

# 3

## Consideraciones sobre los objetos

# Importancia

- ¿Cómo decide qué componentes son necesarios para algo que va a crear, como una casa o un mueble?
- ¿Qué es una taxonomía?
- ¿Cómo se relacionan los organismos de una taxonomía?
- ¿Cuál es la diferencia entre atributos y valores?

# Análisis de un problema mediante el análisis orientado a objetos

Duke's Choice vende ropa de su catálogo. El negocio crece un 30% al año y es necesario un nuevo sistema de introducción de pedidos.



# Proceso de pedido de Duke's Choice



adacion@proydesa.org) has a  
this Student Guide.

## Identificación de un dominio de problemas

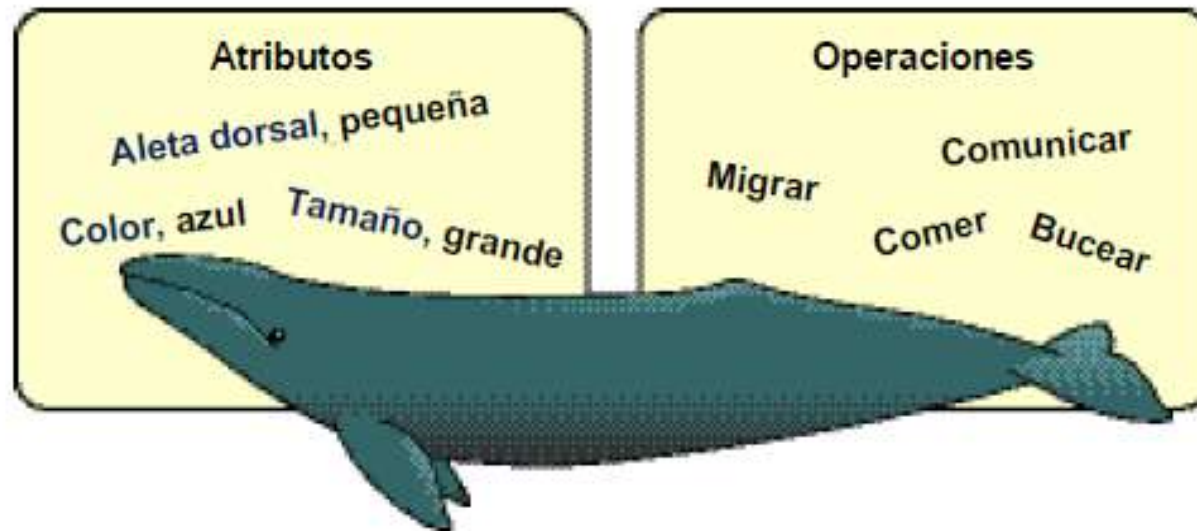
- Un dominio de problemas es el ámbito del problema que va a solucionar.
- Ejemplo: “crear un sistema que permita el método de introducción de pedidos en línea para aceptar y verificar el pago de un pedido”.

(fundacion@proydesa.org) has a  
use this Student Guide.

## Identificación de objetos

- Los objetos pueden ser físicos o conceptuales.
- Los objetos tienen *atributos* (características) como el tamaño, el nombre, la forma, etc.
- Los objetos tienen *operaciones* (cosas que pueden hacer) como la definición de un valor, la visualización de una pantalla o el aumento de la velocidad.

# Identificación de objetos



## Criterios adicionales para reconocer objetos

- Importancia del dominio de problemas:
  - ¿Existe el objeto en los límites del dominio de problemas?
  - ¿Es necesario el objeto para que se termine la solución?
  - ¿Es necesario el objeto como parte de una interacción entre un usuario y el sistema?
- Existencia independiente



## Posibles objetos en el caso práctico de Duke's Choice

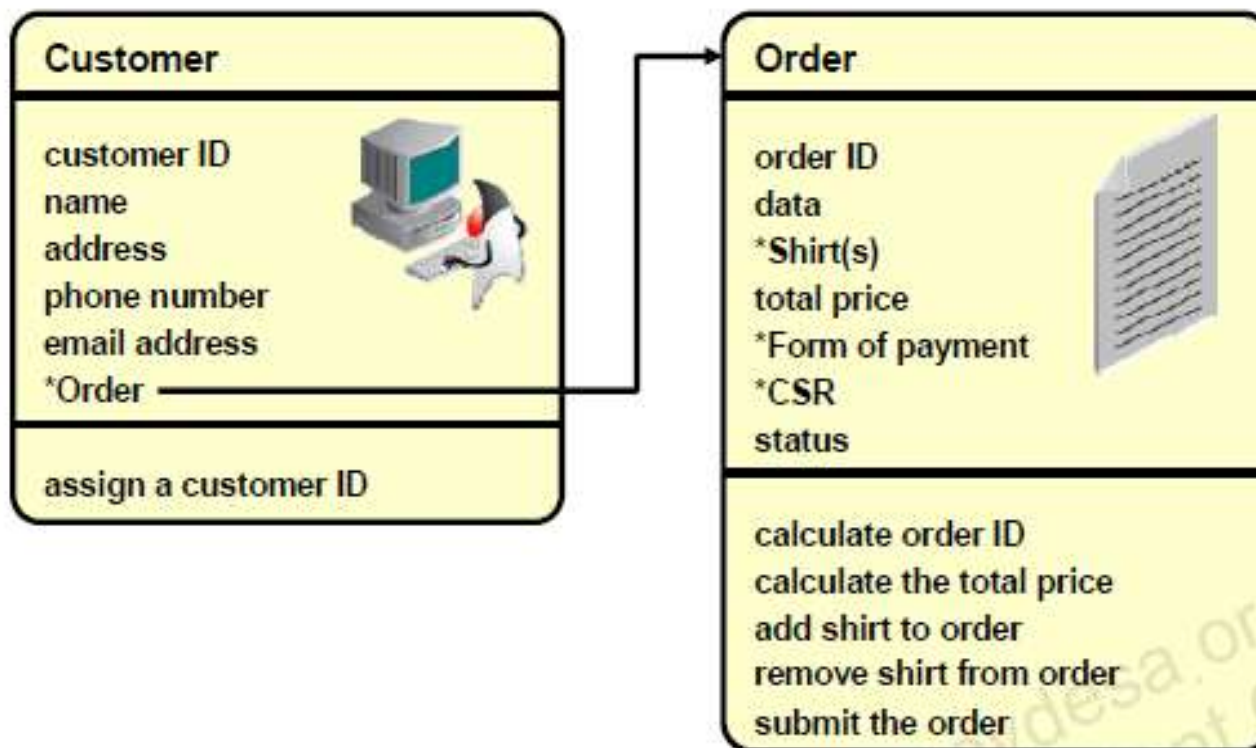


fundacion@proydesa.org) has a  
this Student Guide.




## Identificación de atributos y operaciones de objetos

- Los atributos son datos, como:
  - ID
  - Objeto de pedido
- Las operaciones son acciones, como:
  - Suprimir elemento
  - Cambiar ID

# Objeto con otro objeto como atributo



# Posibles atributos y operaciones para objetos en el caso práctico de Duke's Choice

<p>order ID <b>Order</b> data *Shirt(s) total price *Form of payment *CSR status</p> 	<p>shirt ID price description size color code</p> <p><b>Shirt</b></p> 	<p>customer ID <b>Customer</b> name address phone number email address *Order</p> 
<p>calculate order ID calculate the total price add shirt to order remove shirt from order submit the order</p>	<p>calculate shirt ID display shirt information</p>	<p>assign a customer ID</p>

## Solución del caso práctico: *Clases*

<i>Clase</i>	<b>Order</b>	<b>Shirt</b>	<b>Customer</b>	<b>Form of Payment</b>	<b>Catalog</b>	<b>CSR</b>
--------------	--------------	--------------	-----------------	----------------------------	----------------	------------

## Solución del caso práctico: *Atributos*

Clase	Order	Shirt	Customer
Atributos	order ID date *Shirt(s) total price *Form of payment *CSR status	shirt ID price description size color code	customer ID name address phone number email address *Order

## Solución del caso práctico: *Atributos*

Clase	Form of Payment	Catalog	CSR
Atributos	customer ID name address phone number email address *Order	*Shirt(s)	name extension

## Solución del caso práctico: *Comportamientos*

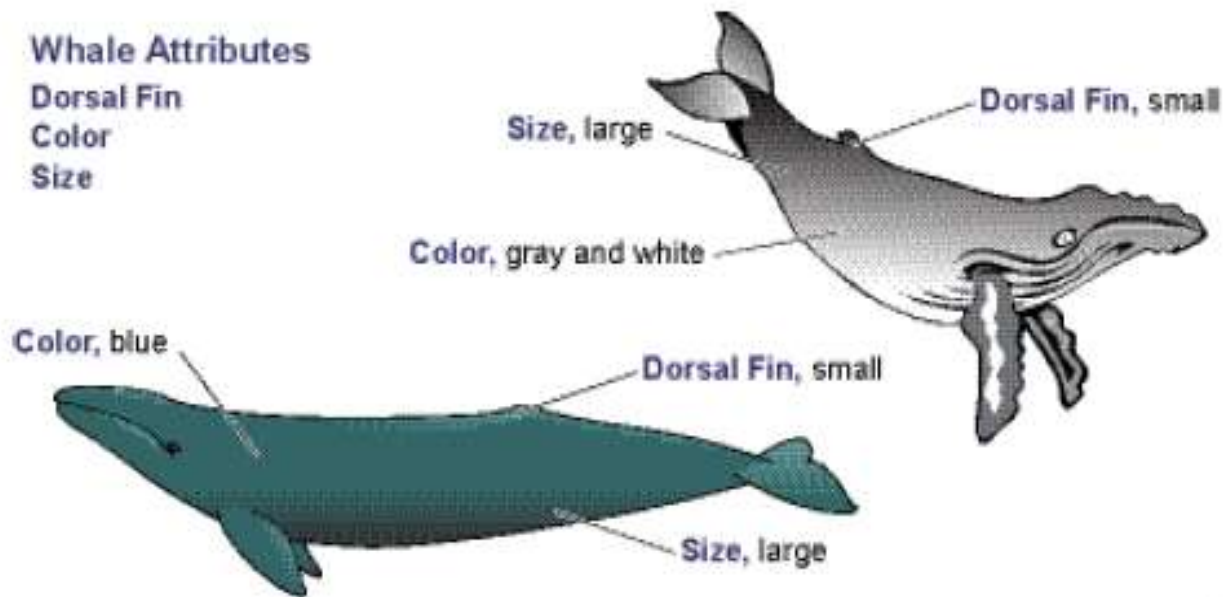
Clase	Order	Shirt	Customer
Atributos	customer ID name address phone number email address *Order	*Shirt(s)	name extension
Comportamientos	verify credit card number verify check payment	add a shirt remove a shirt	process order



# Solución del caso práctico: *Comportamientos*

<i>Clase</i>	<b>Form of Payment</b>	<b>Catalog</b>	<b>CSR</b>
<i>Atributos</i>	customer ID name address phone number email address *Order	*Shirt(s)	name extension
<i>Comportamientos</i>	verify credit card number verify check payment	add a shirt remove a shirt	process order

# Diseño de clases



# Clases y objetos resultantes

Shirt
shirtID price description size colorCode R=Red, B=Blue, G=Green
calculateShirtID ( ) displayShirtInformation ( )

**Clase  
Shirt**



**Objetos Shirt**

# Modelado de clases

Sintaxis:

## **ClassName**

```
attributeVariableName [range of values]  
attributeVariableName [range of values]  
attributeVariableName [range of values]  
...
```

```
methodName()  
methodName()  
methodName()  
...
```

# Modelado de clases

Ejemplo:

## Shirt

shirtID

price

description

size

colorCode R=Red, B=Blue, G=Green

calculateShirtID()

displayInformation()

## Uso del modelado similar a UML

UML: Unified Modeling Language

- UML se utiliza para:
  - Modelar los objetos, los atributos, las operaciones y las relaciones en programas orientados a objetos.
  - Modelar el comportamiento dinámico del sistema mostrando colaboraciones entre objetos y cambios en los estados internos de objetos.
- Hay muchos cursos disponibles que enseñan UML.