

Introducción a la programación con C#

Introducción al lenguaje C# y Algoritmos



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
INSTITUTO REGIONAL BUENOS AIRES

**Centro de
e-Learning**

¿Qué vamos a ver hoy?

- Entender el ecosistema .NET
- Instrucciones básicas de programación
- Primeros conceptos de POO
- Como crear un proyecto desde Visual Studio
- Diferentes tipos de proyectos

.NET



Web

Build web apps and services for macOS, Windows, Linux, and Docker.



Mobile

Use a single codebase to build native mobile apps for iOS, Android, and Windows.



Desktop

Create beautiful and compelling desktop apps for Windows and macOS.



Microservices

Create independently deployable microservices that run on Docker containers.



Cloud

Consume existing cloud services, or create and deploy your own.



Machine Learning

Add vision algorithms, speech processing, predictive models, and more to your apps.



Game Development

Develop 2D and 3D games for the most popular desktops, phones, and consoles.



Internet of Things

Make IoT apps, with native support for the Raspberry Pi and other single-board computers.

Desde [acá](#), podés acceder a la web oficial y bajarte la última versión del framework

¿Qué es la POO?

Es un paradigma de programación. En el que los objetos se utilizan como metáfora para representar las entidades reales de lo que queremos modelar.

Bases o conceptos:

- Abstracción
- Encapsulación
- Modularización
- Jerarquización

Abstracción

Es el proceso mental de extracción de las características esenciales de algo, ignorando los detalles superficiales.

Coche alquilado:

- La tarifa
- Limitación de kms
- ¿Consumo?

Coche particular:

- Consumo
- El color
- Valor de reventa
- Duración de los componentes

Coche para repartos:

- Consumo
- Carga soportada
- Garantía

Encapsulación

Es el proceso en el cual se ocultan los detalles del soporte de las características de una abstracción.

No se trata de ocultar las características en sí, sino de no mostrar cómo guardo o maneja internamente esas características.

Por ej, de la clase fecha nos interesa obtener el día, mes y año. Y no como esta guardado.

Modularización

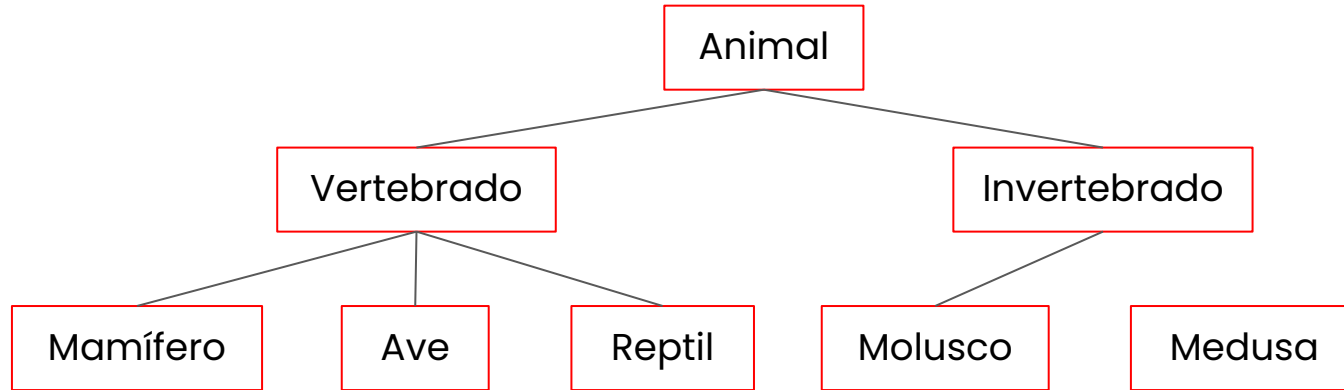
Es el proceso de descomposición de un sistema en un conjunto de módulos o piezas. Que esten poco **acopladas** (independientes) y **cohesivas** (con significado propio).

Se entiende por sistema al conjunto de piezas que colaborar entre sí.

Por ej, el módulo de cierre centralizado con el módulo de la alarma del auto.

Jerarquización

Es el proceso de estructuración por el que se produce una organización de un conjunto de elementos en niveles de responsabilidad.



Elementos

- Clase: descripción de los datos y de las operaciones de un elemento modelado.
- Objeto: instancia de una clase, ejemplar concreto.
- Mensaje: invocación de una operación sobre un objeto.
- Método: definición de una operación de una clase.
- Atributo: los datos de una clase, y por tanto, presente en todos los objetos de esa clase.
- Estado: conjunto de los valores de los atributos que tiene un objeto.

Relación de elementos

Una **clase**, es la definición de los **atributos** y **métodos** que describen el comportamiento de un **objeto**.

Un **objeto**, es un ejemplar concreto de una **clase** que responde a los **mensajes** correspondientes a sus **métodos**, adecuándose al **estado** de sus **atributos**.

Palabras claves

Son identificadores reservados predefinidos que tienen significados especiales para el compilador.

Estos no se pueden utilizar como identificadores en nuestro programa.

Algunos ejemplos:

- abstract
- break
- continue

[Documentación oficial](#)

Modificadores de visibilidad

Todos los tipos y miembros de tipo (clase, método, atributo) tienen un nivel de accesibilidad. Controla si se pueden usar desde otro código en su ensamblado o en otros ensamblados.

- public: puede ser accedido desde cualquier código.
- private: puede ser accedido desde la misma clase.
- internal: puede ser accedido desde el mismo 'assembly'.

[Documentación oficial](#)

Sentencias de flujo de ejecución

Son aquellas sentencias que nos permiten manipular nuestro código. Que, dependiendo determinado criterio, nuestro código salte a una u otra línea.

- If
- Switch
- For
- Foreach
- While
- Do While
- Try & Catch

¡¡A trabajar!!

