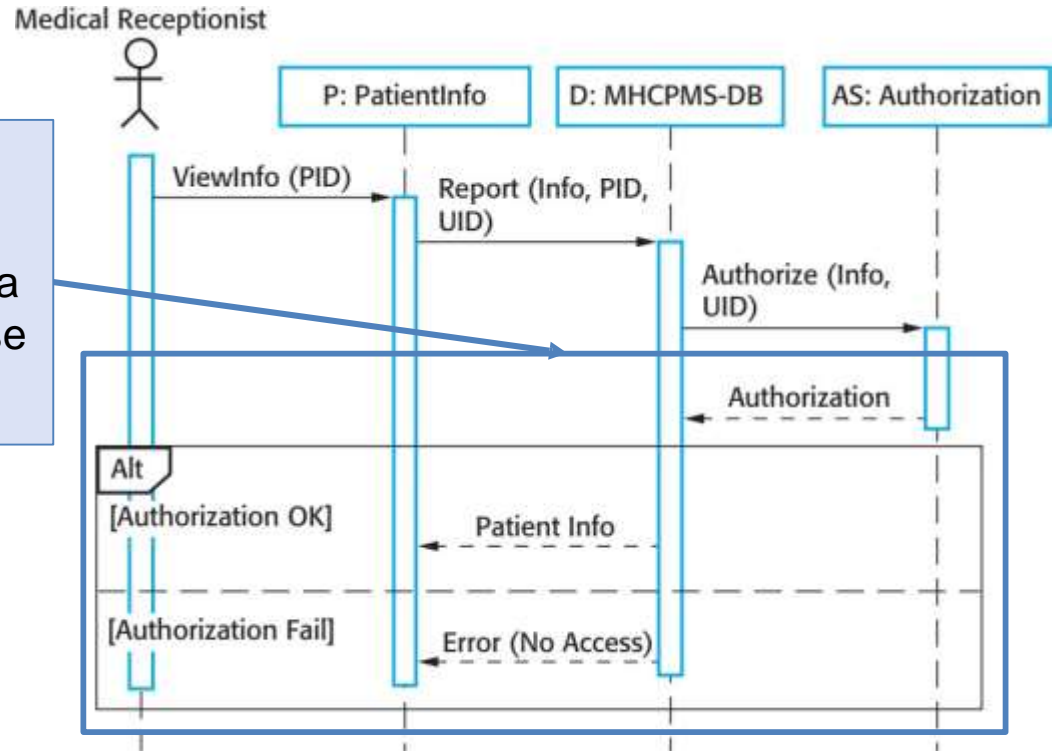


Diagrama de Secuencia para “ver información del paciente”

# Diagra

ción)

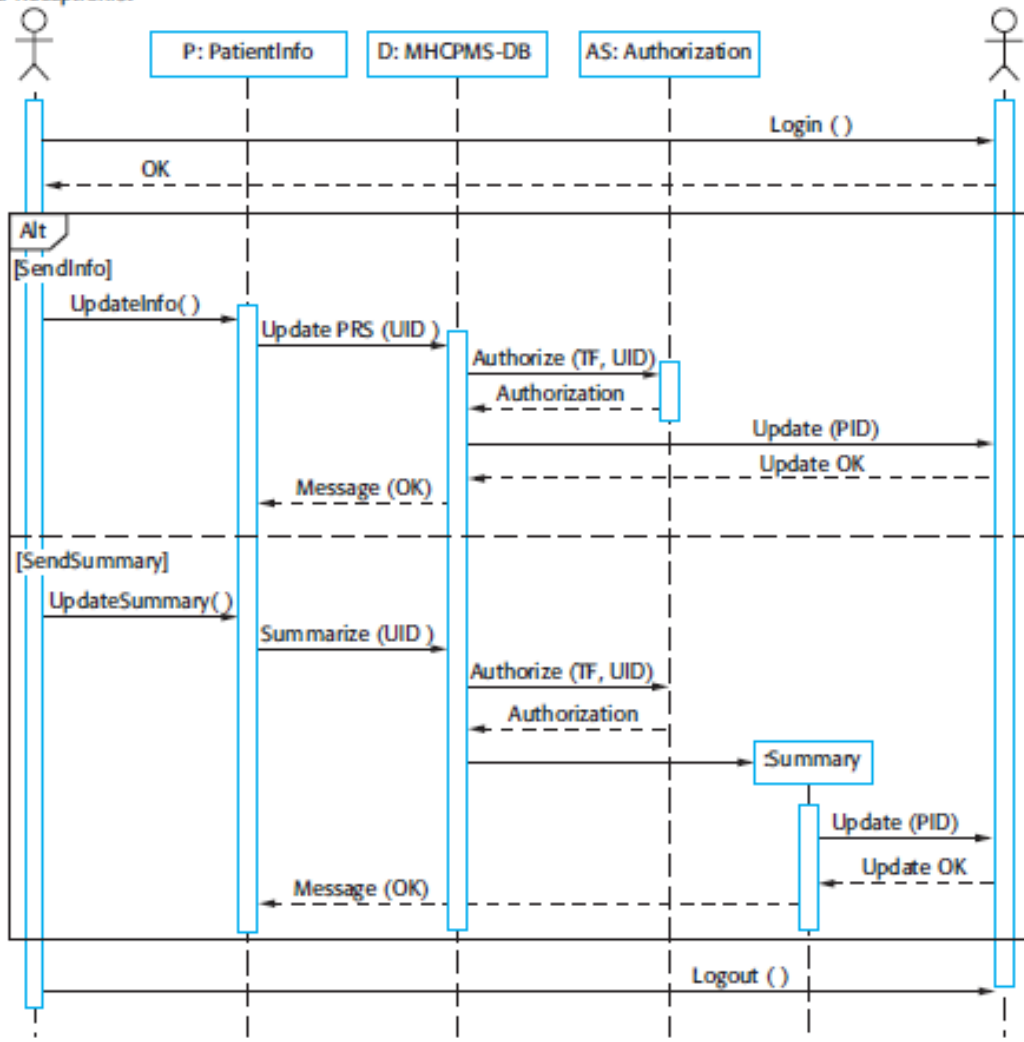
Medical Receptionist

PRS

P: PatientInfo

D: MHCPMS-DB

AS: Authorization



# Diagramas de S

La recepcionista inicia sesión en el PRS.

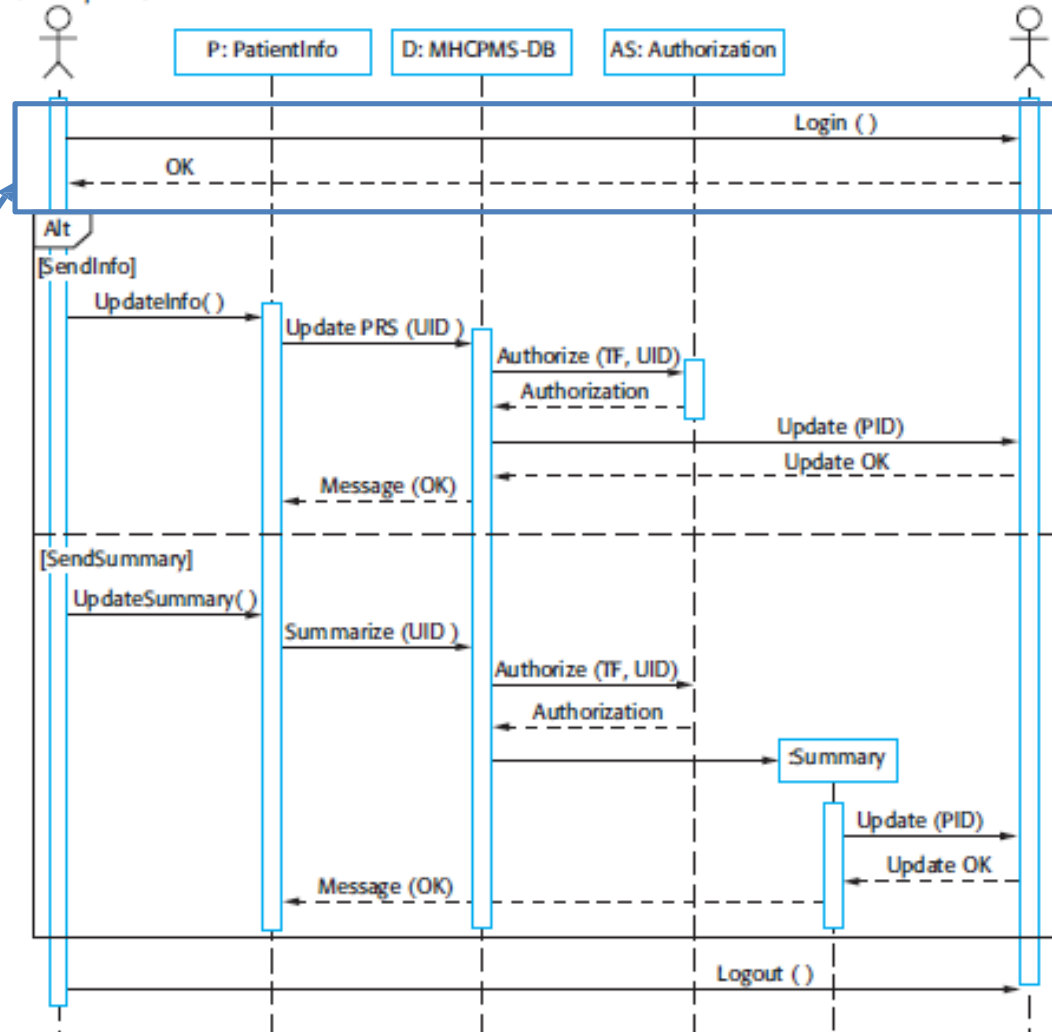
Medical Receptionist

PRS

P: PatientInfo

D: MHCPMS-DB

AS: Authorization

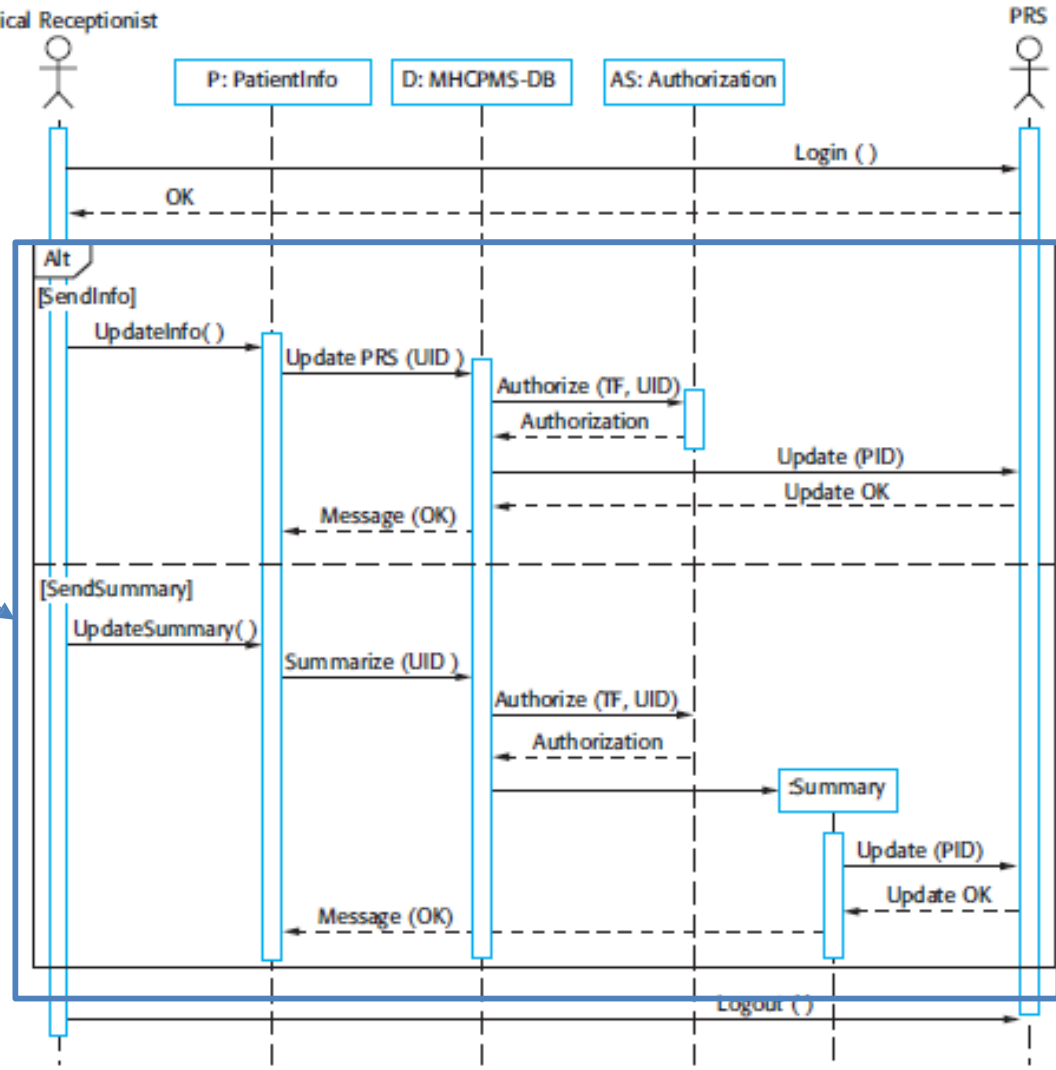


# Diagramas de S

Hay dos alternativas posibles:

1. La transferencia directa de información actualizada del paciente al PRS,
2. La transferencia de datos de salud resumidos del MHC-PMS al PRS.

Medical Receptionist



# Diagramas de S

En cada caso, los permisos de la recepcionista se verifican usando el sistema de autorización.

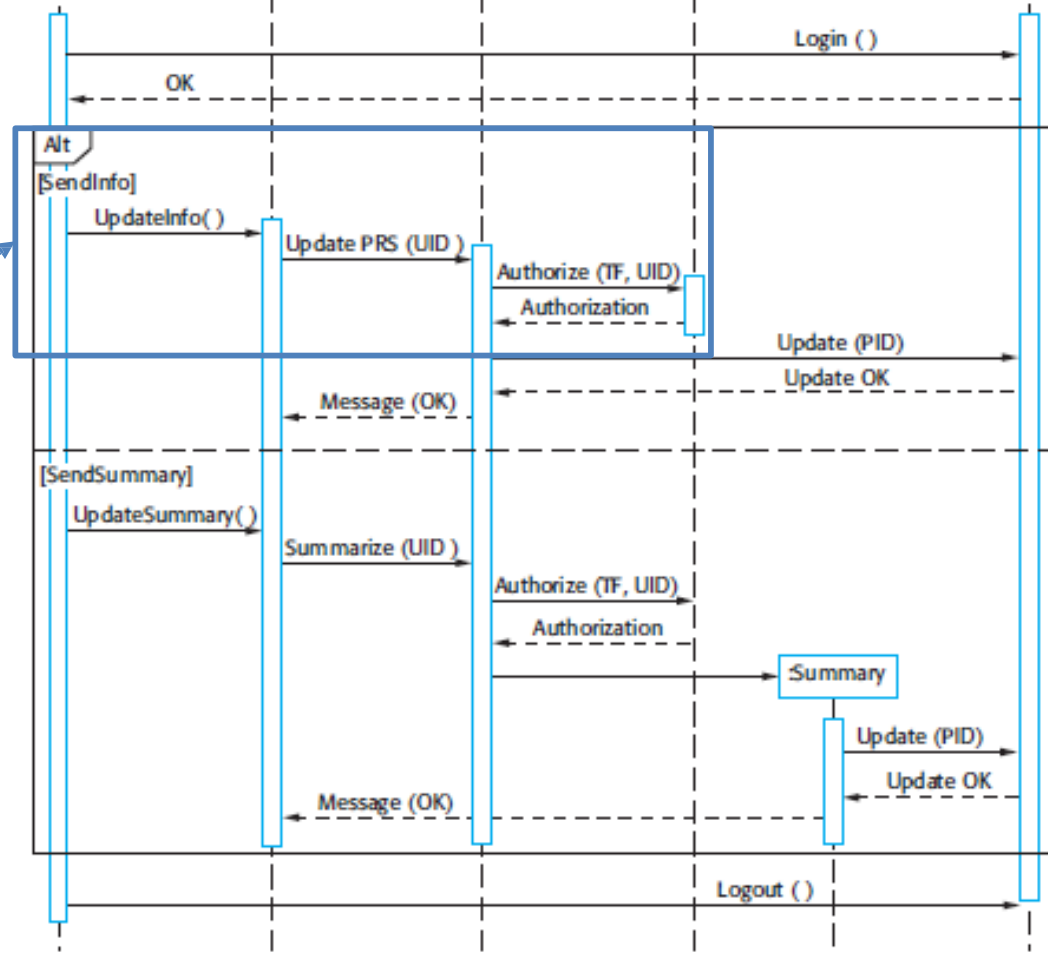
Medical Receptionist



P: PatientInfo

D: MHCPMS-DB

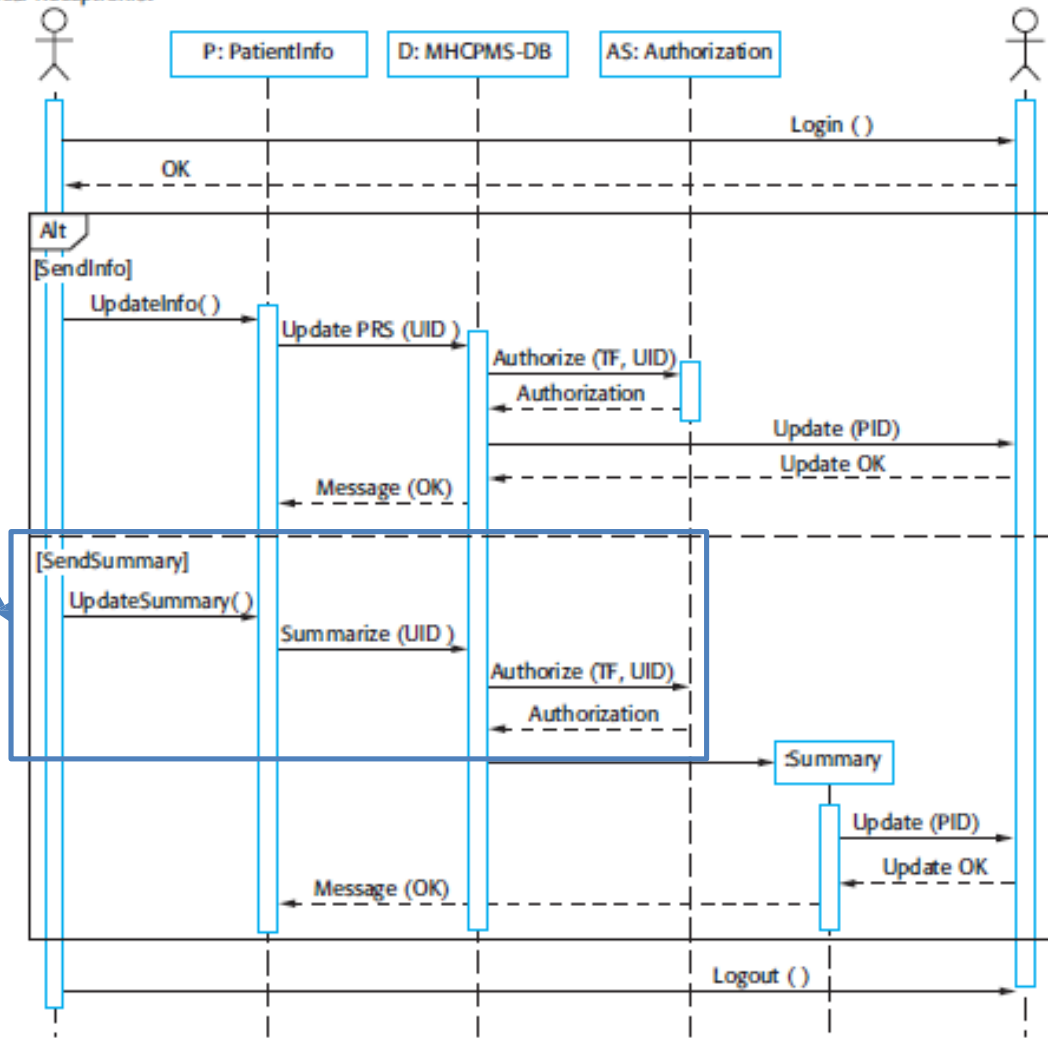
AS: Authorization



# Diagramas de S

En cada caso, los permisos de la recepcionista se verifican usando el sistema de autorización.

Medical Receptionist





# Diagramas de S

La información personal se puede transferir directamente desde el objeto de interfaz de usuario a la PRS.

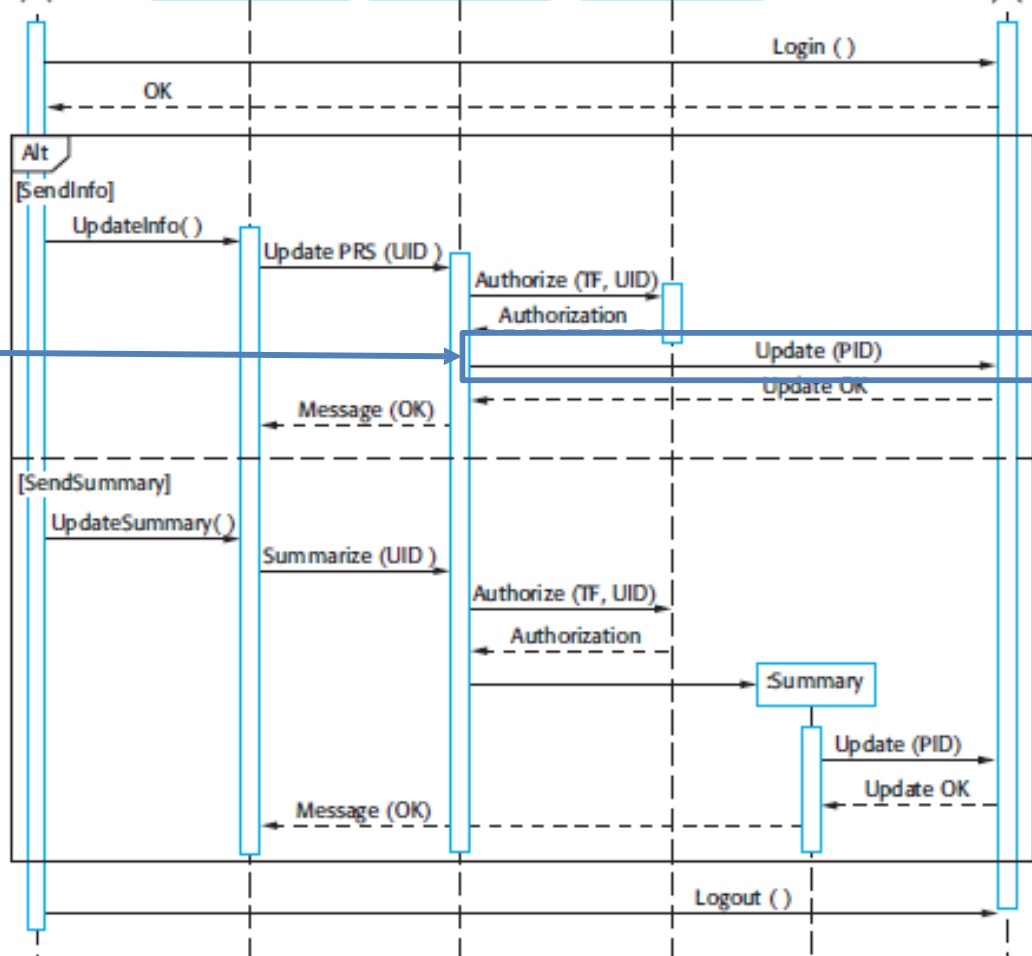
Medical Receptionist



P: PatientInfo

D: MHCPMS-DB

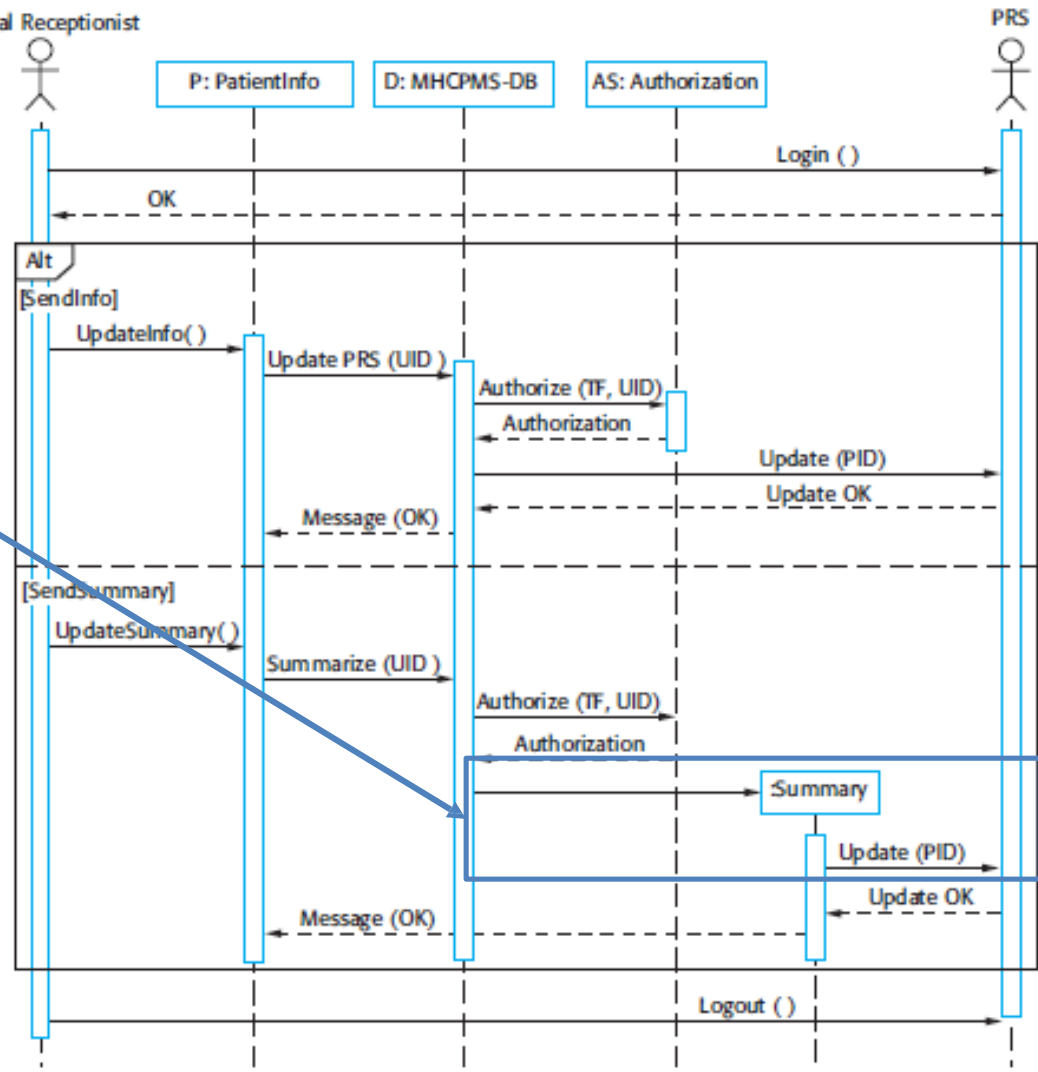
AS: Authorization



# Diagramas de S

Alternativamente, se puede crear un registro de resumen a partir de la base de datos y ese registro se transfiere

Medical Receptionist



# Diagramas de S

Al finalizar la transferencia, la PRS emite un mensaje de estado y el usuario cierra la sesión.

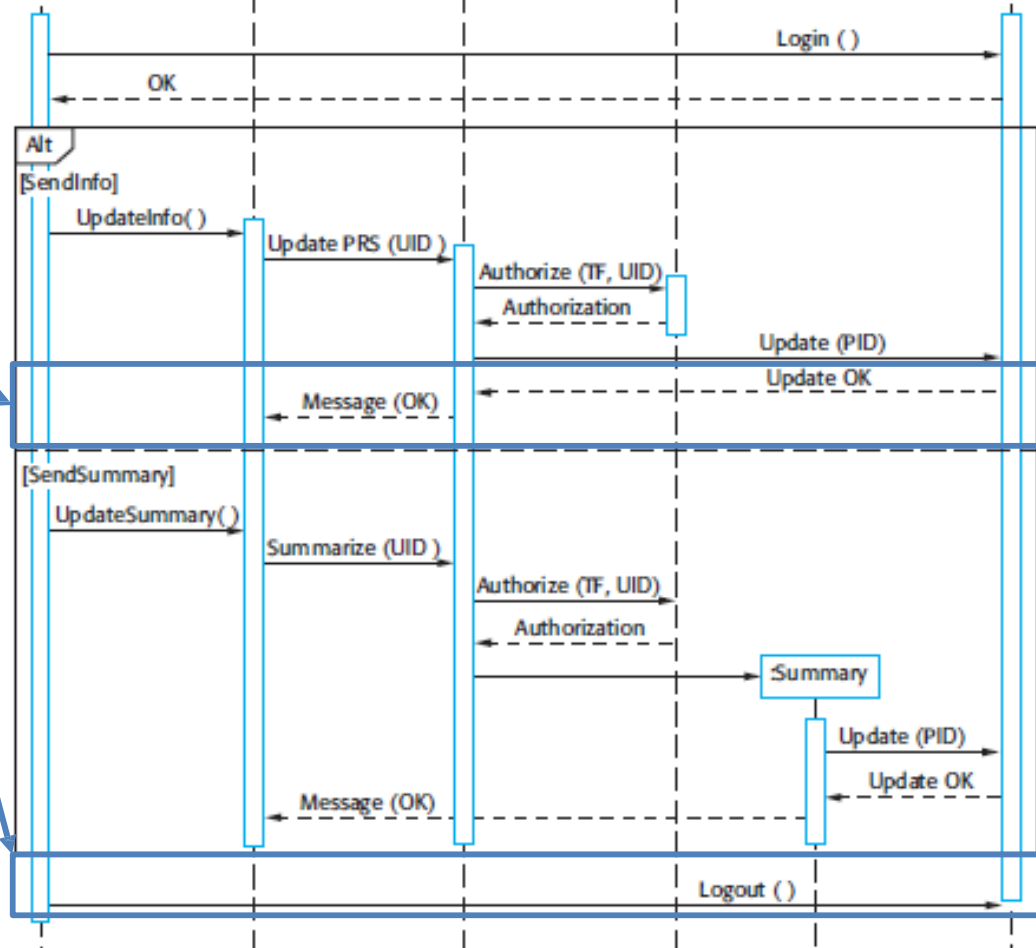
Medical Receptionist



P: PatientInfo

D: MHCPMS-DB

AS: Authorization



# Diagramas de S

Al finalizar la transferencia, la PRS emite un mensaje de estado y el usuario cierra la sesión.

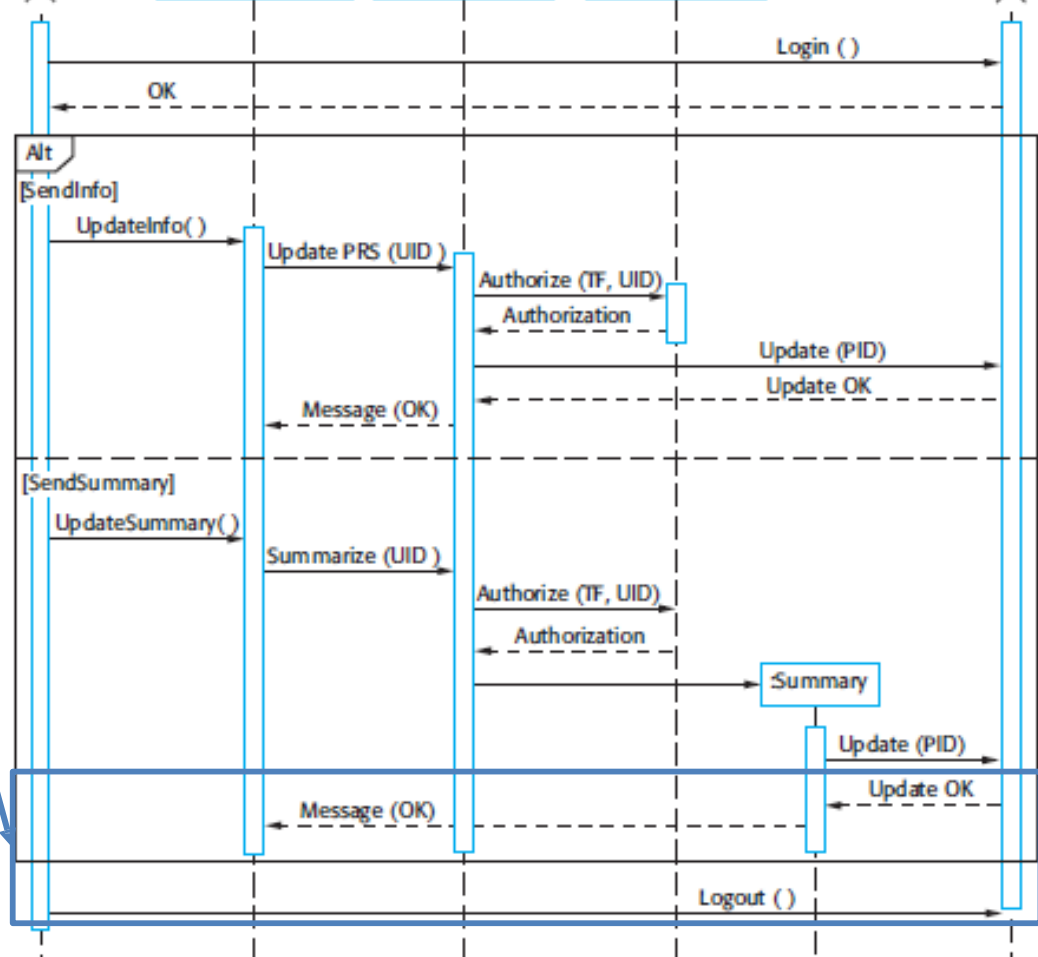
Medical Receptionist



P: PatientInfo

D: MHCPMS-DB

AS: Authorization



# Modelos Estructurales

- Los **modelos estructurales** muestran la **organización** de un sistema en términos de los componentes del sistema y sus relaciones.
- Los modelos estructurales pueden ser:
  - **estáticos**, que muestran la estructura del diseño del sistema.
  - **dinámicos**, que muestran la organización del sistema cuando se está ejecutando.

# Modelos Estructurales

- Los **modelos estructurales** muestran la **organización** de un sistema en términos de los componentes del sistema y sus relaciones.
- Los modelos estructurales pueden ser:
  - **estáticos**, que muestran la estructura del diseño de un sistema. La organización dinámica de un sistema como un conjunto de hilos que interactúan puede ser muy diferente de un modelo estático de los componentes de un sistema.
  - **dinámicos**, que muestran la organización del sistema cuando se está ejecutando.

# Modelos Estructurales

- Los modelos estructurales de un sistema se crean cuando está discutiendo y diseñando la arquitectura del sistema.
- El lenguaje UML (diagrama de componentes, paquetes y desarrollo) se pueden usar al presentar modelos arquitectónicos.

# Diagramas de Clase (Estructurales)

- Los **diagramas de clases** se utilizan en el modelado de un sistema orientado a objetos para mostrar las clases en un sistema y las asociaciones entre ellas
- Inicialmente, los objetos representan elementos del mundo real, como un paciente, una receta, un médico, etc.
- A medida que se desarrolla una implementación, se definen objetos de implementación adicionales necesarios para proporcionar la funcionalidad requerida.

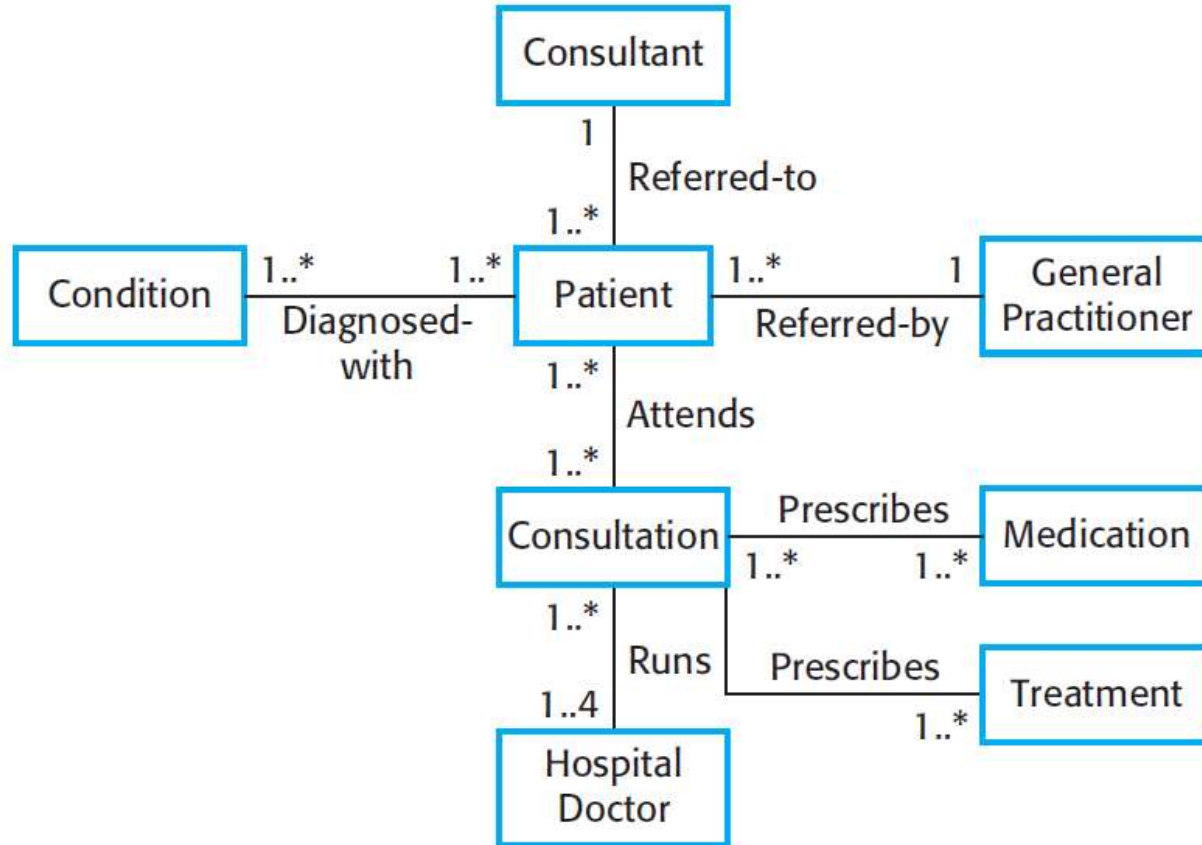


# Diagramas de Clase (Estructurales)

- Los **diagramas de clases** en UML se pueden expresar en diferentes niveles de detalle.



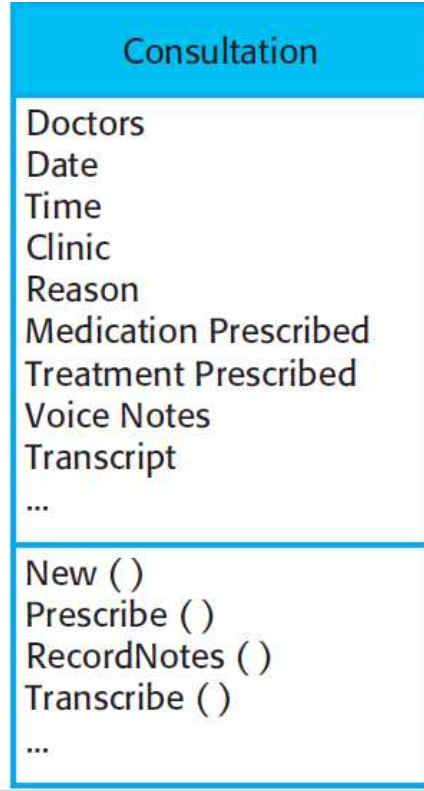
# Diagramas de Clase (Estructurales)



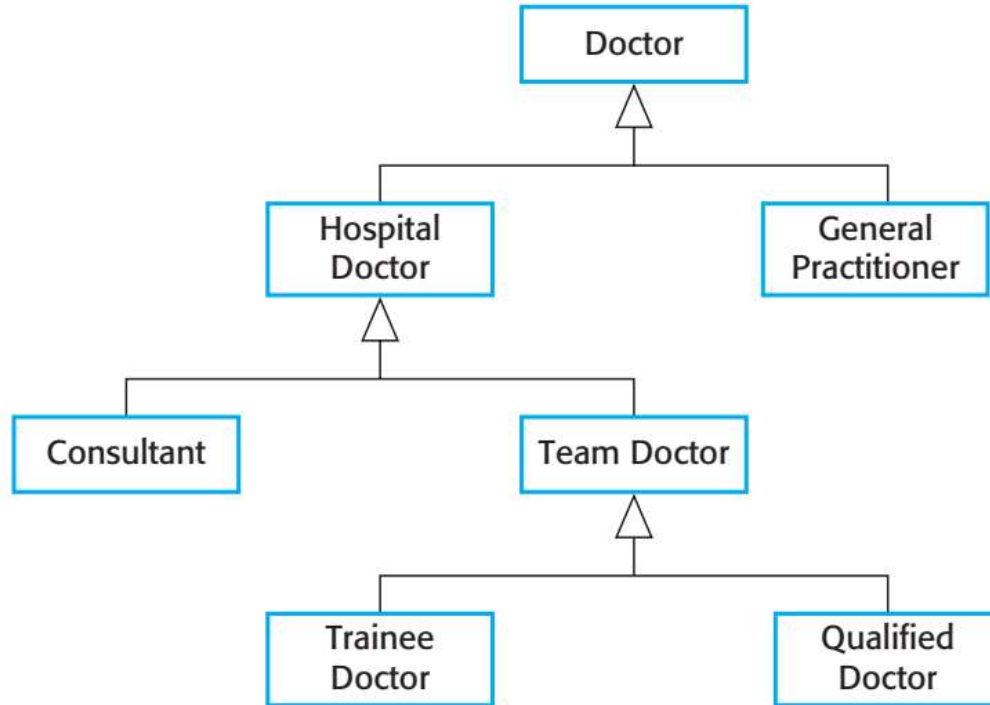
# Diagramas de Clase (Estructurales)

- El UML también permite que se especifique el **rol** de los objetos que participan en la asociación.
- En este nivel de detalle, los diagramas de clase parecen modelos de datos semánticos.
- Los **modelos de datos semánticos** se utilizan en el diseño de la base de datos. Muestran las **entidades** de datos, sus **atributos** asociados y las **relaciones** entre estas entidades.

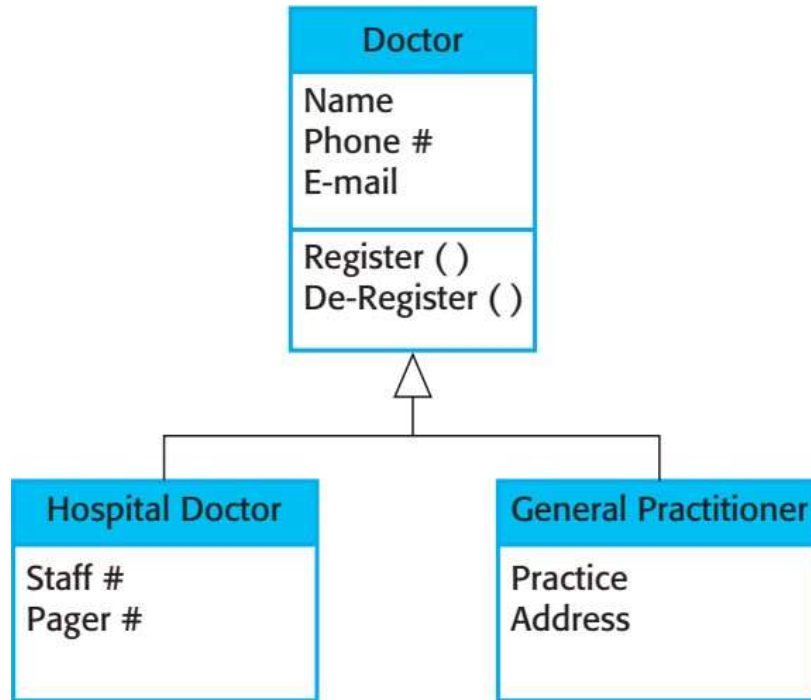
# Diagramas de Clase (Estructurales)



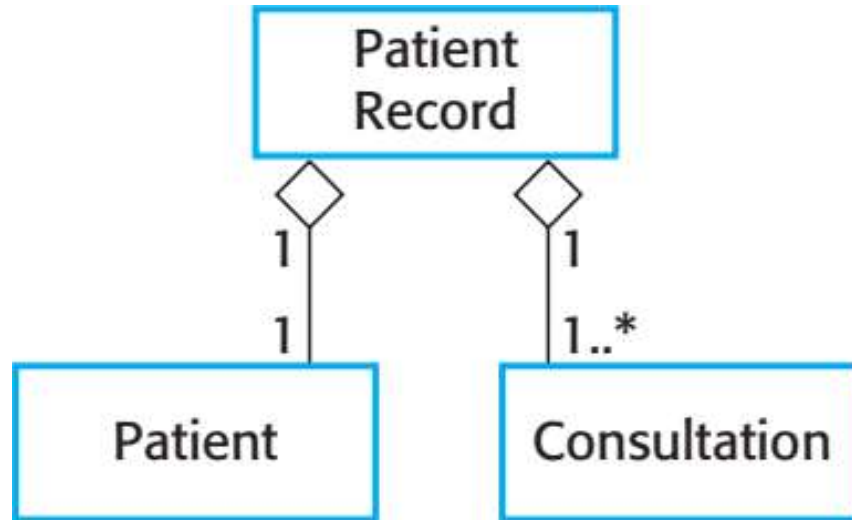
# Diagramas de Clase (Estructurales)



# Diagramas de Clase (Estructurales)



# Diagramas de Clase (Estructurales)



# Modelos de Comportamiento

- Los **modelos de comportamiento** son modelos del comportamiento dinámico del sistema mientras se está ejecutando.
- Muestran lo que sucede o lo que se supone que debe suceder cuando un sistema responde a un estímulo de su entorno.
- Los estímulos pueden ser de dos tipos:
  - **Datos**: Llegan datos que el sistema debe procesar.
  - **Eventos**: Ocurre algún evento que desencadena el procesamiento.



# Modelos de Comportamiento

- Muchos sistemas comerciales están principalmente **impulsados por datos**.
- Son controlados por la entrada de datos al sistema con relativamente poco procesamiento de eventos externos.
- Su procesamiento implica una secuencia de acciones sobre esos datos y la generación de un resultado.
- Por ejemplo, un sistema de facturación telefónica aceptará información sobre llamadas realizadas por un cliente, calculará los costos de estas llamadas y generará una factura para enviar a ese cliente.

# Modelos de Comportamiento

- Los sistemas en tiempo real a menudo son **impulsados por eventos**.
- Este tipo de sistemas involucra un procesamiento de datos mínimo.
- Por ejemplo, un sistema de conmutación de teléfono fijo responde a eventos tales como 'receptor descolgado' al generar un tono de marcado, o presionar teclas en un teléfono al capturar el número de teléfono, etc.

# Modelado Basado en Datos (Comp.)

- Los **modelos basados en datos** muestran la secuencia de acciones involucradas en el procesamiento de los datos de entrada y la generación de un resultado asociado.
- Son particularmente útiles durante el análisis de los requisitos, ya que se pueden usar para mostrar el procesamiento de extremo a extremo en un sistema.
- Muestran **la secuencia completa de acciones** que se lleva a cabo desde que se procesa una entrada hasta alcanzar el resultado correspondiente.

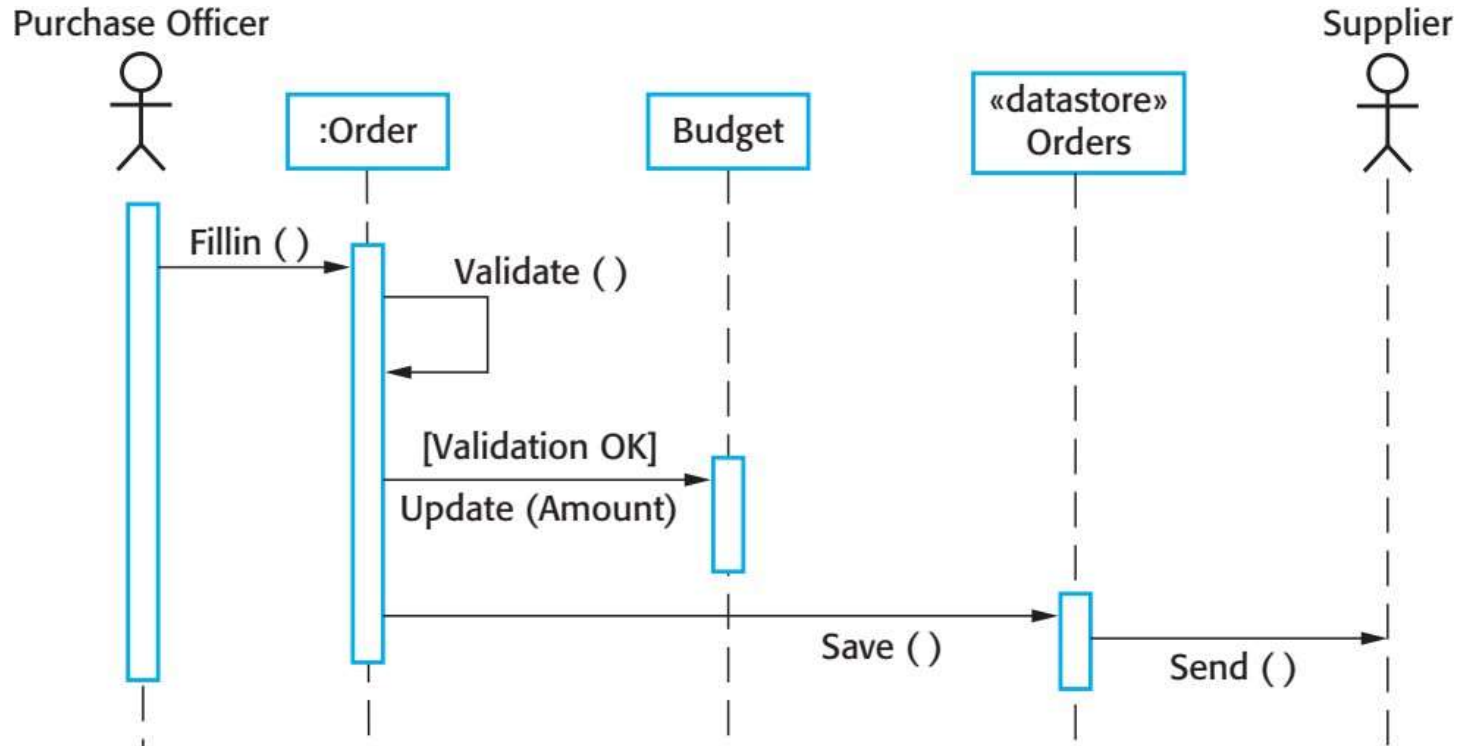
# Modelado Basado en Datos (Comp.)

- Los **diagramas de flujo de datos (DFD)** ilustran los pasos de procesamiento en un sistema.
- Los **modelos de flujo de datos** permiten el seguimiento de cómo se mueven los datos asociados con un proceso en particular a través del sistema.
- UML no admite diagramas de flujo de datos como se propusieron originalmente, sin embargo, UML 2.0 introdujo **diagramas de actividad**.

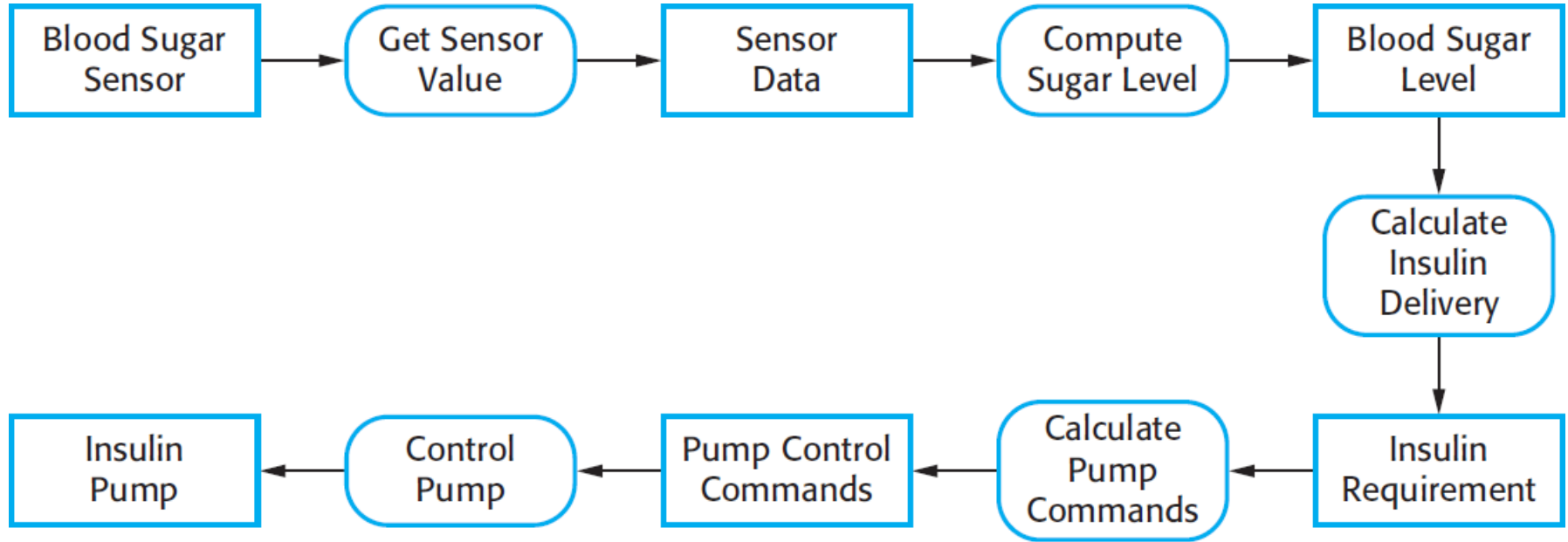
# Modelado Basado en Datos (Comp.)

- Se puede utilizar un diagrama de actividades para representar un flujo de datos representando:
  - los pasos de procesamiento como actividades
  - los datos que fluyen entre estos pasos como objetos.
- Una forma alternativa de mostrar la secuencia de procesamiento en un sistema es usar **diagramas de secuencia UML**.

# Modelado Basado en Datos (Comp.)



# Modelado Basado en Datos (Comp.)



Procesamiento de una orden de pedido.

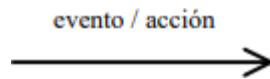
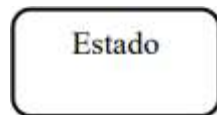
# Modelado Dirigido por Eventos (Comp.)

- El **modelado basado en eventos** muestra cómo un sistema responde a eventos externos e internos.
- Se basa en la suposición de que un sistema tiene un número finito de estados y que los eventos (estímulos) pueden causar una transición de un estado a otro.
- Los **diagramas de estado** muestran estados del sistema y eventos que causan transiciones de un estado a otro.
- No muestran el flujo de datos dentro del sistema, pero pueden incluir información adicional sobre los cálculos realizados en cada estado.



# Modelado Dirigido por Eventos (Comp.)

- En **diagramas de estados UML**:
  - Los rectángulos redondeados representan los **estados** del sistema. Pueden incluir una breve descripción (después de "hacer") de las acciones tomadas en ese estado.
  - Las flechas etiquetadas representan **estímulos** que fuerzan una transición de un estado a otro.
  - Puede indicar los estados de **inicio** y **final** utilizando círculos rellenos, como en los diagramas de actividad.



**Estado Inicial**



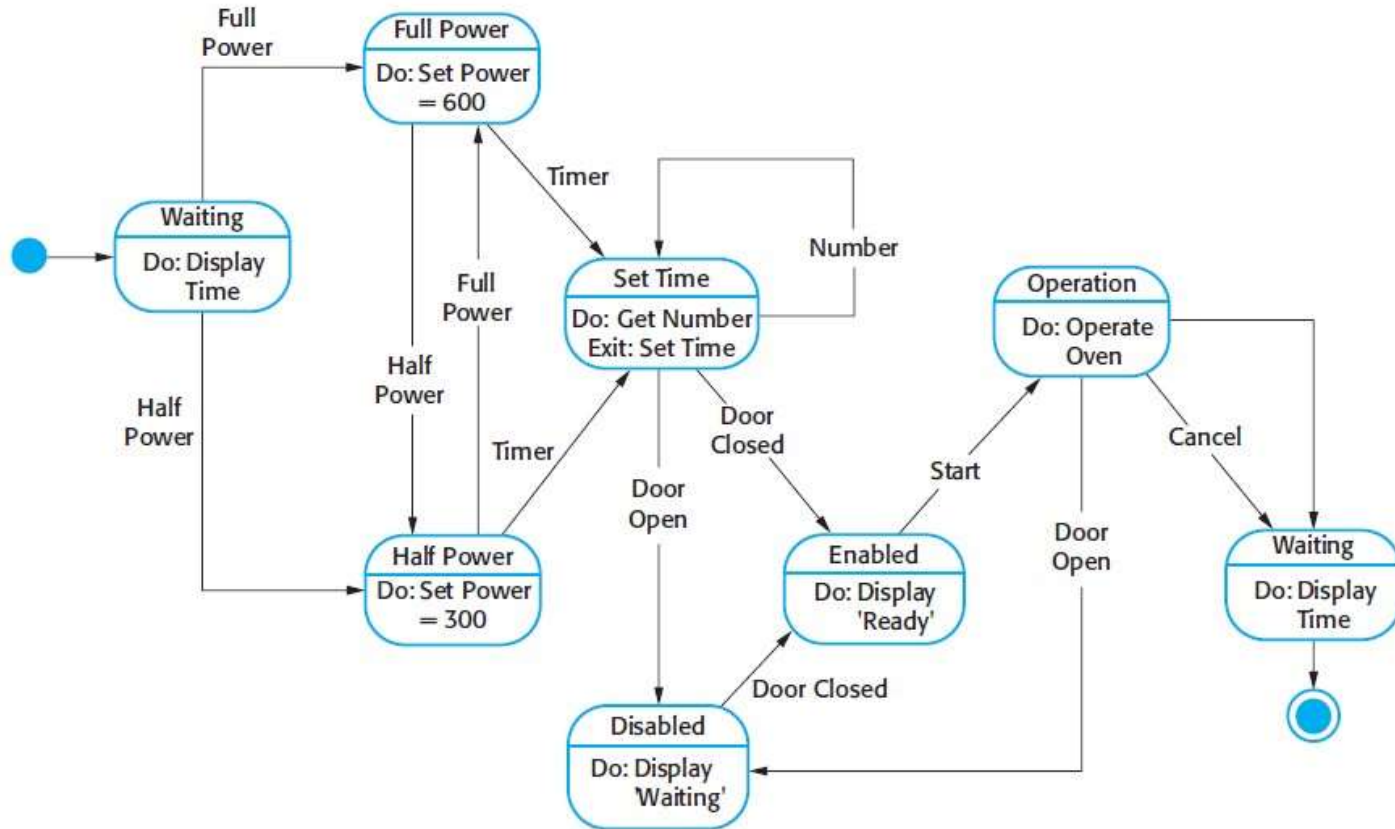
**Estado Final**

# Modelado Dirigido por Eventos (Comp.)

## Ejemplo

- Considere un microondas simple que tiene un interruptor para seleccionar potencia total o media, un teclado numérico para ingresar el tiempo de cocción, un botón de inicio/parada y una pantalla alfanumérica.
- Supongamos una secuencia de acciones al usar el microondas:
  1. Seleccione el nivel de potencia.
  2. Ingrese el tiempo de cocción usando un teclado numérico.
  3. Presione *Start* y la comida se cocinará por el tiempo dado.
- El horno no debe funcionar cuando la puerta está abierta y, al finalizar la cocción, suena un zumbador. El horno tiene una pantalla alfanumérica para mostrar varias alertas y mensajes de advertencia.

# Modelado Dirigido por Eventos (Comp.)



# Modelado Dirigido por Eventos (Comp.)

Estado	Descripción
Esperando	El horno está esperando la entrada. La pantalla muestra la hora actual.
Media Potencia	La potencia del horno está configurada en 300 vatios. La pantalla muestra 'Media Potencia'.
Máxima Potencia	La potencia del horno está configurada en 600 vatios. La pantalla muestra 'Máxima Potenciar'.
Configurar el tiempo	El tiempo de cocción se establece en el valor de entrada del usuario. La pantalla muestra el tiempo de cocción seleccionado y se actualiza a medida que se configura la hora.
Deshabilitado	La operación del horno está desactivada por seguridad. La luz interior del horno está encendida. La pantalla muestra 'Deshabilitado'.
Habilitado	La operación del horno está habilitada. La luz interior del horno está apagada. La pantalla muestra 'Listo para cocinar'.
En Funcionamiento	Horno en funcionamiento La luz interior del horno está encendida. La pantalla muestra la cuenta regresiva del temporizador. Al finalizar la cocción, el zumbador suena durante cinco segundos. La luz del horno está encendida. La pantalla muestra "Completo" mientras suena el zumbador.

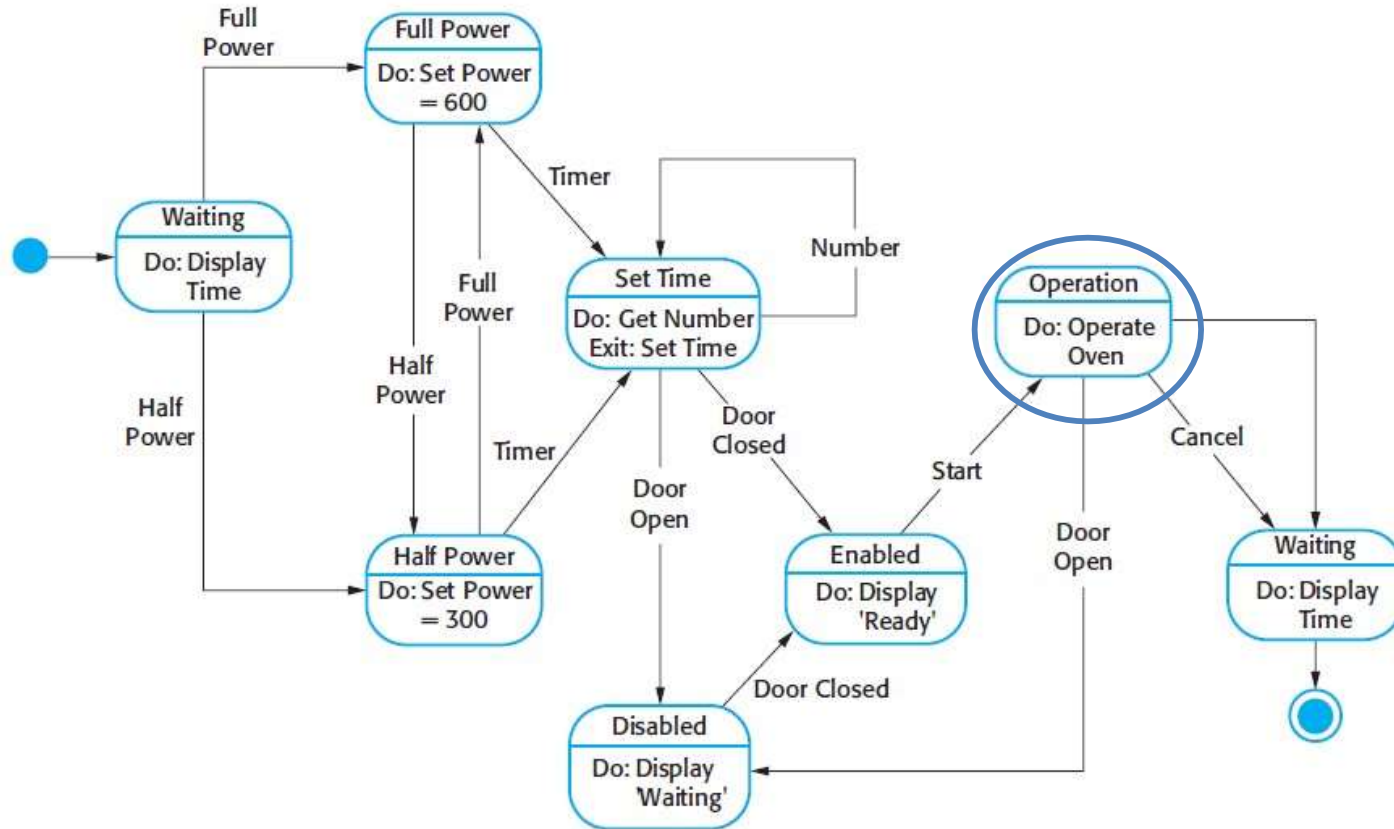
# Modelado Dirigido por Eventos (Comp.)

Estímulo	Descripción
Half Power	El usuario ha presionado el botón de la mitad de la potencia.
Full Power	El usuario ha presionado el botón de encendido completo.
Temporizador	El usuario ha presionado uno de los botones del temporizador
Número	El usuario ha presionado una tecla numérica.
Puerta Abierta	El interruptor de la puerta del horno no está cerrado.
Puerta Cerrada	El interruptor de la puerta del horno está cerrado.
Inicio	El usuario ha presionado el botón de Inicio.
Cancelar	El usuario ha presionado el botón Cancelar.

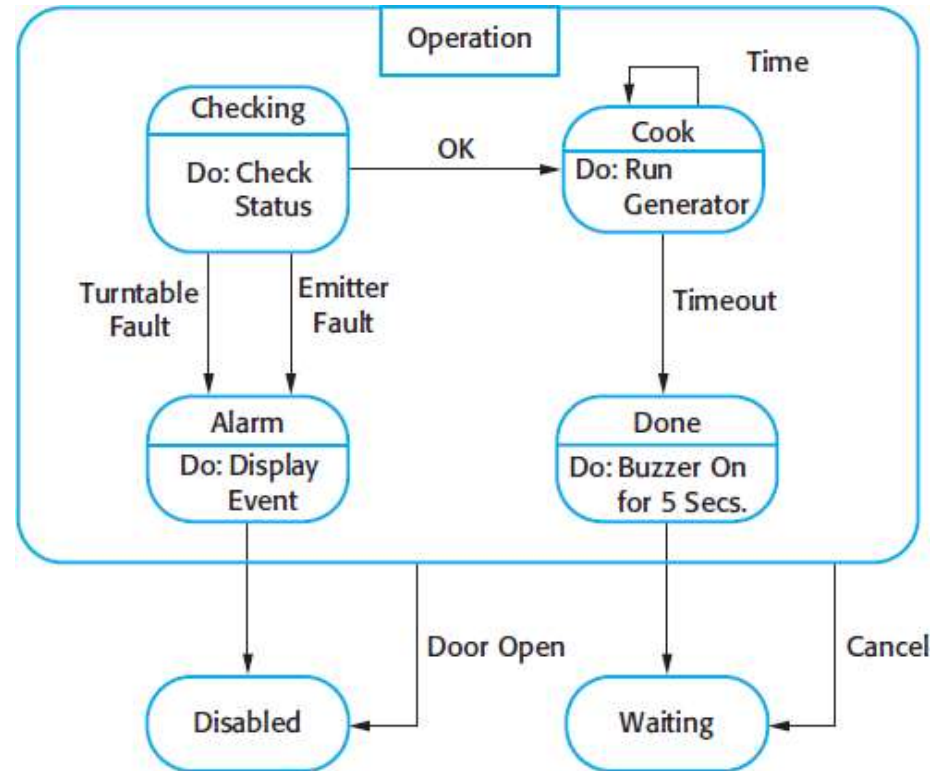
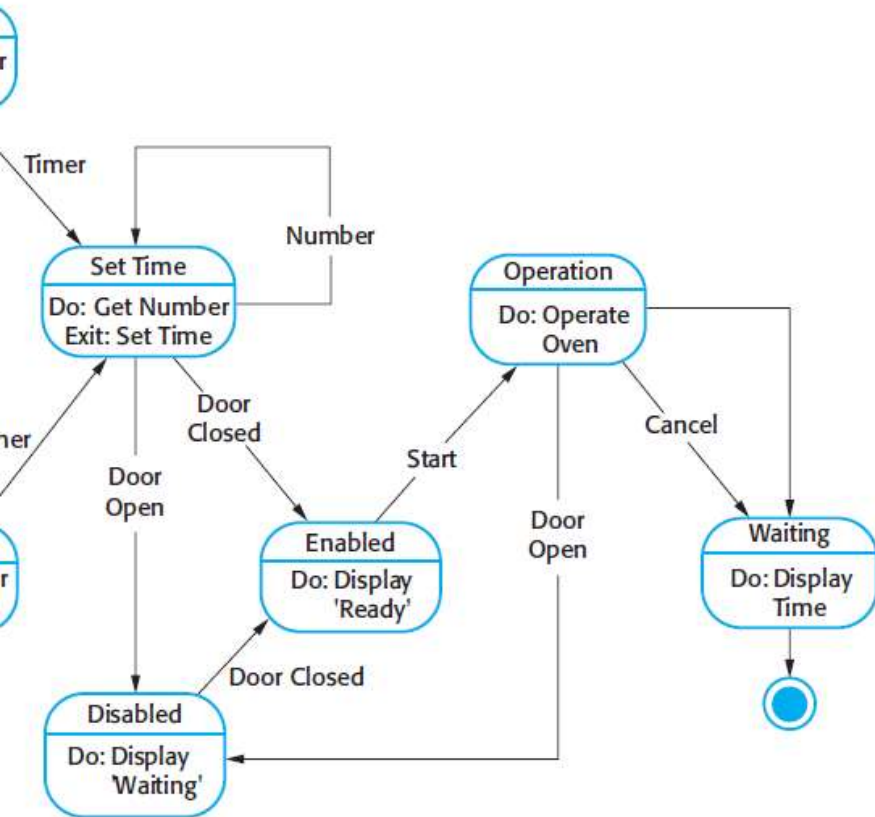
# Modelado Dirigido por Eventos (Comp.)

- El problema con el modelado basado en estado es que la cantidad de estados posibles aumenta rápidamente.
- Para modelos de sistemas grandes, se necesita ocultar detalles en los modelos.
- Una forma de ocultar detalles es usar la noción de un **super estado** que encapsula una cantidad de estados separados.
- Este **super estado** se ve como un estado único en un modelo de alto nivel, pero luego se expande para mostrar más detalles en un diagrama separado.

# Modelado Dirigido por Eventos (Comp.)



# Modelado Dirigido por Eventos (Comp.)





# Ejercicio

Construya un diagrama de secuencia que modele los comportamientos descritos a continuación:

- Una persona pide delivery de comidas. Al solicitar el servicio puede seleccionar un restaurante de aquellos que suele frecuentar. Una vez seleccionado el mismo, se le permite seleccionar un plato del menú a partir de sus preferencias en dicho restaurante.
- Al caso anterior agregarle el comportamiento de que la persona puede seleccionar en un único delivery una serie de platos que son provistos por distintos restaurantes.

# Material Bibliográfico

- Ian Sommerville. 2010. Software Engineering (9th ed.). Addison-Wesley Publishing Company, USA.
- Cadle, J., & Yeates, D. (Eds.). 2004. *Project management for information systems*. Pearson education.
- Epstein, D., & Maltzman, R. 2013. *Project workflow management: a business process approach*. J. Ross Publishing.

# Recordemos que...

- Fecha límite de confirmación de comisiones e ideas del proyecto:

**Hoy!**

- Presentación de la idea del proyecto

**Martes 03/09**

# PRÓXIMA CLASE

---

Diseño de Software