



TECNOLÓGICO
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO DE NUEVO LAREDO

“Con la Ciencia por la Humanidad”

Introducción a la Ingeniería en Sistemas Computacionales y al Diseño Orientado a Objetos

Curso propedéutico

Instructor:

Bruno López Takeyas

bruno.lt@nlaredo.tecnm.mx

9.- INTRODUCCIÓN AL PARADIGMA ORIENTADO A OBJETOS

- 9.1 Elementos del modelo de objetos
 - 9.1.1 Clase
 - 9.1.2 Objeto
 - 9.1.3 Instancia
 - 9.1.4 Control de acceso a los elementos de una clase
 - 9.1.5 Atributos
 - 9.1.6 Métodos
 - 9.1.7 Propiedades
 - 9.1.8 Métodos *mutator* y *accessor*
 - 9.3 El Lenguaje Unificado de Modelado (UML)
 - 9.3.1 Diagramas de clases en UML
-

El Paradigma Orientado a Objetos

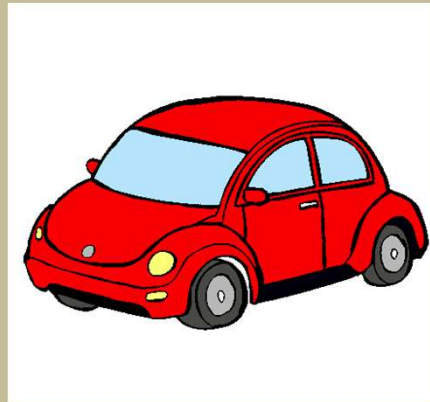
- La Programación Orientada a Objetos (POO), es un conjunto de técnicas de descripción, diseño, modelado y programación que se utilizan para desarrollar proyectos de manera eficiente donde los objetos son los elementos principales de construcción.

- La vida cotidiana esta llena de **objetos**, como por ejemplo: una mesa, un libro, un automóvil.
- Los objetos no necesariamente representan cosas, también pueden modelar personas, animales o algunas otras entidades abstractas o intangibles como una cuenta bancaria, una factura electrónica, un mensaje de correo electrónico, etc. donde cada objeto tiene características que lo distinguen y sirve para identificarlo y tienen acciones asociadas a ellos.

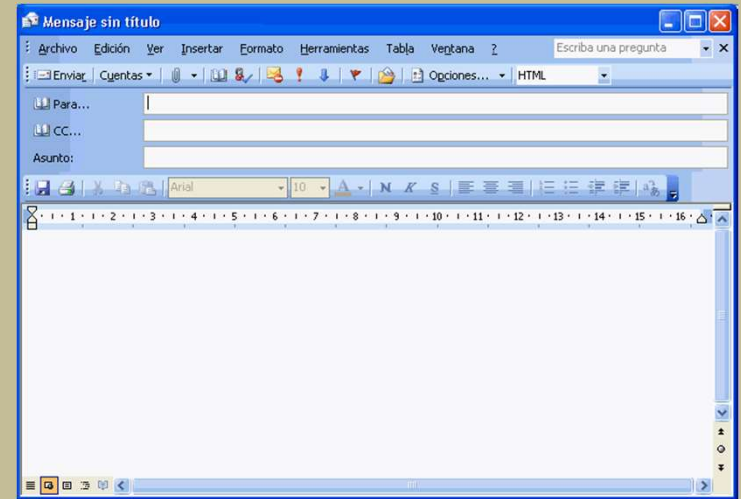
Objetos tangibles e intangibles



Mesa



Automóvil



Email



Personas

[Nombre de la compañía] **FACTURA**

[Dirección de la calle]
[Ciudad, Estado Código postal]
Teléfono: (000) 000-0000

FACTURA #	FECHA
2034	21/02/2018

FACTURAR A	IDENTIFICACIÓN DEL CLIENTE	TÉRMINOS
[Nombre] [Nombre de la compañía] [Dirección de la calle] [Ciudad, Estado Código postal] [Teléfono] [Dirección de correo electrónico]	564	Pagadero al recibirse

DESCRIPCIÓN	CANT	PRECIO UNITARIO	MONTO
Honorarios por servicios	1	200,00	200,00
Mano de obra: 5 horas a \$75/hr	5	75,00	375,00
Descuento por nuevo cliente		(50,00)	(50,00)

Factura

¿Qué es una clase?

- Es una abstracción que representa a un conjunto de objetos con un comportamiento e interfaz común.
- Una **Clase** es un molde o plantilla que representa de manera general las características y conductas de una entidad que sirve para crear objetos.

- Los valores representan las características que pueden tener los objetos y las conductas representan las acciones que pueden realizar.
- Por ejemplo, si se define una clase llamada **Estudiante**, las características pueden ser el número de matrícula, nombre, fecha de nacimiento, domicilio, mientras que las acciones que puede realizar son inscribirse, acreditar materias, reprobar o graduarse.

Nombre de las clases

- Al definir una clase se recomienda utilizar un sustantivo en singular para identificarla, por ejemplo: **Alumno**, **Profesor**, **Cheque**.

¿Qué es un objeto?

- Es un elemento particular creado de una clase, que tiene sus propias características y conductas (diferentes a las de otros objetos).
- Sabemos que algo es un objeto si tiene un nombre, pueden definirse sus características y es capaz de realizar alguna actividad o comportarse de cierta manera.

- Por ejemplo, cada uno de los estudiantes de una universidad, representan objetos de la clase **Estudiante** (mencionada anteriormente), con sus propias características y acciones individuales, es decir, cada uno tiene un número de matrícula, nombre, fecha de nacimiento, domicilio, etc. diferente al resto de los estudiantes y puede inscribirse, acreditar, reprobado o graduarse de manera distinta a los demás.

Ejemplo: Clase “Perro”



- “Firulais” es un “PERRO”
- “Firulais” es un OBJETO
- “PERRO” es la clase de “Firulais”



- “Chester” es un “PERRO”
- “Chester” es un OBJETO
- “PERRO” es la clase de “Chester”



Un objeto:

- **Características:**
 - Nombre: “Firulais”
 - Raza: “Chihuahua”
 - Color: “Café”
- **Acciones:**
 - Ladrar()
 - Comer()
 - Dormir()



Otro objeto:

- **Características:**
 - Nombre: “Chester”
 - Raza: “Schnauzer”
 - Color: “Negro”
- **Acciones:**
 - Ladrar()
 - Comer()
 - Dormir()

Instancia

- Es la creación o manifestación concreta de un objeto a partir de una clase, que tiene sus propias características y acciones que lo identifican y diferencian del resto de los objetos.
- Un objeto es la **instancia** de una clase.



Instancias

Control de acceso a los elementos de una clase

- El diseño orientado a objetos permite establecer controles para restringir la visibilidad y acceso a los miembros de una clase, de tal forma que pueden proteger y ocultar los datos y acciones definidas en las clases del acceso de otras clases (y por ende de los objetos creados a partir de ellas).

- Al definir los elementos de una clase (características y acciones), debe especificarse el nivel de restricción que se asignará a cada uno de ellos mediante los modificadores de acceso (público / privado).

- Los miembros **privados** de una clase están ocultos para las demás clases, es decir, solo pueden ser accedidos por los miembros internos de la clase que los contiene.
- Los miembros **públicos** están visibles para las demás clases y por lo tanto pueden ser accedidos por ellos.
- Existen otros modificadores de acceso que serán explicados mas adelante.

¿Qué es un atributo?

- Representa una característica o dato de un objeto (definidos en la clase respectiva).
- Un objeto puede tener muchos atributos.
- Los atributos son controlados a través de variables, por lo tanto se debe identificar los datos que tendrá el objeto y determinar el tipo de dato.

Métodos

- Un objeto no solo almacena datos sino también realiza algunas acciones que definen su comportamiento.
- Los métodos son las actividades que pueden realizar o tener los objeto.

- Recordando que un método consiste en escribir un segmento de código una sola vez en un módulo e invocarlo las veces que sean necesarias.
- En el paradigma orientado a objetos, los métodos juegan un papel fundamental para que los objetos muestren un determinado comportamiento y brindar la posibilidad de cooperación entre varios objetos.

Propiedades

- Una propiedad es un mecanismo que funge como canal mediante el cual se accede a consultar o modificar un atributo.
- Una propiedad puede tener asociado código al momento de leer o modificar el valor de un atributo, de tal forma que cuando accede al valor puede validar que sea un valor adecuado o asegurarse que este dentro de un rango permitido

Descriptores de acceso

- Las propiedades pueden tener dos descriptores de acceso: *get* y *set*.
- El descriptor *get* se utiliza para leer o consultar el valor de un atributo.
- El descriptor *set* se implementa cuando se desea modificar el valor del atributo.

- La forma tradicional de implementar propiedades es declarar los atributos de modo privado y asociar una propiedad pública a cada uno de ellos.

Métodos *Mutator* y *Accessor*

- Las propiedades son el mecanismo natural para introducir o consultar los valores de los atributos, sin embargo, no todos los lenguajes de programación contienen propiedades.
- Cuando se carece de propiedades entonces la forma de interactuar con los datos de un objeto es a través de métodos conocidos como *Mutator* y *Accessor*.

- Un *Mutator* es un método que sirve para introducir valores a los atributos privados de un objeto. Se trata de un procedimiento ya que no devuelve dato, solamente recibe como parámetro el valor que desea introducir.
- Un *Accessor* es un método que sirve para extraer o consultar un valor que contiene un atributo privado de un objeto. Se trata de una función que no recibe parámetro, simplemente contiene una sentencia *return* para devolver el valor del atributo.

El Lenguaje Unificado de Modelado

UML

- El Lenguaje Unificado de Modelado (UML por sus siglas en inglés Unified Modeling Language) es un lenguaje gráfico para modelar sistemas computacionales que sirve para especificar, diseñar, visualizar y documentar un sistema.
- Un diagrama de clases es un tipo de diagrama que muestra el diseño de un sistema con las clases, sus atributos, métodos, interfaces y las relaciones con otras clases.

- En UML una clase se representa mediante un rectángulo dividido en tres secciones: en la primera sección se coloca el nombre de la clase, en la segunda los datos (atributos) y en la tercera las acciones (métodos y propiedades).
- Un diagrama de clases también puede contener comentarios, los cuales se colocan en un rectángulo con una esquina doblada.

Ejemplo de clase con comentario

Nombre de la Clase

Automóvil

Atributos

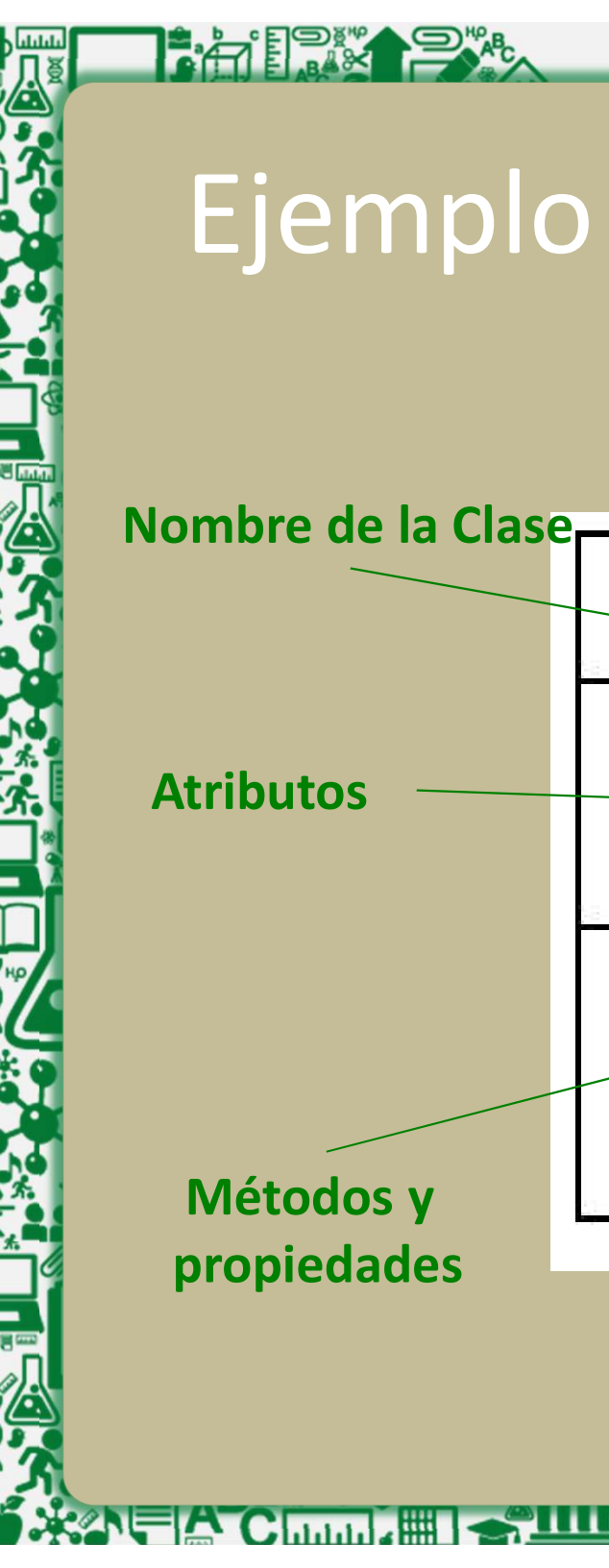
marca
modelo
tipo

Comentario

Este es un comentario

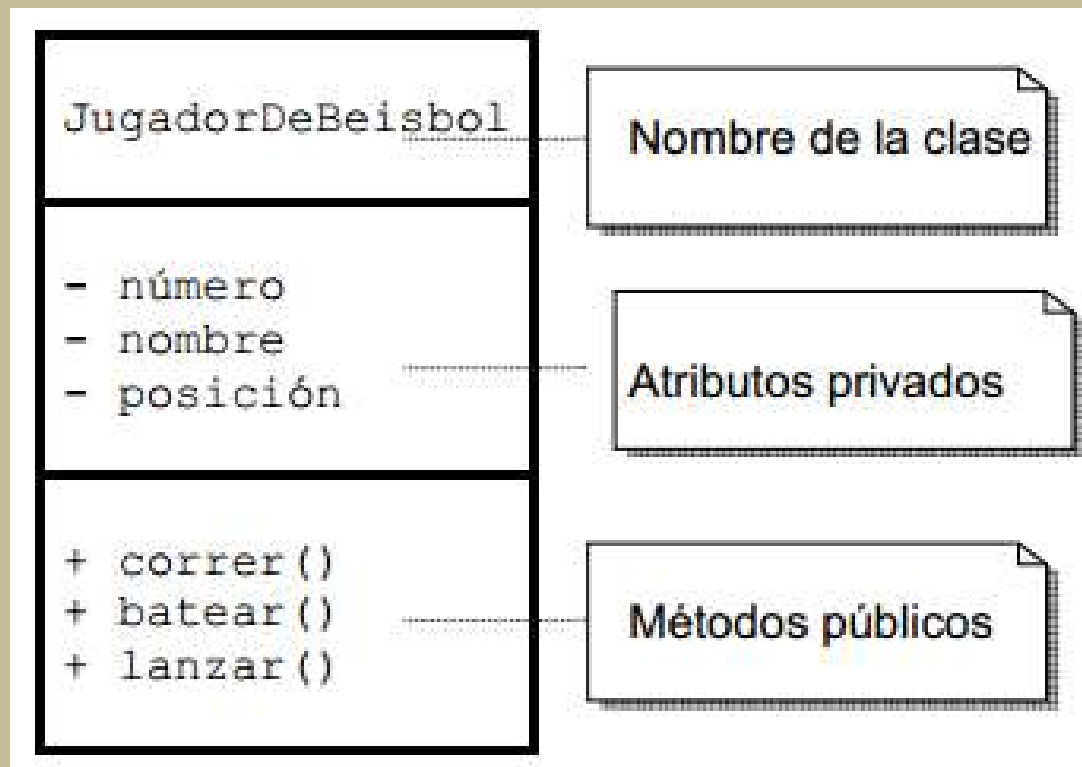
Métodos y propiedades

encender()
apagar()
acelerar()
frenar()

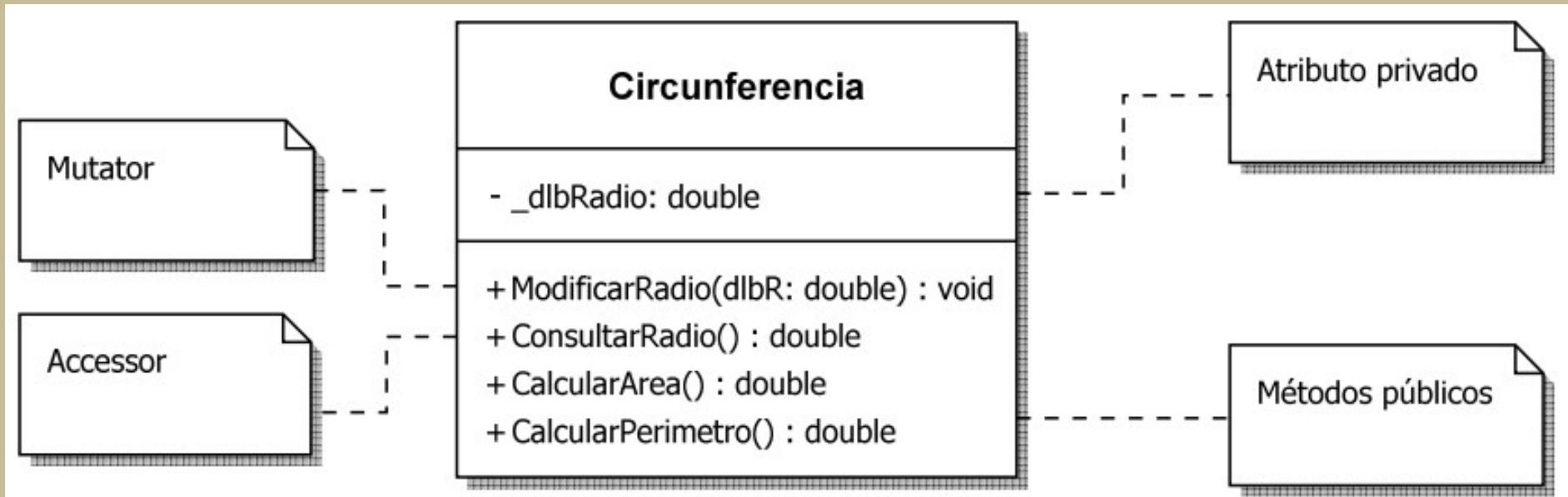


Ejemplo de clase estableciendo el tipo de acceso

- Para establecer el tipo de acceso a un miembro de la clase se antepone el símbolo **-** para privado y el símbolo **+** para público.



Ejemplo de una clase con *Accessor* y *Mutator*



Consejos al diseñar clases y utilizar objetos

- No se deben capturar datos calculables
- Los datos calculables **NO** deben ser atributos
- Una vez creado el objeto, primero se deben introducir sus valores y después ejecutar sus métodos
- No se recomienda capturar ni imprimir datos desde un método ubicado dentro de una clase.



NClass

- Software de libre distribución que se utiliza para el diseño de diagramas de clases UML
- Puede descargarse en

<http://nclass.sourceforge.net>

NClass

Home

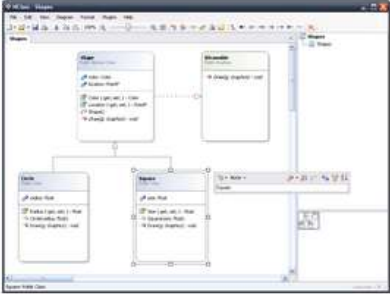
Overview

NClass is a free tool to easily create UML class diagrams with full C# and Java language support. The user interface is designed to be simple and user-friendly for easy and fast development. Properties, enums, delegates and other language specific elements are fully supported with strict syntactical and semantical verification.

Design your application with just a few clicks - the main goal is to provide a simple but powerful class designer that is very intuitive to use. Diagram styles help you to create professional looking diagrams, just like in Visual Studio or other commercial products. Furthermore, you can generate code from your models or you can also import classes from existing .NET assemblies.

Please note, NClass is far from complete. There are many missing features that are planned for future versions. But if you have an idea, please share your thoughts in one of the way listed on the support page.

[Download NClass](#) ↓



Prácticas

■ Descargue del sitio web:

<https://nlaredo.tecnm.mx/takeyas/LibroISC>

■ *Práctica 9.1.- Clases y objetos*



Tarea

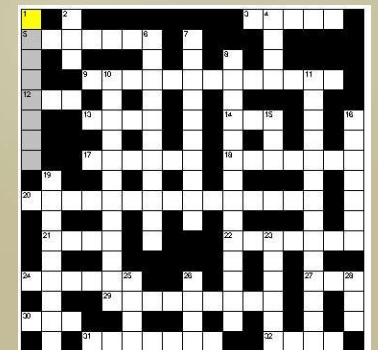
Resuelva en el sitio web

<https://nlaredo.tecnm.mx/takeyas/LibroISC>

Cuestionario 9.1



Crucigrama 9.1

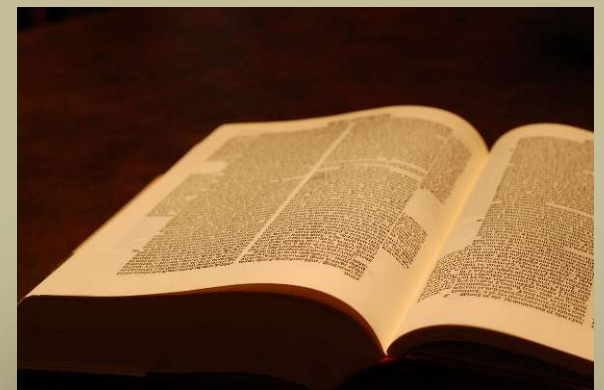


Lectura complementaria

Consulte en el sitio web

<https://nlaredo.tecnm.mx/takeyas/LibroISC>

- Lectura complementaria: Nomenclatura sugerida para identificar los componentes de un proyecto



Fuentes de información:

- López Takeyas, Bruno. (2019) “Introducción a la Ingeniería en Sistemas Computacionales y al diseño orientado a objetos”. Editorial Pearson.
- <https://nlarredo.tecnm.mx/takeyas/LibroISC/>

