

# Introducción a sistemas y paradigmas

Sitio: Agencia de Habilidades para el Futuro  
Curso: Desarrollo de Sistemas Orientado a Objetos 1º D  
Libro: Introducción a sistemas y paradigmas

Imprimido por: Eduardo Moreno  
Día: miércoles, 19 de marzo de 2025, 23:05

# Tabla de contenidos

## **1. Preguntas orientadoras**

## **2. Introducción**

## **3. Sistemas**

### **3.1. Ejemplo**

### **3.2. ¿Por qué hablar de sistemas?**

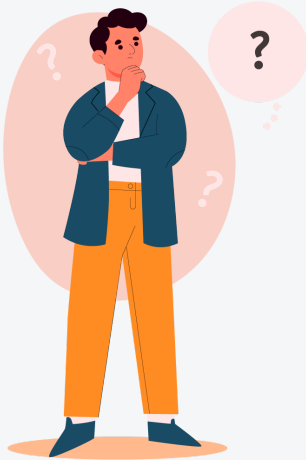
## **4. Paradigma orientado a objetos**

### **4.1. Sistema de negocio en línea**

### **4.2. El proceso de comunicación**



## Preguntas orientadoras



- ¿Para qué nos sirve un sistema?
- ¿Sistema y paradigma significan lo mismo?
- ¿Debemos tener en cuenta los requerimientos del cliente para el desarrollo de un sistema exitoso?
- ¿Formamos parte del desarrollo o nuestra como desarrolladores/as posición es externa?
- ¿Resulta fundamental la relación de los elementos del sistema? ¿O podemos tomar decisiones de forma aislada?



## Introducción

Durante la cursada vas a conocer y usar el lenguaje de programación C# y su entorno de desarrollo, pero antes de comenzar con el lenguaje nos preguntamos:

¿Qué se entiende por **sistemas** ? y ¿por qué dentro de los paradigmas nos quedamos con el **orientado a objetos**?

El propósito de la cursada es que al finalizar puedas presentar un producto final. Para lograrlo:

**Simularemos** un cliente potencial que llega con sus requerimientos, entonces tendremos desarrollar un software para satisfacerlos.

Los conceptos que presentamos a continuación son los básicos para que comprendas el entorno Visual Studio que vamos a usar para programar.

Veamos de que se trata.



La palabra **sistema** tiene muchas acepciones. Y en esta materia, hacemos referencias a los sistemas de información, como por ejemplo para una empresa u organización de gestión.

### Profundicemos en la conceptualización

Podemos definir un sistema como un conjunto de elementos que interaccionan entre sí, orientados a la consecución de un objetivo común.

Un sistema suele estar situado en un entorno o ambiente con el que interactúa, recibe entradas y produce salidas. Entre sus elementos se establece una relación de "funcionamiento" y entre los elementos del sistema y los que interactúan con él se establece una relación de "comportamiento".

Pero, ¿qué significa establecer este tipo de relaciones?

- Las relaciones de funcionamiento se refieren a la manera que interactúan los elementos entre sí, es similar a los vínculos entre los integrantes del núcleo familiar. Si alguno funciona mal, la armonía del hogar se ve afectada.
- Las relaciones de comportamiento hacen referencia a los vínculos entre los elementos del sistema y el mundo exterior; vínculos como los que podemos establecer entre nuestra familia con algún vecino del barrio.

Y, ¿para qué nos sirve un sistema? El objetivo de todo sistema debe ser único, abstracto, cuantificable y realizarlo con los medios económicos mínimos.

El **objetivo** del sistema **debe desagregarse**, es decir, dividirse en pequeños objetivos, para comprender cada uno de los procesos involucrados en el sistema. Cada parte representa un **sub-objetivo** con las características propias de cualquier objetivo.

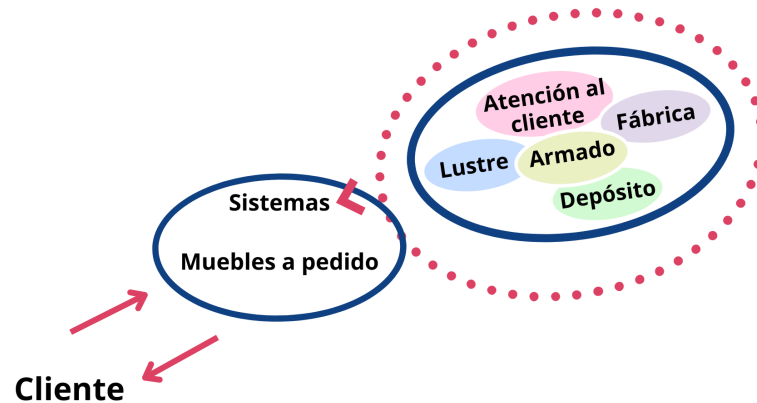
Para cumplir con el **objetivo general** del sistema, la sumatoria de los sub-objetivos debe resultar igual a él. La cantidad de niveles alcanzados en la desagregación depende de la complejidad del sistema.

[Veamos un ejemplo...](#)



## Grafiquemos con un ejemplo

En el esquema que se muestra debajo, identificamos que un sistema puede formar parte de otro más general. El sistema general "*Muebles a pedido*" que sería su entorno, está formado por otros sistemas, que en este caso lo tendrían a él como entorno común denominándose subsistemas (*fábrica, lustre, depósito, atención al cliente, armado*).



## Para tener en cuenta

Los sistemas, desde su concepción hasta su desaparición, pasan por una serie de etapas. En otras palabras, cuando se desarrolla un sistema, nunca funciona de entrada, siempre hay que testearlo, agregar otros módulos y mantenerlo hasta que deje de existir, por eso tiene continuas revisiones.

### Paso a paso:

- Surge de la necesidad del cliente.
- Se diseña la construcción del sistema.
- Se pone en marcha.
- Tiene constantes revisiones hasta su abandono o reemplazo.

Así, el sistema pasa por una serie de fases que constituirán su **ciclo de vida**.



## ¿Por qué nos referimos a "sistemas" ?

La materia Desarrollo de Sistemas Orientados a Objetos te muestra e invita a que al final de la cursada presentes un producto desarrollado bajo el paradigma orientado de objeto, y este producto es un **sistema**.

Por eso, nos referimos a sistemas, pues, reconocer y diferenciar un producto que puede dividirse en porciones fáciles de resolver te permite desarrollar tu propio sistema.

A lo largo de tu tarea como programador/a, recibirás distintos requerimientos por parte de tus clientes. La división del objetivo que te proponga la solicitud dará origen a cada uno de los procesos que vayas a codificar.

### Acerca de los agentes externos

Frente a todo requerimiento, encontrarás un límite (desde dónde y hasta dónde alcanzar el objetivo) y, también, descubrirás que existen elementos externos al propio sistema; estos elementos interactúan y despiertan las tareas que contiene y se denominan agentes externos.

En el sistema que vas a desarrollar, el agente externo es el usuario destinatario y el límite se define cuando te encuentres frente a la práctica formativa integradora.

¡Sigamos avanzando con la semana! Volveremos sobre este punto en la semana de entrega del entorno del desarrollo.



## Del sistema al paradigma de programación

Un **paradigma de programación**:

- es un modelo básico de diseño e implementación de programas,
- permite desarrollar software conforme a ciertos principios o fundamentos específicos que se aceptan como válidos.
- es un marco conceptual que determina los bloques básicos de construcción de software y los criterios para su uso y combinación.

Entonces, se puede decir que un paradigma puede ser implementado en diferentes lenguajes de programación, pero eso no implica que si sabemos cuál es el **lenguaje** en el que se construyó un programa podamos saber cuál es el paradigma que hay detrás.

Esto se debe a que el lenguaje es lo concreto, es la forma en que se va a codificar y el paradigma es más abstracto, está en el orden de lo conceptual, de cómo pensamos la solución a un problema.

Es decir, que hay una diferencia entre lo que es el lenguaje y el paradigma de programación.



### ¿Hay diferentes paradigmas de programación?

Sí, y en esta materia nos enfocamos en el **paradigma orientado a objetos** en el que se ve cómo un conjunto de objetos interactúan entre sí a través del envío de mensajes.





## Ejemplo: Sistema de negocios en línea

Veamos un ejemplo centrado en un sistema para un negocio e-commerce de ropa para niños y niñas.

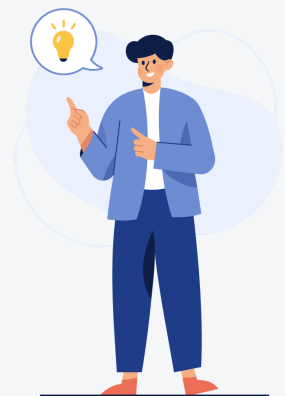
**¿Por que si hay varios paradigmas seleccionamos el orientados a objetos para programar sistemas?**

Tradicionalmente el programador escribía línea a línea de código, una debajo de otra, con decisiones destinadas a bifurcar caminos para resolver un requerimiento de usuario, pero cuando los proyectos son muy grandes se complican.

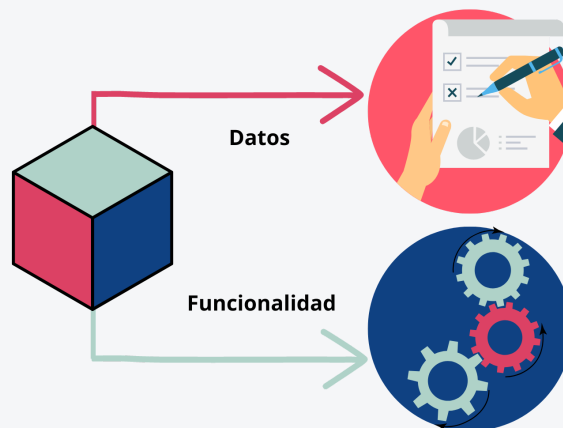
Supongamos que el desarrollo se centra en un sistema de un negocio en línea, este negocio es de venta de ropa de niños y niñas

Entre los servicios que ofrece tenemos:

- el filtro por producto para leer las características y el precio,
- la reserva de clientes para comprar al por mayor,
- el procesamiento del carrito de compras que a su vez se vincula con algún proceso de pago en línea.



**Observemos el siguiente esquema**



**¿Qué información nos ofrece este esquema?** Desde nuestro negocio, cargamos los datos recibidos por el cliente.

Así, pasamos a definir procedimientos y construir una funcionalidad, aquella referida a las acciones y tareas que tiene el proceso, mezclada a varios objetos con otras funciones que se comunican entre sí.

Esta comunicación generará una nueva funcionalidad, un nuevo comportamiento, a través de un nuevo objeto. O bien, también, podrás agregarle características a los ya existentes.

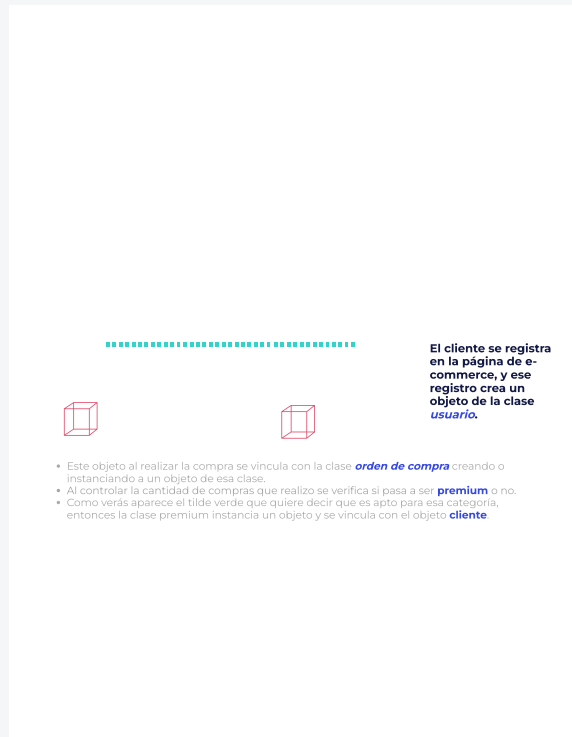
A los datos con sus valores utilizados los denominamos **"atributos"** y a su comportamiento, es decir, la funcionalidad lo denominamos **"métodos"**.



## Vinculación cliente-objeto

Entonces, ¿cómo vinculamos los objetos, los métodos y lo que veremos en el curso con el desarrollo de sistemas?

En el negocio de ropa de niños, los clientes se registran, en una página e-commerce, para comprar los productos, y llegada a una determinada cantidad de compras definida por la empresa se convierte en cliente premium.



Tal vez las palabras *clase*, *objeto*, *instancia* sean raras en este momento. Estas palabras son parte del paradigma orientado a objetos que comenzarás a transitar. Lo importante es que comiences a pensar que el sistema tiene objetos reales, que van a usar métodos y que se comunican entre sí.



### Para tener en cuenta:

Cuando se diseñan correctamente las clases, se pueden usar en distintas partes del programa y en diferentes proyectos por lo tanto permite reutilizar código. Esta forma de programar facilita la detección de errores en el código y la modularidad.