

Programación Orientada a Objetos



Programación Orientada a Objetos

¿Qué datos y funcionalidades maneja una aplicación de estas características?



Algunos datos . . .

- Necesitamos disponer de **contactos** y de grupos de contactos.
- **Mensajes (texto, emoticon, imagen/foto, voz)**

Algunas funcionalidades . . .

- Crear/Editar/Borrar/Bloquear Contacto
- **Enviar mensaje**
- Armar/Editar/Borrar grupo
- Salir de un grupo

Programación Orientada a Objetos

¿Qué tipos de objetos manipulamos?

Contacto

nombre
imagen
numero
estado
contactos (conjunto
de **Contacto** o
Grupo)

agregar/editar/borrar contacto
armar/editar/borrar grupo
enviar mensaje
...

Grupo

nombre
imagen
participantes
(conjunto de
Contacto)

enviar mensaje
agregarParticipante
...

Mensaje

origen, destino
Contenido (texto,
imagen, sonido, video)
fechaHora salida
fechaHora recibido
fechaHora leído

mostrar
...

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?

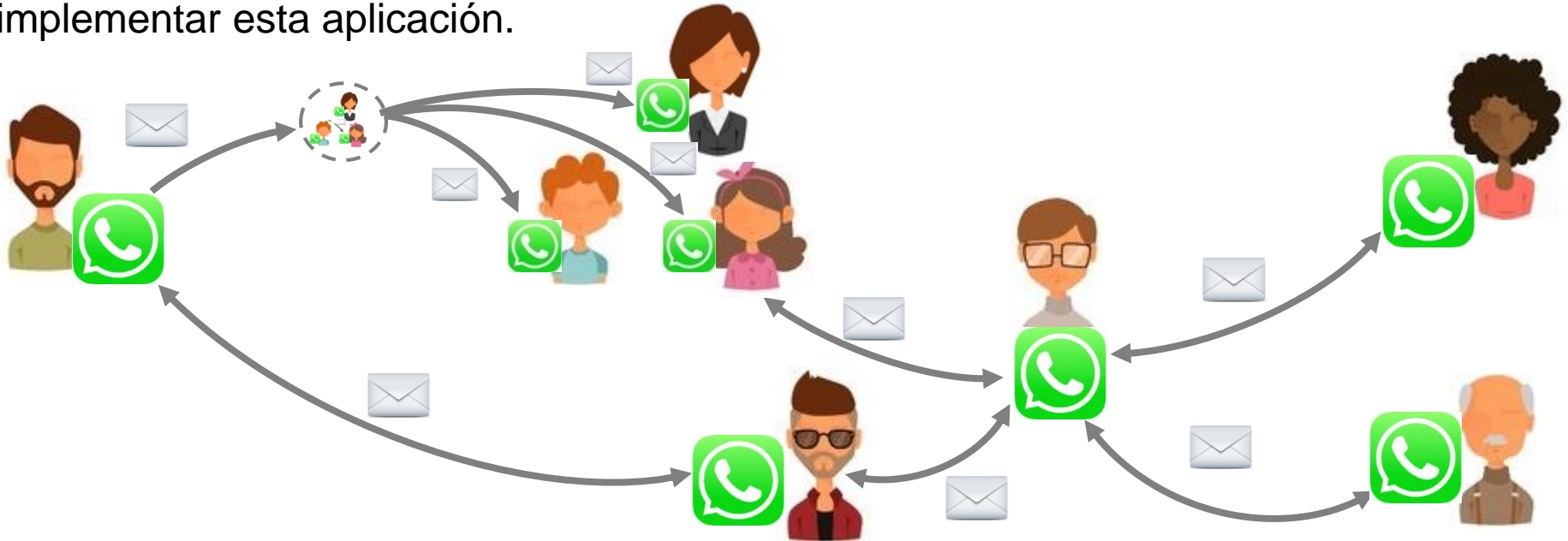
Organizar los datos del programa y el procesamiento asociado a ellos, en entidades coherentes, llamadas **objetos**. Cada objeto abstrae un dato del programa y lo que puede hacerse sobre él.

Programación Orientada a Objetos

Fundamentos

agentes u objetos

Encontramos que algunos objetos como mensajes, contactos, grupos, etc. son necesarios para implementar esta aplicación.



1º principio de programación orientado a objetos

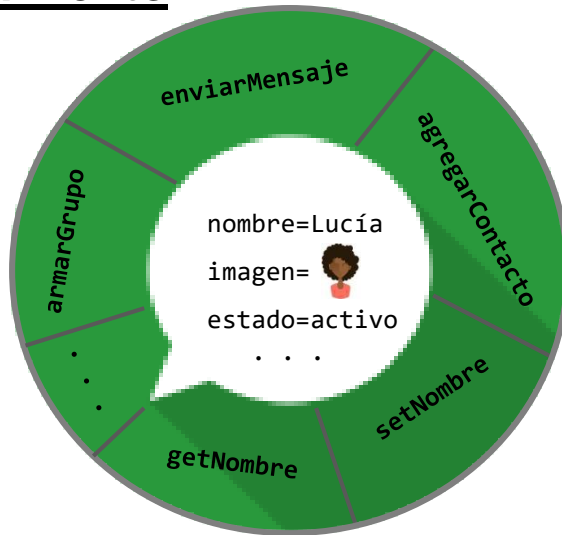
Un programa **orientado a objetos**, está organizado como una comunidad de agentes interactuando, llamados **objetos**. Cada objeto cumple un rol. Cada objeto provee un servicio o ejecuta una acción, que es usada por otros miembros de la comunidad.

Programación Orientada a Objetos

Fundamentos

Ocultamiento de información

Un objeto 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:

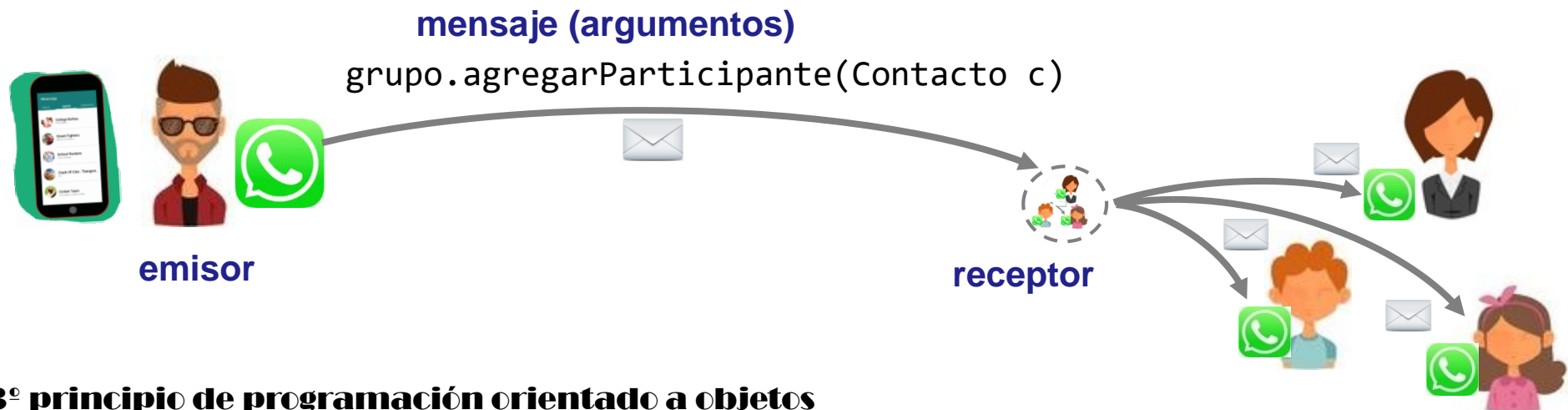
1. Se abstrae la funcionalidad y la información relacionada y se encapsulan en un objeto.
2. Se decide que funcionalidad e información, podrá ser requerida por otros objetos y el resto se oculta.

Programación Orientada a Objetos

Fundamentos

mensajes y métodos

En general a través de la Interfaz de Usuario Gráfica se generan eventos que envían mensajes a ciertos objetos, los cuales a su vez envían mensajes a otros objetos, hasta que comunitariamente completan la tarea.



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 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:

- Cuando se envía un mensaje, siempre hay 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.

`grupo.agregarParticipante(Contacto c)`

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

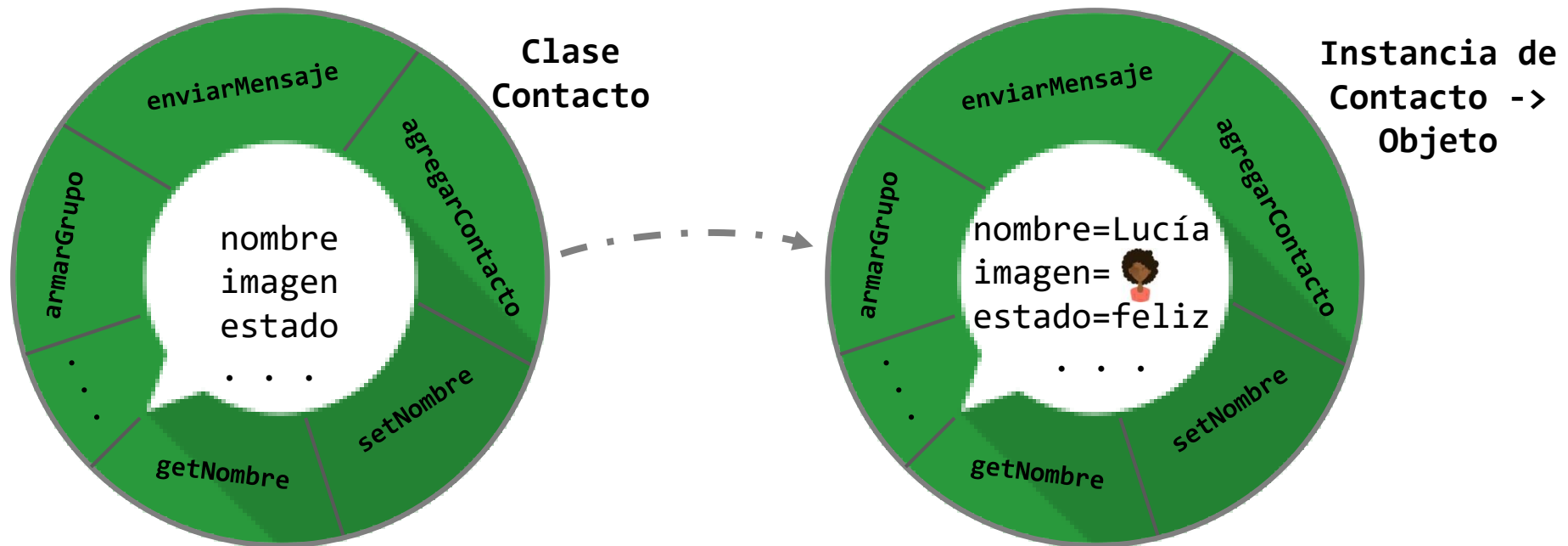


Programación Orientada a Objetos

Fundamentos

Clases e instancias

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



4º principio de programación orientado a objetos

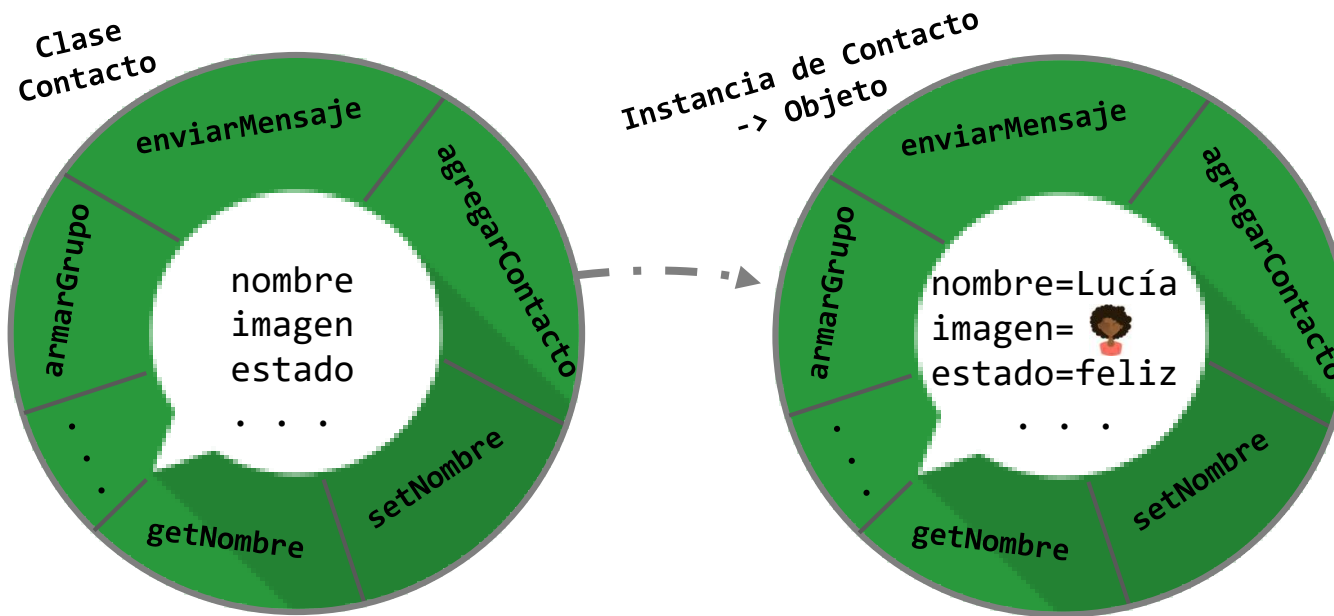
Todos los objetos son instancias de una clase. El método invocado por un objeto en respuesta a un mensaje es determinado por la clase del objeto receptor.

Programación Orientada a Objetos

Fundamentos

Clases e instancias

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

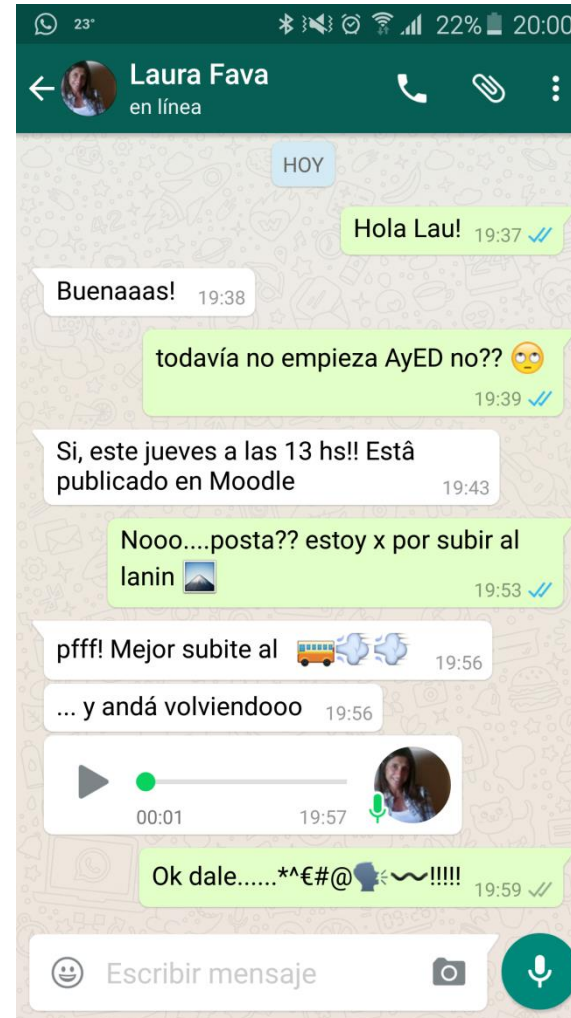


Todos los objetos de tipo **Contacto** tendrán un nombre, una imagen, un estado, etc. pero cada uno tendrá un valor particular en esos atributos.

- El **estado** de un objeto es todo lo que el objeto **conoce de sí 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

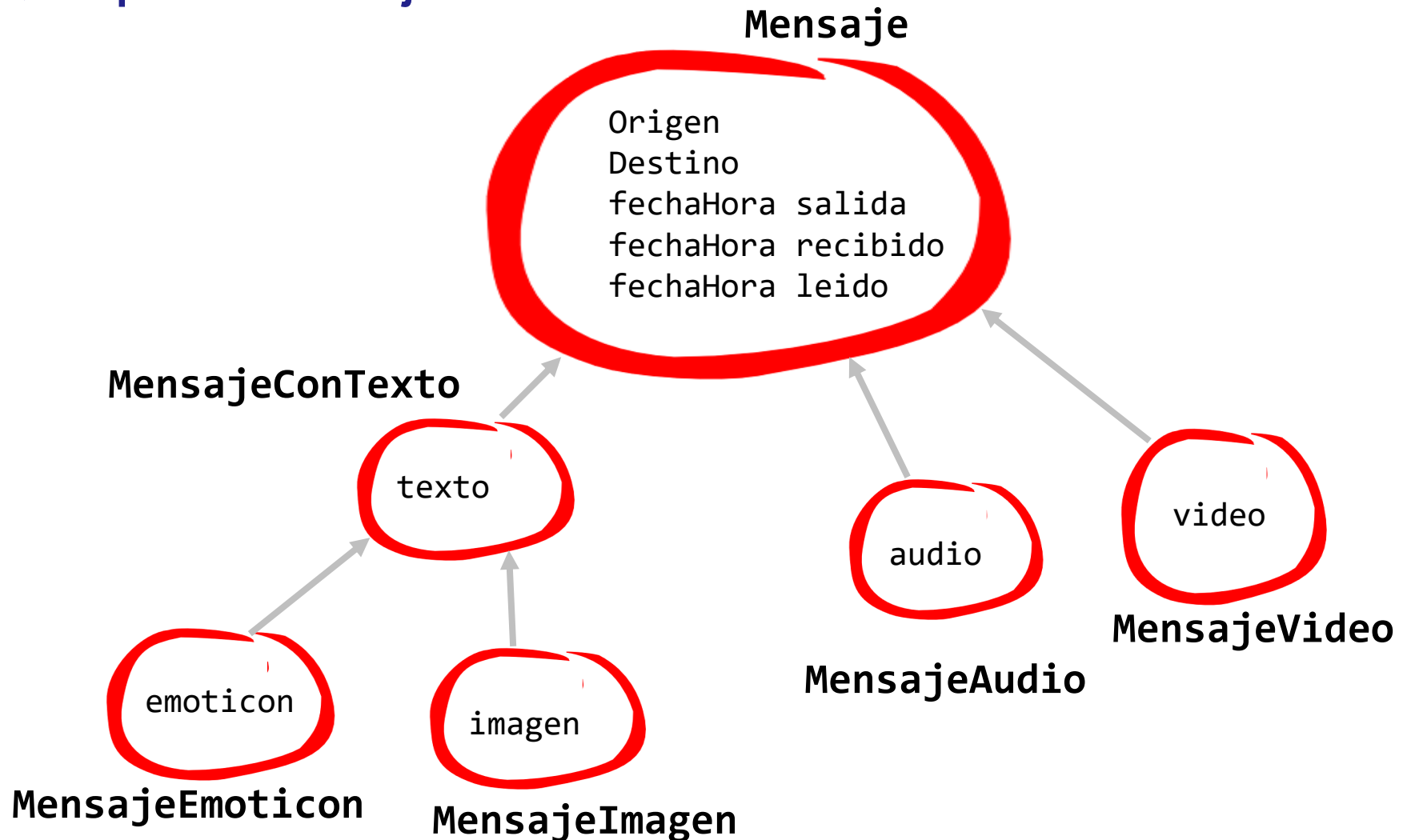
¿Qué tipos de mensajes tenemos?



Programación Orientada a Objetos

Fundamentos

¿Qué tipos de mensajes tenemos?



Programación Orientada a Objetos

Fundamentos

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.



La programación orientada a objetos (POO) permite a las clases expresar similitudes entre objetos que tienen características y comportamiento común. Estas similitudes pueden expresarse usando **herencia**.

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.