Programación Orientada a Objetos

Fundamentos

El proceso de abstracción

Las aplicaciones de software típicas, modelan el mundo real. El **mundo real es complejo** a simple vista y, cuando se lo observa con más detalle, el nivel de complejidad crece.

¿cómo modelamos este mundo tan complejo?

Los humanos entendemos al mundo, construyendo modelos mentales de partes del mismo. Un modelo mental es una **visión simplificada de cómo las cosas funcionan y cómo podemos interactuar con ellas**.
La abstracción es uno de los mecanismos que los humanos utilizamos para combatir la complejidad La orientación a objetos, maneja la complejidad de los problemas del mundo real, abstrayendo su conocimiento y encapsulándolo en objetos **=>** es clave en el desarrollo de software.

¿Cuál es el objetivo buscado por la programación orientada a objetos?

Organizar los **datos del programa y el procesamiento asociado a ellos**, en entidades coherentes, llamadas **objetos**. Cada objeto abstrae un dato del programa y lo que puede hacerse sobre él.

Programación Orientada a Objetos
Fundamentos

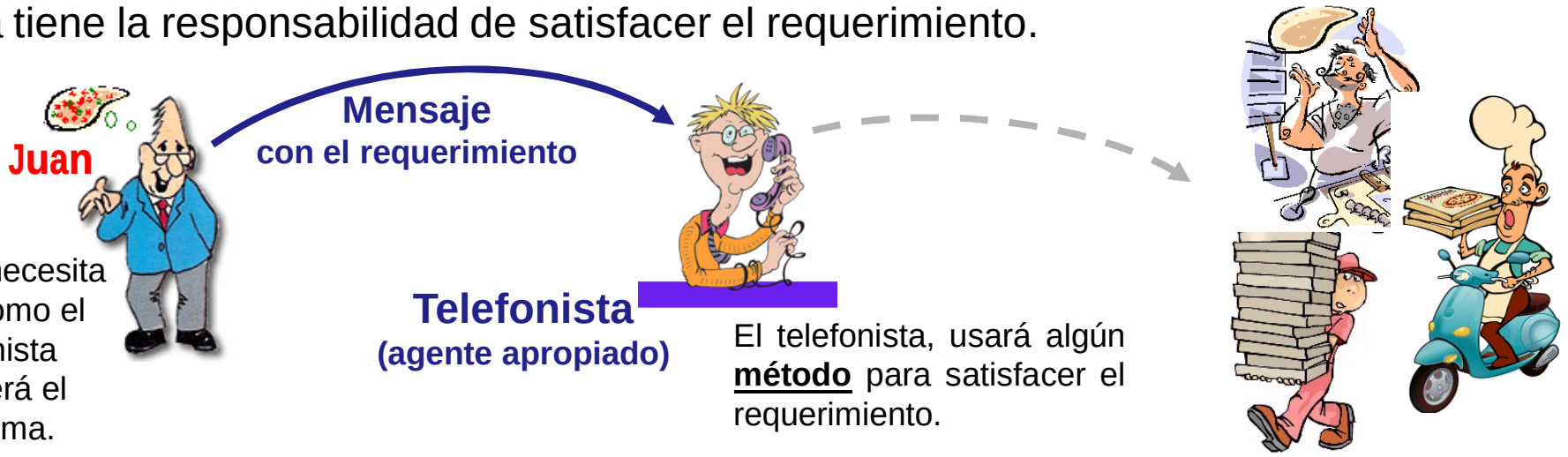
Definición de un problema

Supongamos que Juan quiere hacer un pedido de pizzas para que se las entreguen en su domicilio.

¿qué hace?

Consigue el teléfono de la pizzería y llama. Lo atiende un telefonista, le hace el pedido deseado, indicándole la cantidad y tipos de pizzas que desea y el domicilio a donde debe enviarse el pedido **=> se resolvió el problema**.

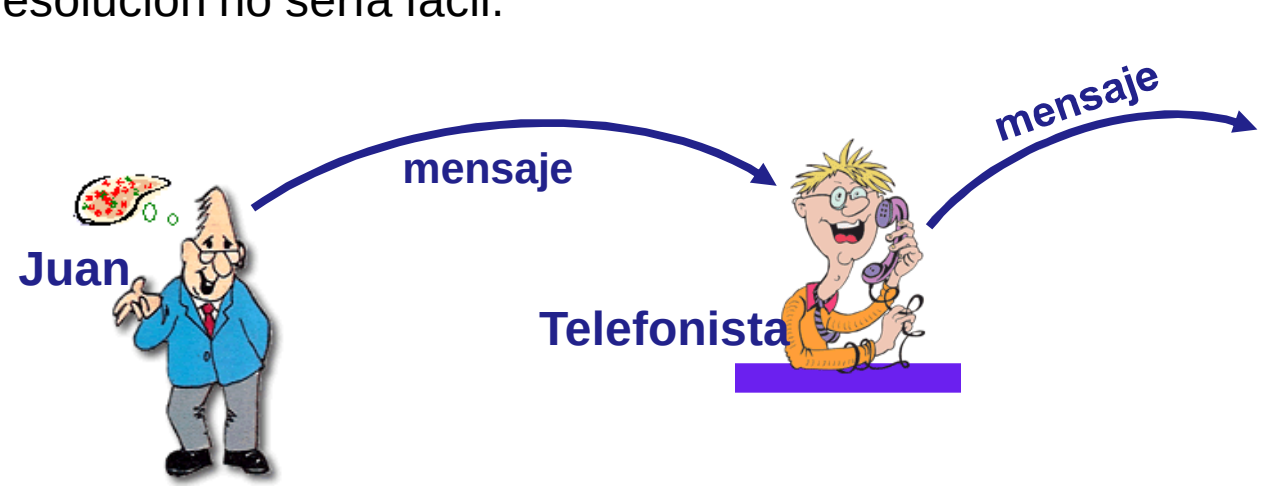
Juan se comunicó con el **Telefonista**, y le pasó un **mensaje** con el requerimiento. El telefonista tiene la responsabilidad de satisfacer el requerimiento.

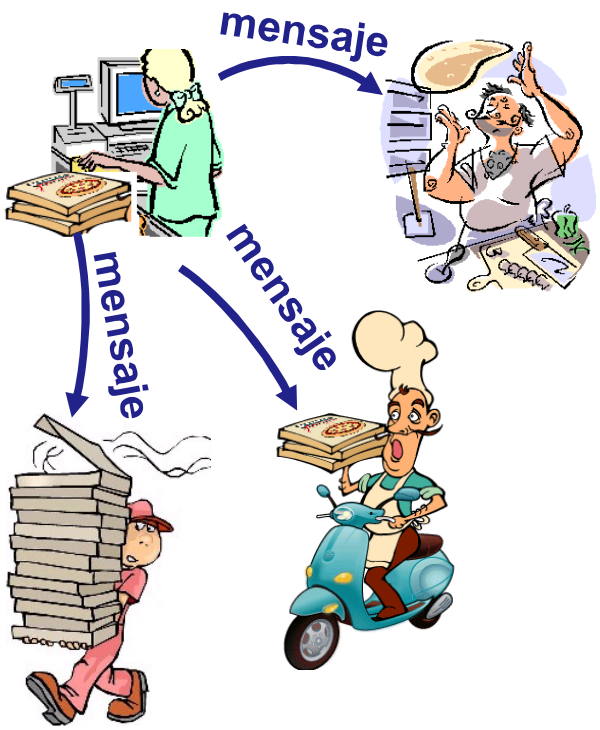


Programación Orientada a Objetos
Fundamentos

agentes u objetos

La resolución del problema, requiere de la ayuda de otros individuos. Sin su colaboración, la resolución no sería fácil.





1º principio de programación orientado a objetos

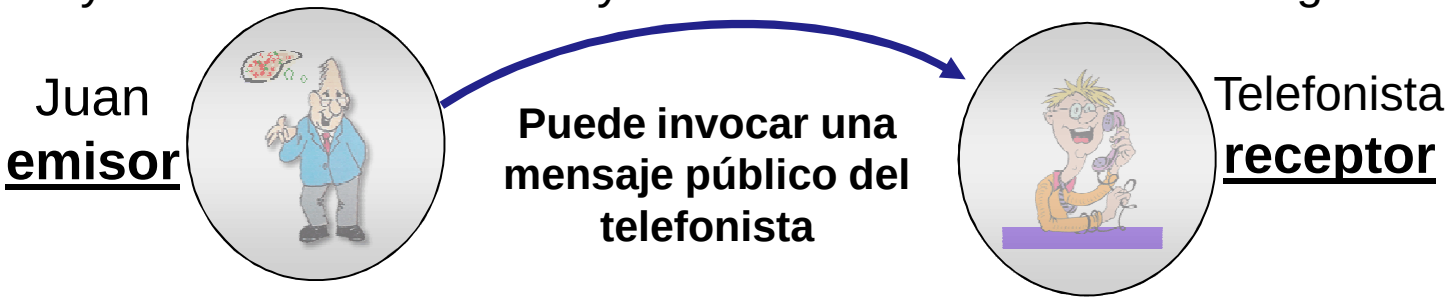
Un programa **orientado a objetos**, está organizado como una comunidad de agentes interactuando, llamados **objetos**. Cada objeto cumple un rol. Cada objeto provee un servicio o ejecuta una acción, que es usada por otros miembros de la comunidad.

Programación Orientada a Objetos
Fundamentos

Ocultamiento de información

Un objeto (cliente, telefonista, repartidor, etc.) es una entidad que contiene información y operaciones relacionadas, que tiene sentido agrupar (empaquetar). Este concepto, en el contexto de POO es conocido como **encapsulamiento**.

Comúnmente los objetos son como cápsulas opacas, con una interfaz pública y una representación privada. Este concepto se conoce como **ocultamiento de información** (***information hiding***). Permite eliminar de la vista cierta información propia del objeto, logrando mayor nivel de abstracción y facilitando los cambios del código.



2º principio de programación orientado a objetos

El **encapsulamiento y el ocultamiento de información** se complementan, para aislar las diferentes partes de un sistema, permitiendo que el código sea modificado, extendido y que se puedan corregir errores, sin el riesgo de producir efectos colaterales no intencionados.
Los objetos ponen en práctica estos dos conceptos:

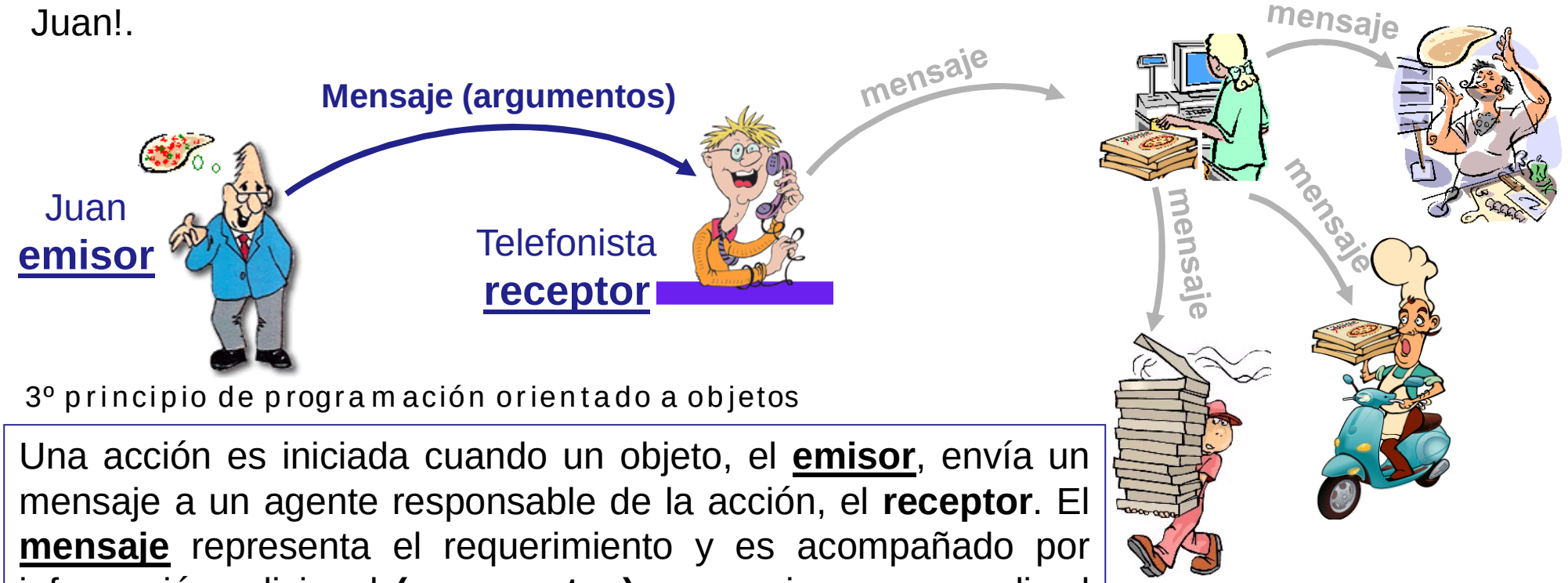
1. Se abstrae la funcionalidad y la información relacionada y se encapsulan en un objeto.
2. Se decide que funcionalidad e información, podrá ser requerida por otros objetos y el resto se oculta.

Programación Orientada a Objetos

Fundamentos

mensajes y métodos

Juan hace un primer requerimiento al telefonista, quien hace otro requerimiento que conduce a más y más requerimientos, hasta que se resuelve el problema: la pizza le llega a Juan!.



3º principio de programación orientado a objetos

Una acción es iniciada cuando un objeto, el **emisor**, envía un mensaje a un agente responsable de la acción, el **receptor**. El **mensaje** representa el requerimiento y es acompañado por información adicional **(argumentos)** necesaria para cumplir el requerimiento. El receptor es el objeto a quien se le envía el mensaje. El receptor en respuesta al mensaje ejecutará un conjunto de acciones o método para satisfacer el requerimiento.

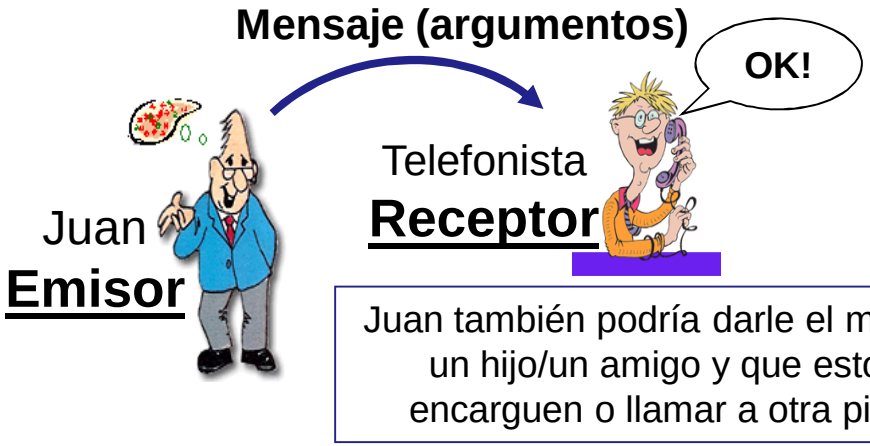
Programación Orientada a Objetos
Fundamentos

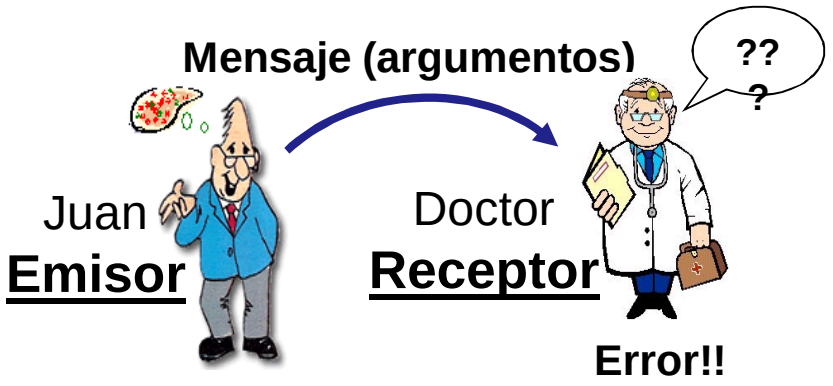
mensajes y métodos vs. llamadas a procedimiento

Existen 2 distinciones importantes:

(1) En un mensaje, siempre hay designado un receptor para aquel mensaje; el receptor es algún objeto, al cual se le envía un mensaje. Cuando se llama a un procedimiento, NO hay receptor.

(2) La interpretación del mensaje (el método usado para responder al mensaje) es determinado por el receptor y podría variar para diferentes receptores.





Programación Orientada a Objetos
Fundamentos

Clases e instancias

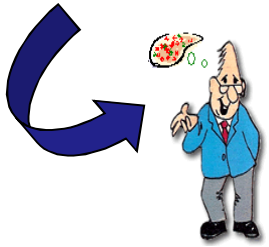
¿qué sabe el telefonista acerca de Juan?

El telefonista sabe que quien está llamando, **es un cliente** y puede asumir, por ello, ciertas cosas. Cree por ejemplo, que Juan, su cliente, le abonará las pizzas cuando las reciba en su domicilio => se comporta como un cliente. Esto es porque Juan pertenece a una **categoría** o **clase** que podríamos llamar **Cliente** y el telefonista espera que Juan siendo una **instancia** de esta categoría, se ajuste a un patrón.

4º principio de programación orientado a objetos

Todos los objetos son instancias de una clase. El método invocado por un objeto en respuesta a un mensaje es determinado por la clase del objeto receptor.

Una clase es un molde a partir de la cual se crean instancias con las mismas características y comportamiento.



Juan, es una instancia de la clase Cliente.

Programación Orientada a Objetos
Fundamentos

Clases e instancias

Una **instancia u objeto** es una entidad de software que combina un **estado/datos** y **comportamiento/métodos**.

Juan, es una instancia de la clase Cliente, es un objeto de tipo Cliente

Todos los objeto de tipo Cliente
- **estado:** #cliente, domicilio de entrega, deuda.
- c**omportamiento:** dar#Cliente, abonar pedido, ...

- El **estado** de un objeto es todo lo que el objeto **conoce de si mismo** y, el **comportamiento** es todo lo que el objeto **puede hacer**.
- Un **objeto** mantiene su **estado** en **variables** y su **comportamiento** está implementado en los **métodos** de la clase a la que pertenece.

Programación Orientada a Objetos
Fundamentos

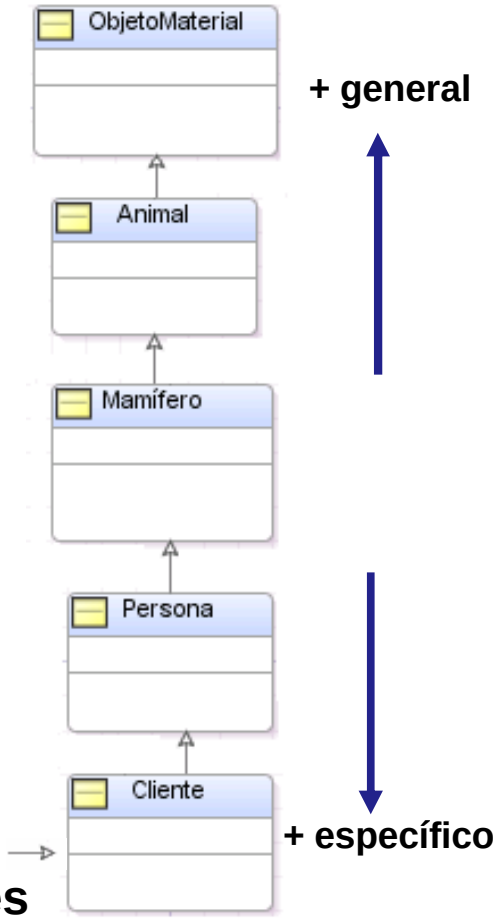
Herencia

¿qué más sabe el telefonista acerca de Juan?

El podría pensar u organizar el conocimiento en términos de una **jerarquía de categorías**. Juan, es un cliente, es una persona especial. Al conocer que es una persona o humano, sabe que es bípedo y como también es un mamífero sabe que tiene pelo y como es animal, sabe que respira oxígeno.

5º principio de programación orientado a objetos

El conocimiento de una categoría más general, es también aplicable a una categoría mas específica y se denomina **herencia.**



Las clases pueden ser organizadas en jerarquías de herencia donde, las clases hijas o **subclases**, heredarán **estado y comportamiento** de las clases que se encuentran más arriba en la jerarquía, llamadas **superclases**.
Las subclases pueden agregar nuevas variables y métodos y pueden cambiar el comportamiento de los métodos heredados.

Juan, es un Cliente