



# Tecnologías de Programación

## Paradigma Orientado a Objetos

# Objetivos

- Generales:
  - Aplicar el paradigma con orientación a objetos en la resolución de problemas.
  - Dominar el vocabulario técnico del paradigma orientado a objetos.
- Específicos:
  - Comprender los fundamentos del paradigma con orientación a objetos utilizados por los lenguajes de programación.
  - Conocer el modelo formal o semiformal subyacente del paradigma y la forma en que el mismo es incorporado en un lenguaje de programación.
  - Resolver problemas a través del modelo de programación orientado a objetos

# Generalidades

- Está basado en la encapsulación del estado (con atributos) y el comportamiento (métodos)
- Procura favorecer un buen diseño modular (pocas interfaces, interfaces pequeñas y explícitas, y ocultamiento de información).

# Ventajas

Algunas de las ventajas de la programación orientada a objetos son los siguientes:

- Los componentes se pueden reutilizar.
- Facilidad de mantenimiento y modificación de los objetos existentes.
- Una estructura modular clara se puede obtener, la cual no revelará el mecanismo detrás del diseño.
- Se proporciona un buen marco que facilita la creación de rica interfaz gráfica de usuario aplicaciones (GUI).
- Se acopla bien a la utilización de bases de datos, debido a la correspondencia entre las estructuras.

# Desventajas

- Limitaciones del programador: Es posible el programador desconozca algunas características del paradigma y de hecho siga utilizando el paradigma estructurado.
- No hay una forma única de resolver los problemas. Esto puede llevar a que diferentes interpretaciones de la solución planteada emerjan.
- Se requiere una documentación amplia para determinar la solución planteada

# 1 - Conceptos introductorios

- Definición fundamental del paradigma:

“Un programa es un conjunto de objetos que colaboran entre sí e enviándose mensajes”

**OBJETOS** + **MENSAJES** = **PROGRAMA**

# 1 - Conceptos introductorios

- OBJETO:

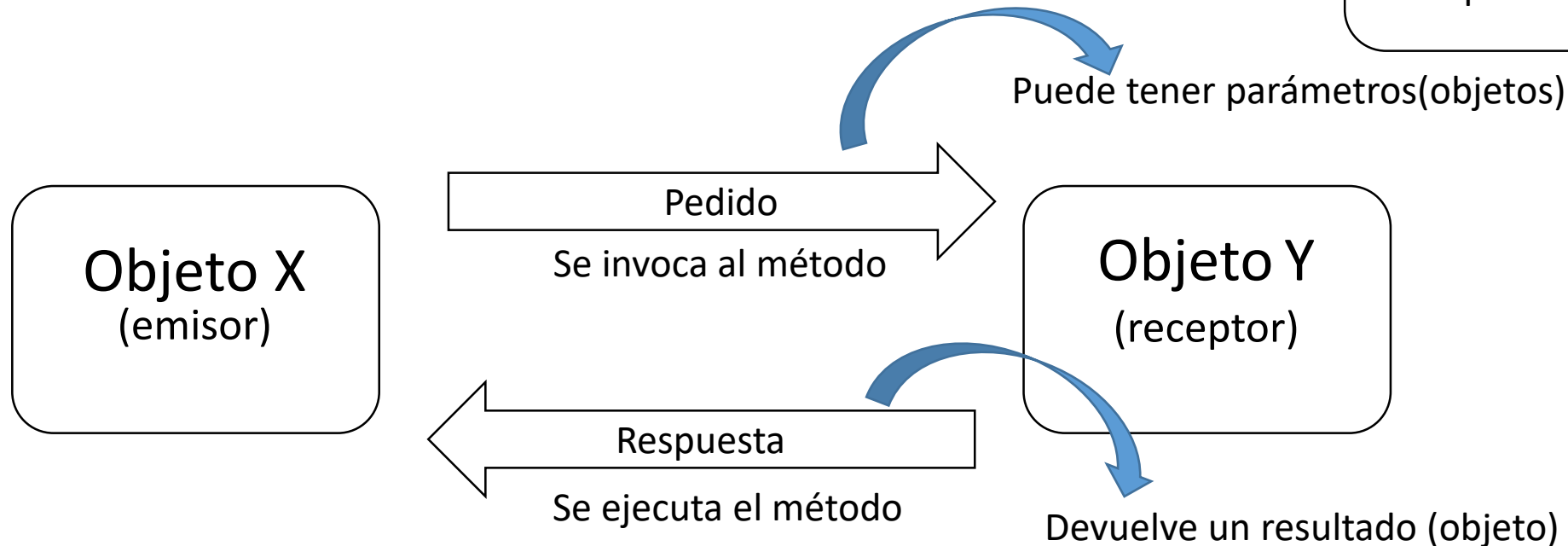
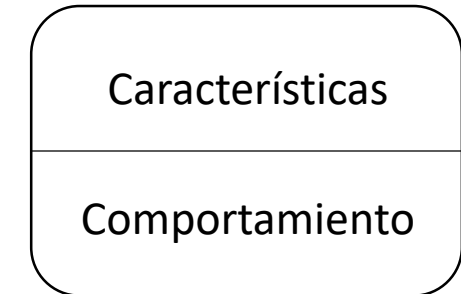
“Unidad de software de la cual lo que me interesa es: qué le puedo preguntar y/o pedir y a qué otros objetos conoce. Los objetos responden a los pedidos interactuando con los otros objetos que conoce.”

**PENSAR EN OBJETOS**

Un objeto es una abstracción conceptual del mundo real que se puede traducir a un lenguaje computacional o de programación OO

# 1 - Conceptos introductorios

- OBJETO:  
Un objeto tiene características y comportamiento  
A través de mensajes se relaciona con otros objetos





## 2 - Características de la POO

- Abstracción
- Modularidad
- Encapsulamiento
- Principio de ocultación
- Cohesión y acoplamiento
- Herencia
- Polimorfismo
- Recolección de basura

## 2 - Características de la POO

- Abstracción
  - Denota las características esenciales de un objeto donde se capturan sus comportamientos.
  - Permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real. Por medio de ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se necesita abordar y resolver.

## 2 - Características de la POO

- Abstracción

Características genéricas

Características esenciales según el dominio

### Persona



Documento, nombre (identificación)  
Teléfono, domicilio y otros (contacto)  
Altura, peso y otros (contextura física)  
Estudios secundarios y otros (formación)

Persona en la Universidad  
Persona en un equipo de fútbol  
Persona en la empresa



## 2 - Características de la POO

- Modularidad
  - Nos permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes. Estos módulos se pueden compilar por separado, pero tienen conexiones con otros módulos

## 2 - Características de la POO

- **CARACTERÍSTICAS: Encapsulamiento**
  - Nos permite reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto aumenta la cohesión de los componentes del sistema.
  - Nos provee el mecanismo para empaquetar datos y procedimientos en los objetos. Los objetos sustentan el encapsulado.

## 2 - Características de la POO

- Principio de ocultación
  - Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz (protocolo) a otros objetos que especifica cómo pueden interactuar con los objetos de la clase.
  - El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado.
  - Esto asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas.

## 2 - Características de la POO

- Cohesión y acoplamiento
  - La cohesión mide las responsabilidades asignadas a cada objeto, mientras que el acoplamiento mide las relaciones entre los objetos.
  - En el Modelado Orientado a Objetos, lo que se busca es tener ALTA cohesión (cada objeto se responsabilice por una sola cosa) y BAJO Acoplamiento (Poca o nula interdependencia)

## 2 - Características de la POO

- Herencia
  - Las clases se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple.
  - Nos provee el mecanismo para diseminar la información definida en clases genéricas a otras clases consideradas casos especiales de las anteriores. Las clases sustentan la herencia.



## 2 - Características de la POO

- Polimorfismo
  - El polimorfismo es originalmente una palabra griega que significa la capacidad de adoptar múltiples formas.
  - Comportamientos diferentes asociados a objetos distintos pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando.
  - Nos da la habilidad de implementar el mismo mensaje en formas diferentes en objetos diferentes. Los mensajes sustentan el polimorfismo.

## 2 - Características de la POO

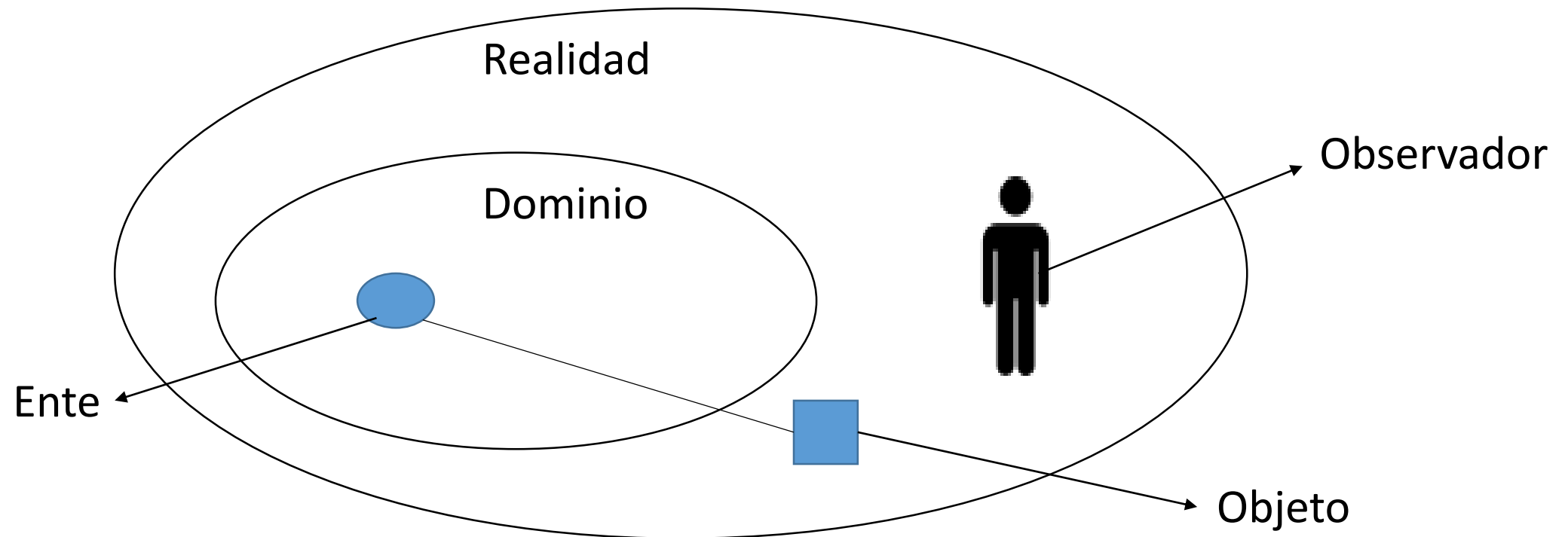
- Recolección de basura
  - La recolección de basura o garbage collector es la técnica por la cual el entorno de objetos se encarga de destruir automáticamente, y por tanto desvincular la memoria asociada, los objetos que hayan quedado sin ninguna referencia a ellos.
  - Esto significa que el programador no debe preocuparse por la asignación o liberación de memoria, ya que el entorno la asignará al crear un nuevo objeto y la liberará cuando nadie lo esté usando

# 3 - Objeto

- “Una entidad discreta con un límite bien definido que encapsula el estado y el comportamiento; una instancia de una clase” The UML Reference Manual – Rumbaugh et al. (2004)
- Los empleamos para representar computacionalmente a una entidad de la vida real que tiene propiedades particulares (atributos) y una manera de operar con ellas (métodos) y puede representar:
  - Lo que tiene vida: personas, animales y otros
  - Organizaciones: una cátedra, un equipo de básquet, una empresa y otros
  - Lo que se toca es solo un soporte: una película, un libro, una ley y otros
  - Cosas abstractas: un nombre, una letra, un número y otros

### 3 - Objeto

- Observador: alguien que va a trabajar/interactuar con el objeto. Es alguien que desempeña un rol y los objetos que ve son aquellos con los que precisa interactuar de alguna forma para cumplir su rol.

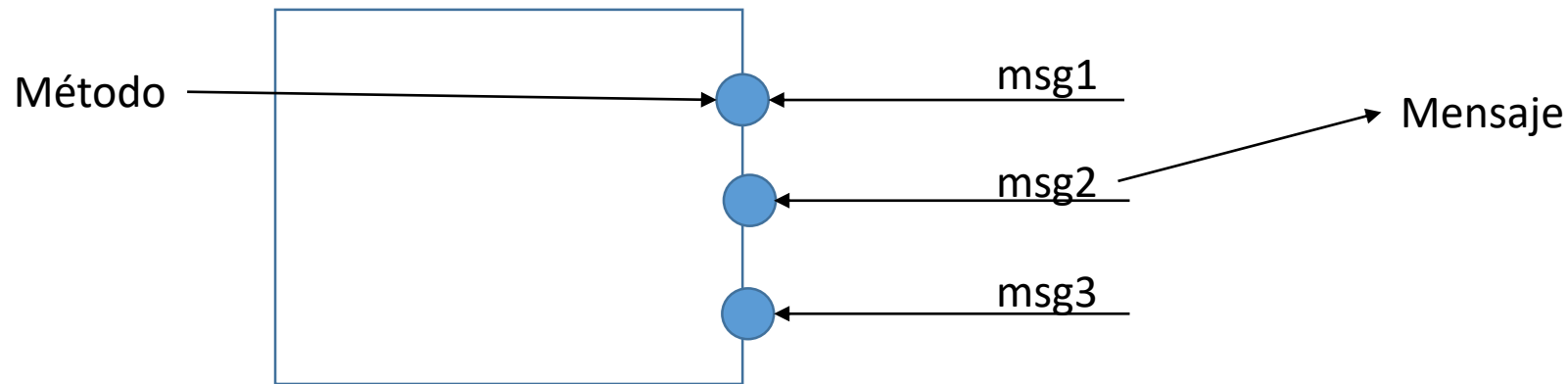


## 4 – Los objetos en el paradigma

- Un objeto debe tener:
  - Abstracción: representación computacional de entes de la realidad y tiene dos características:
    1. Esenciales: hacen que el ente sea lo que es (no puede cambiar)
    2. Accidentales: si el ente no lo tiene o se cambia no deja de ser lo que es
  - Identidad: es la existencia única del objeto en tiempo y espacio.
  - Comportamiento: es el conjunto de mensajes que el objeto puede responder. Este conjunto se llama Protocolo. Los mensajes son el Vocabulario que el objeto es capaz de entender.
  - Capacidad de inspección: visualizar en todo momento quienes son sus colaboradores y eventualmente cambiárselo.

## 5 – Colaboraciones:

- Mensajes y Métodos:
  - Cuando un objeto recibe un mensaje lo único que puede hacer es enviar un mensaje a otro objeto. Esto se llama colaboración.



- Método: es el conjunto de colaboraciones que un objeto tendrá con otros. Le dice al objeto qué hacer con cada mensaje que reciba. Los métodos asociados a un objeto implementan el comportamiento del objeto.

## 5 – Colaboraciones:

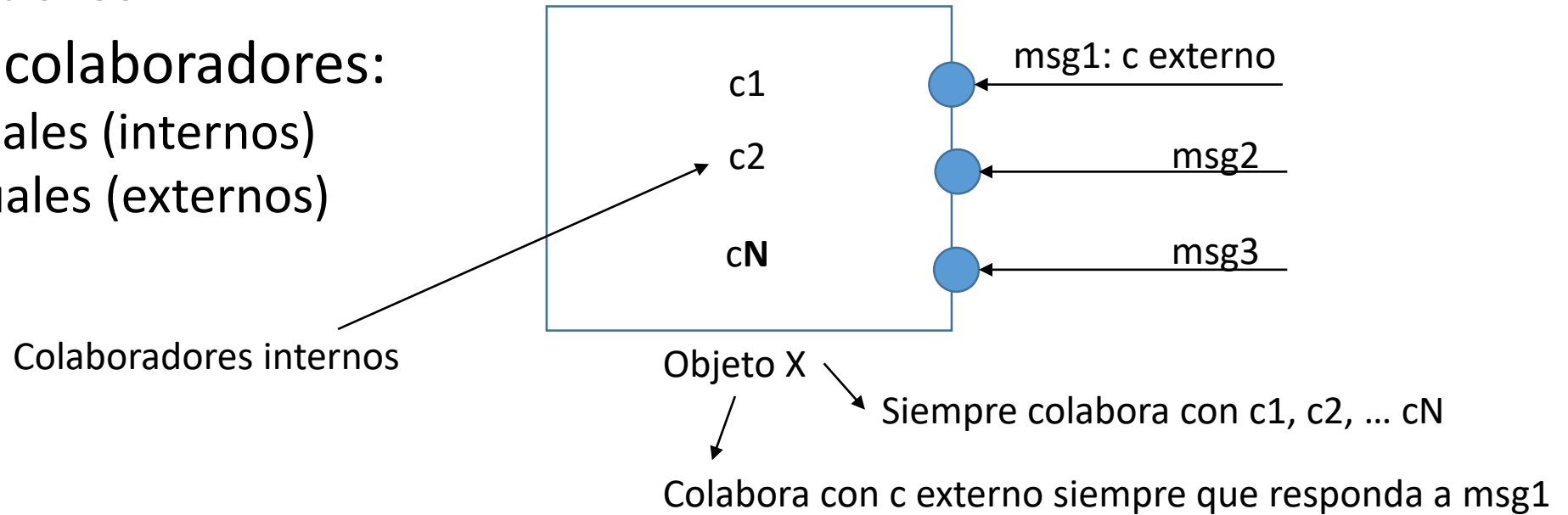
- Cuando se envía un mensaje hay:
  - Un emisor
  - Un receptor
  - Un nombre que identifica el mensaje
- Mensaje: cada una de las formas posibles de interactuar con un objeto, que se identifica por un nombre y puede tener parámetros.
- Envío de Mensajes: cada interacción con un objeto. Tiene emisor, receptor, identificador o selector y eventualmente parámetros (objetos) y un resultado (otro objeto)

## 5 – Colaboraciones:

- Para poder especificar un MÉTODO debo pedir identificar con mis colaboradores.

- Tipos de colaboradores:

- Habituales (internos)
- Eventuales (externos)



Protocolo (o interface) es público

La definición de métodos y colaboradores es privada



## 5 – Colaboraciones:

La definición de métodos y colaboradores es privada



Es interna al objeto , por lo tanto es la implementación del objeto...  
y está encapsulada

¿Puede ser cambiada en cualquier momento?

Si!!! Mientras se mantenga invariante su protocolo (interface)

## 6 – Colaboradores internos:

- Colaborador interno o estado interno: es el conjunto de variables que va a tener un objeto.
- Una “variable” es el nombre de un objeto, entonces lo que me estoy guardando en realidad es el objeto que está nombrando cada variable.
- Una variable no es un nombre de una sección de memoria. En resumen, el estado interno de un objeto: es el conjunto de variables que contiene, y se define al programar el Objeto.

## 7 – Encapsulamiento:

- Permite cambiar la implementación de un objeto y que el usuario del objeto no se entere. Esto es factible porque no se cambia el protocolo; es decir, el conjunto de mensajes que entiende. El usuario del objeto no ve cómo el objeto lo implementa (métodos y estado interno)
- El encapsulamiento consiste en combinar datos y comportamiento en un paquete y ocultar los detalles de la implementación del usuario del objeto.
- Para pensar!!! Qué ventajas tiene?

## 8 – Interpretación Colaboraciones

Un contrato entre emisor y receptor. El emisor debe escribir bien el mensaje y enviar un colaborador externo que sepa hacer cosas que requiere el receptor. Por su lado el receptor debe devolver un objeto que sepa hacer cosas que requiere el emisor.

Errores que pueden ocurrir:

El único error que podría ocurrir es que un objeto reciba un mensaje que no entiende:

- a) Mensaje equivocado al objeto correcto
- b) Mensaje correcto al objeto equivocado. Esto acota y facilita el manejo/descubrimiento de errores

## 9 – Polimorfismo

Dos o más objetos son polimórficos respecto de un conjunto de mensajes si todos ellos pueden responder de manera semánticamente equivalente, aún si su implementación es distinta.

Los objetos polimórficos son intercambiables. Esto otorga enorme flexibilidad en el desarrollo. El acoplamiento entre objetos se hace a través del protocolo, o sea acoplamiento de mensajes.

# 10 – Conclusiones

El paradigma orientado a objetos está basado en la encapsulación de estado y comportamiento en entidades llamadas objetos.

Emplea herencia para reutilizar código