

# C#

TechGuide - Alura, FIAP e PM3

---

## Nivel 1

### C# - Fundamentos:

- C# es un lenguaje de programación, multiparadigma, con tipado fuerte, desarrollado por Microsoft como parte de la plataforma .NET. El código fuente se compila a Common Intermediate Language (CIL), que es interpretado por la máquina virtual Common Language Runtime (CLR). Está diseñado para funcionar en la Common Language Infrastructure de la plataforma .NET Framework.
- Conocer los tipos primitivos
- Declarar variables, considerando los diferentes tipos
- Utilizar estructuras condicionales ('if', 'else')
- Conocer los operadores de asignación y comparación
- Utilizar estructuras de repetición y bucles ('while', 'for')
- Utilizar funciones, pasando parámetros y argumentos
- Manejar métodos
- Manejar arrays y listas
- Obtener datos de una API
- Realizar llamadas asíncronas, etc.
- Crear constructores

### Contenidos

- **Web** Paseo por el lenguaje C# (<https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>)
- **Web** Excepciones y control de excepciones (<https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/exceptions/>)
- **Web** Operadores y expresiones de C# (referencia de C#) (<https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/operators/>)

- **Web** Constructores: Guía de programación de C# (<https://learn.microsoft.com/es-es/dotnet/csharp/programming-guide/classes-and-structs/constructors>)
- **Web** Programación asincrónica (<https://learn.microsoft.com/es-es/dotnet/csharp/asynchronous-programming/async-scenarios>)
- **Web** Cadenas con formato de fecha y hora estándar (<https://learn.microsoft.com/es-es/dotnet/standard/base-types/standard-date-and-time-format-strings>)
- **Web** Procedimiento para separar cadenas mediante String.Split en C# (<https://learn.microsoft.com/es-es/dotnet/csharp/how-to/parse-strings-using-split>)
- **Web** Structs - Especificación del lenguaje C# (<https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/language-specification/structs>)
- **Web** C# | El lenguaje de programación moderno, innovador y de código abierto para crear todas sus aplicaciones (<https://dotnet.microsoft.com/es-es/languages/csharp>)
- **Web** Creación de una aplicación de Windows Forms en Visual Studio con C# (<https://learn.microsoft.com/es-es/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022>)
- **Artículo** Variables y Tipos en C# (<https://desarrolloweb.com/articulos/variables-tipos-csharp>)
- **YouTube** ¿QUÉ ES .NET? 🌟 [Lo más BÁSICO y lo más SENCILLO] (<https://www.youtube.com/watch?v=AU2WmbIU2J0>)
- **YouTube** List y ArrayList C# (<https://www.youtube.com/watch?v=k431eUiMRXA>)
- **YouTube** Tipos de datos primitivos y Variables (<https://www.youtube.com/watch?v=Hhe6EtOsG5o&t=655s>)
- **YouTube** Consumir un API con C# 9.0 y Pegar JSON Como Clases Visual Studio 2019 (<https://www.youtube.com/watch?v=yPgLYaqMn8>)
- **YouTube** APRENDE Windows Forms DESDE 0 Utilizando C# y Visual Studio 2022 (<https://www.youtube.com/watch?v=FFlvrBpY73w>)
- **YouTube** Lenguaje C Sharp en 3 Minutos o más! (<https://www.youtube.com/watch?v=rr5nZq61WIE>)

### Contenidos Alura:

- **Artículo** ¿Que es .Net framework? (<https://www.aluracursos.com/blog/que-es-net-framework>)
- **Artículo** Manejando fechas y horas en C# (<https://www.aluracursos.com/blog/manejando-fechas-y-horas-en-csharp>)
- **Artículo** C#: entender clases y structs (<https://www.aluracursos.com/blog/entender-clases-structs>)

- **Artículo** ¿Cómo separar palabras en String con C#? (<https://www.aluracursos.com/blog/como-separar-palabras-en-string>)
- **Curso** Formación C# y Orientación a Objetos (<https://app.aluracursos.com/formacion-c-sharpe-orientacion-a-objetos>)
- **Curso** Curso C#: conociendo el lenguaje (<https://app.aluracursos.com/course/csharp-conociendo-el-lenguaje>)
- **Curso** Curso C#: array y tipos genéricos (<https://app.aluracursos.com/course/csharp-array-tipos-genericos>)
- **Curso** Curso C#: strings, expresiones regulares y clase Object (<https://app.aluracursos.com/course/csharp-strings-expresiones-regulares-clase-object>)

## Conceptos de Orientación a Objetos:

- La Programación Orientada a Objetos es un paradigma de programación de software basado en la composición e interacción entre diversas unidades llamadas 'objetos' y las clases, que contienen una identidad, propiedades y métodos. Se basa en cuatro componentes de la programación - abstracción digital, encapsulación, herencia y polimorfismo.
- Cómo funcionan los objetos
- Crear y utilizar constructores
- Qué son las clases
- Crear y utilizar métodos
- Cómo funciona la encapsulación
- Qué es la herencia
- Qué es el polimorfismo
- Cómo funcionan las interfaces
- Qué son las abstracciones

## Contenidos

- **Web** Programación orientada a objetos (C#) (<https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/tutorials/oop>)
- **Web** Programación orientada a objetos - IBM (<https://www.ibm.com/docs/es/spss-modeler/saas?topic=language-object-oriented-programming>)
- **Web** Introducción a los objetos JavaScript (<https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects>)
- **Artículo** Introducción a POO con JavaScript ES6 (<https://medium.com/academia-hack/introducci%C3%B3n-a-poo-con-javascript-es6-80074fde0cdf>)

- **Artículo** Programación Orientada a Objetos (<https://ellibrodepython.com/programacion-orientada-a-objetos-python>)
- **YouTube** 4 Principios de la Programación Orientada a Objetos (<https://www.youtube.com/watch?v=Uk1C1NARZjU>)
- **YouTube** Mini Curso: POO con PHP (básico) ([https://youtu.be/Ben\\_VC2rm10](https://youtu.be/Ben_VC2rm10))
- **YouTube** 🟡 ¿Qué es la Programación Orientada a Objetos en PYTHON? - [ Con EJEMPLOS ] | Python desde CERO #13 (<https://www.youtube.com/watch?v=KwT1F7uL5rA>)

### Contenidos Alura:

- **YouTube** Alura Latam: ¿Qué es la Programación Orientada a Objetos? (<https://www.youtube.com/watch?v=Oigen2sjagk>)
- **Artículo** POO: ¿Qué es la programación orientada a objetos? (<https://www.aluracursos.com/blog/poo-que-es-la-programacion-orientada-a-objetos>)
- **Artículo** Revisando la Orientación a Objetos: encapsulación de Java (<https://www.aluracursos.com/blog/revisando-la-orientacion-a-objetos-encapsulacion-de-java>)
- **Artículo** ¿Qué es encapsulamiento? (<https://www.aluracursos.com/blog/Que-es-encapsulamiento>)
- **Artículo** Herencia en JavaScript (<https://www.aluracursos.com/blog/herencia-en-javascript>)
- **Artículo** Interfaces Gráficas con Eclipse WindowBuilder (<https://www.aluracursos.com/blog/interfaces-graficas-con-eclipse-windowbuilder>)
- **Artículo** Cómo no aprender Java y Orientación a Objetos: getters y setters (<https://www.aluracursos.com/blog/como-no-aprender-java-y-orientacion-a-objetos-getters-y-setters>)
- **Artículo** Ordenar una lista de objetos en Java (<https://www.aluracursos.com/blog/ordenar-una-lista-de-objetos-en-java>)
- **Curso** Formación Java Orientado a Objetos (<https://app.aluracursos.com/formacion-javaoo>)
- **Curso** JavaScript: Introducción a la Orientación a Objetos (<https://app.aluracursos.com/course/javascript-introduccion-orientacion-objetos>)
- **Curso** JavaScript: Herencia e Interfaces en Orientación a Objetos (<https://app.aluracursos.com/course/javascript-herencia-interfaces-orientacion-objetos>)
- **Curso** C# y Orientación a Objetos (<https://app.aluracursos.com/formacion-c-sharpe-orientacion-a-objetos>)
- **Curso** Python: comprensión de la Orientación a Objetos (<https://app.aluracursos.com/course/python-comprension-orientacion-objetos>)

- **Curso** Python: avanzando en la orientación a objetos (<https://app.aluracursos.com/course/python-avanzando-orientacion-objetos>)
- **Curso** Java Polimorfismo: Entendiendo herencia e interfaces (<https://app.aluracursos.com/course/java-parte-3-entendiendo-herencia-interfaces>)
- **Curso** Curso de JavaScript: Objetos ([https://app.aluracursos.com/course/javascript-objetos?utm\\_source=gnarus&utm\\_medium=timeline](https://app.aluracursos.com/course/javascript-objetos?utm_source=gnarus&utm_medium=timeline))
- **Curso** Formación: Aprenda a programar en Python con orientación a objetos (<https://app.aluracursos.com/formacion-lenguaje-python>)

## C# - Colecciones:

- Una colección representa un grupo de objetos, conocidos como sus elementos. Son como contenedores que agrupan varios elementos en una sola unidad. Algunas colecciones permiten la duplicación de elementos y otras no. Algunas son ordenadas y otras no ordenadas.
- Aprender los usos y diferencias entre Dictionary, List, Queue, SortedList y Stack
- Aprende a trabajar con ArrayList y Hashtable
- Iterators

## Contenidos

- **Web** Colecciones (C#) (<https://learn.microsoft.com/es-es/dotnet/csharp/programming-guide/concepts/collections>)
- **Web** Tipos de colección utilizados normalmente (<https://learn.microsoft.com/es-es/dotnet/standard/collections/commonly-used-collection-types>)
- **Web** Iterators (<https://learn.microsoft.com/es-es/dotnet/csharp/iterators>)
- **Artículo** 3 maneras diferentes de recorrer una lista en C# (<https://www.arungudelli.com/es/csharp-tips/loop-through-c-sharp-list/>)
- **Artículo** Estructuras de datos en .NET con C# (<https://www.genesisrrios.com/es/blog/estructuras-de-datos-en-csharp/>)
- **Artículo** Colecciones en C#, ¿cuál elegir? (<https://thatcsharpguy.com/post/colecciones-en-c-cual-elegir>)
- **Artículo** Lists / Listas (<https://csharp.net-tutorials.com/es/389/collections-colecciones/lists-listas/>)
- **YouTube** Colecciones en C# en detalle (<https://www.youtube.com/watch?v=ViZSfgq6Cyo>)
- **YouTube** Tutorial C# nivel Avanzado 5 - Hashtable I ([https://www.youtube.com/watch?v=fFXHDib-x\\_0](https://www.youtube.com/watch?v=fFXHDib-x_0))
- **YouTube** List y ArrayList C# (<https://www.youtube.com/watch?v=k431eUiMRXA&t=374s>)

- **YouTube** Tutorial C# nivel Avanzado 8 - SortedList (<https://www.youtube.com/watch?v=3VsCtkfQMPI>)
- **YouTube** Tabla de Hash: Qué es y cómo funcionan (<https://www.youtube.com/watch?v=9tZsDJ3JBUA>)

### Contenidos Alura:

- **Curso** Curso C#: array y tipos genéricos (<https://www.aluracursos.com/curso-online-csharp-array-tipos-genericos>)

### NuGet:

- NuGet es un gestor de paquetes para la plataforma .NET. Define cómo se crean, publican y consumen los paquetes en esta plataforma, y proporciona herramientas para cada una de estas funciones.
- Gestión de paquetes
- Compartir bibliotecas

### Contenidos

- **Web** Una introducción a NuGet (<https://learn.microsoft.com/es-es/nuget/what-is-nuget>)
- **Web** Inicio rápido: Creación y publicación de un paquete NuGet con Visual Studio (solo Windows) (<https://learn.microsoft.com/es-es/nuget/quickstart/create-and-publish-a-package-using-visual-studio?tabs=netcore-cli>)
- **Web** Administre bibliotecas de proyectos con NuGet (<https://learn.microsoft.com/es-es/archive/msdn-magazine/2011/november/nuget-manage-project-libraries-with-nuget>)
- **YouTube** Instalar paquete de NuGet ([https://www.youtube.com/watch?v=J3\\_io5mBEnU&ab\\_channel=alks091](https://www.youtube.com/watch?v=J3_io5mBEnU&ab_channel=alks091))
- **YouTube** ¿Cómo publicar un paquete en NuGet? (<https://www.youtube.com/watch?v=NrgZUkloois>)
- **YouTube** Crear librería (DLL) y añadirla a proyecto con C# y .NET ([https://www.youtube.com/watch?v=JAU8Cc7jX\\_Y&ab\\_channel=Inform%C3%A1ticayProgramaci%C3%B3n](https://www.youtube.com/watch?v=JAU8Cc7jX_Y&ab_channel=Inform%C3%A1ticayProgramaci%C3%B3n))

### Contenidos Alura:

- **Curso** Curso C#: bibliotecas DLL, documentación y uso de NuGet (<https://www.aluracursos.com/curso-online-csharp-bibliotecas-dll-documentacion-nuget>)
- **Curso** Curso de C#: creación y distribución de bibliotecas .NET (<https://app.aluracursos.com/course/c-sharp-creacion-distribucion-bibliotecas-dotnet>)

## C# - System.IO:

- El espacio de nombres (namespace) System.IO consiste en clases, estructuras, delegates y enumeraciones relacionadas con la entrada y salida de datos (IO). Estas clases pueden ser utilizadas para leer y escribir datos en archivos o streams de datos. También contiene clases de soporte para archivos y directorios.
- Leer datos de archivos
- Escribir datos en archivos
- Gestionar archivos con el uso de Using

### Contenidos

- **Web** System.IO Espacio de nombres (<https://learn.microsoft.com/es-es/dotnet/api/system.io?view=net-6.0>)
- **Web** E/S de archivos y secuencias (<https://learn.microsoft.com/es-es/dotnet/standard/io/>)
- **Artículo** Manejo de archivos en C#. (<https://www.clasesdeinformaticaweb.com/csharp/manejo-de-archivos-en-c/>)
- **Artículo** Escribir un fichero de texto con C# (<https://www.discoduroderoer.es/escribir-un-fichero-de-texto-con-c-sharp/>)
- **YouTube** Crear, leer y escribir un archivo de texto ([https://www.youtube.com/watch?v=iSoixuj5wB8&ab\\_channel=CodingconC](https://www.youtube.com/watch?v=iSoixuj5wB8&ab_channel=CodingconC))
- **YouTube** Cuatro formas de usar la palabra clave USING en C# ([https://www.youtube.com/watch?v=69WSdXnvZCA&ab\\_channel=ConsejosC%23](https://www.youtube.com/watch?v=69WSdXnvZCA&ab_channel=ConsejosC%23))
- **YouTube** Cómo leer archivos en .NET con C# - Cómo crear directorios y trabajar con archivos en .NET ([https://www.youtube.com/watch?v=kTOoS5jQZdk&ab\\_channel=TheCoderCaveesp](https://www.youtube.com/watch?v=kTOoS5jQZdk&ab_channel=TheCoderCaveesp))
- **YouTube** Cómo eliminar archivos desde C# ([https://www.youtube.com/watch?v=8P-4BBgZI3I&ab\\_channel=CodingconC](https://www.youtube.com/watch?v=8P-4BBgZI3I&ab_channel=CodingconC))

### Contenidos Alura:

- **Curso** Curso C#: entrada y salida (E/S) con flujos (<https://www.aluracursos.com/curso-online-csharp-entrada-salida-flujos>)

## C# - Administración de Memoria:

- El manejo automático de memoria es uno de los servicios que el Common Language Runtime ofrece durante la Ejecución Administrada (Managed Execution). El recolector de basura (Garbage Collector) del Common Language Runtime gestiona la asignación y liberación de memoria para una aplicación.
- Comprender cómo se administra la memoria

- Conocer el Recolector de Basura (Garbage Collector)
- Entender el Stack y el Heap administrado

## Contenidos

- **Web** Gestión de Memoria Automática (<https://learn.microsoft.com/es-es/dotnet/standard/automatic-memory-management>)
- **Web** Administración de memoria y recolección de elementos no utilizados (GC) en ASP.NET Core (<https://learn.microsoft.com/es-es/aspnet/core/performance/memory?view=aspnetcore-7.0>)
- **Artículo** La Pila (The Stack) y el Montículo (The Heap) en C# .NET (<https://joffremoncayo.com/blog/la-pila-the-stack-y-el-monticulo-the-heap-en-csharp-dot-net/>)
- **YouTube** El recolector de basura o garbage collector (GC) ([https://www.youtube.com/watch?v=URap94jL0wo&ab\\_channel=GerardoPortillo%28GJParts%29](https://www.youtube.com/watch?v=URap94jL0wo&ab_channel=GerardoPortillo%28GJParts%29))

## C# - Pruebas:

- La prueba de software es el proceso de evaluación y verificación de que un software realmente hace lo que debe hacer. Los beneficios de las pruebas incluyen la prevención de errores, la reducción de los costos de desarrollo y la mejora del rendimiento.
- Usar pruebas unitarias
- Usar pruebas de integración
- Usar pruebas de comportamiento (behavior)
- Usar mocks

## Contenidos

- **Web** Tutorial: Crear y ejecutar pruebas unitarias en código administrado (<https://learn.microsoft.com/es-es/visualstudio/test/walkthrough-creating-and-running-unit-tests-for-managed-code?view=vs-2022>)
- **Web** Tutorial: Desarrollo controlado por pruebas con el Explorador de pruebas (<https://learn.microsoft.com/es-es/visualstudio/test/quick-start-test-driven-development-with-test-explorer?view=vs-2022>)
- **Artículo** ¿Qué es la prueba de software? (<https://www.ibm.com/es-es/topics/software-testing>)
- **Artículo** Pruebas unitarias con C# (<https://aspnetcoremaster.com/dotnet/2019/06/13/pruebas-unitarias-csharp.html>)



- **Artículo** PRUEBAS UNITARIAS: ¿MOCK O STUB?  
(<https://itblogsogeti.com/2015/03/26/desarrollo-pruebas-unitarias-trinitario-gomez-sogeti/>)
- **Artículo** BDD Testing. ¿Cómo funciona el Behavior Driven Development?  
(<https://www.hiberus.com/crecemos-contigo/bdd-behavior-driven-developement/>)
- **Artículo** 3 Ejemplos de TDD en .NET C# (<https://arbems.com/3-ejemplos-practicos-de-tdd-en-csharp-net/>)
- **Artículo** BDD Testing. ¿Cómo funciona el Behavior Driven Development?  
(<https://www.hiberus.com/crecemos-contigo/bdd-behavior-driven-developement/>)
- **YouTube** Introducción a pruebas unitarias en C# .Net | Unit Testing  
(<https://www.youtube.com/watch?v=lcPqONZEKAo>)
- **YouTube** UNIT TESTING en BACKEND | C# .NET ([https://www.youtube.com/watch?v=i7h92cR\\_8dA](https://www.youtube.com/watch?v=i7h92cR_8dA))
- **YouTube** MOCK en UNIT TEST para tratar con DEPENDENCIAS EN C#  
(<https://www.youtube.com/watch?v=pgCMTsVIHrg>)
- **YouTube** Introducción a Pruebas Unitarias en C# con XUNIT en .NET Core | Unit Testing desde Cero ([https://www.youtube.com/watch?v=hq8\\_XRgK6k](https://www.youtube.com/watch?v=hq8_XRgK6k))
- **YouTube** Tests de integración con C# y Docker - Con ejemplo práctico  
(<https://www.youtube.com/watch?v=sFRjgnPEEzg>)
- **YouTube** Cómo hacer pruebas automatizadas con Selenium y Visual Studio C#  
([https://www.youtube.com/watch?v=ZkHAhtUQOws&ab\\_channel=PACOMARROQUIN](https://www.youtube.com/watch?v=ZkHAhtUQOws&ab_channel=PACOMARROQUIN))

### Contenidos Alura:

- **Artículo** Tipos de pruebas: ¿cuáles son las principales y por qué utilizarlas?  
(<https://www.aluracursos.com/blog/tipos-de-pruebas-cual-utilizar>)
- **YouTube** ¿Qué es Test-Driven Development (TDD)? #AluraMás  
([https://www.youtube.com/watch?v=qA3Klo0xZbE&t=1s&ab\\_channel=AluraLatam](https://www.youtube.com/watch?v=qA3Klo0xZbE&t=1s&ab_channel=AluraLatam))

### ADO.NET:

- ADO.NET es un conjunto de clases que exponen servicios de acceso a datos para desarrolladores del .NET Framework. ADO.NET provee un conjunto rico de componentes para crear aplicaciones distribuidas y compartir datos. Forma parte del .NET Framework, proporcionando acceso a datos de aplicaciones relacionales y XML.
- Manipular bases de datos
- Conocer el DataSet y DataTable
- Realizar conexiones
- Manipular documentos XML

## Contenidos

- **Web** ADO.NET (<https://learn.microsoft.com/es-es/dotnet/framework/data/adonet/>)
- **Web** Información General - ADO.NET (<https://learn.microsoft.com/es-es/dotnet/framework/data/adonet/ado-net-overview>)
- **Web** Desarrollo de aplicaciones ADO.NET (<https://www.ibm.com/docs/es/db2/11.5?topic=da-adonet>)
- **Web** Objetos DataSet, DataTable y DataView (<https://learn.microsoft.com/es-es/dotnet/framework/data/adonet/dataset-datatable-dataview/>)
- **Artículo** ¿Qué es ADO.NET? Bases de datos con ADO.NET ([https://my-entertainment-place.blogspot.com/2020/03/que-es-adonet-bases-de-datos-con-adonet.html#google\\_vignette](https://my-entertainment-place.blogspot.com/2020/03/que-es-adonet-bases-de-datos-con-adonet.html#google_vignette))
- **Artículo** Diferencia entre DataSet y DataTable (<https://www.kyocode.com/2019/04/diferencia-entre-dataset-y-datatable/>)
- **Artículo** XML en la plataforma .NET (<http://elvex.ugr.es/decsai/csharp/xml/xml-plat-net.xml>)
- **Artículo** Como Escribir Un Fichero En XML En C# (<https://www.discoduroderoer.es/como-escribir-un-fichero-en-xml-en-c-sharp/>)
- **YouTube** Introducción a .NET con MySQL - Crea una API Rest con MySQL - ([https://www.youtube.com/watch?v=E75IOkduaFg&ab\\_channel=TheCoderCaveesp](https://www.youtube.com/watch?v=E75IOkduaFg&ab_channel=TheCoderCaveesp))

## Entity Framework Core:

- Entity Framework Core es un mapeador objeto-relacional (ORM). El mapeo objeto-relacional es una técnica que permite a los desarrolladores trabajar con datos de manera orientada a objetos, realizando el trabajo necesario para mapear entre objetos definidos en un lenguaje de programación de la aplicación y los datos almacenados en fuentes de datos relacionales.
- Manipular bases de datos usando objetos .NET
- Crear modelos correspondientes a una base de datos
- Realizar consultas

## Contenidos

- **Web** Entity Framework Core (<https://learn.microsoft.com/es-es/ef/core/>)
- **Web** Administración de esquemas de base de datos (<https://learn.microsoft.com/es-es/ef/core/managing-schemas/>)
- **Web** Consulta de datos (<https://learn.microsoft.com/es-es/ef/core/querying/>)

- **Artículo** Introducción completa a Entity Framework Core (<https://estradawebgroup.com/Post/Introduccion-completa-a-Entity-Framework-Core-Simplifica-la-interaccion-con-bases-de-datos-en-tus-proyectos-ASP-NET-Core/20702#top>)
- **YouTube** Como hacer un modelo de datos con Base de Datos SQL Server en ASP.NET Core 3.1 Visual Studio 2019 ([https://www.youtube.com/watch?v=FSKgwTpYf-I&ab\\_channel=AguilarSystemsMX](https://www.youtube.com/watch?v=FSKgwTpYf-I&ab_channel=AguilarSystemsMX))
- **YouTube** ¿Cómo utilizar Entity Framework con 2 o más bases de datos distintas? ([https://www.youtube.com/watch?v=rWgMMTP9GZ0&ab\\_channel=hdeleon.net](https://www.youtube.com/watch?v=rWgMMTP9GZ0&ab_channel=hdeleon.net))
- **YouTube** Entity Framework: Consultas Tipo SQL ([https://www.youtube.com/watch?v=ry\\_pCvkkilI&ab\\_channel=CodeStack](https://www.youtube.com/watch?v=ry_pCvkkilI&ab_channel=CodeStack))
- **YouTube** Entity Framework: Consultas con ADO.Net ([https://www.youtube.com/watch?v=erEe2NoAH1U&ab\\_channel=CodeStack](https://www.youtube.com/watch?v=erEe2NoAH1U&ab_channel=CodeStack))

## Estructura de Datos:

- En el contexto de los ordenadores, una estructura de datos es una forma específica de almacenar y organizar los datos en la memoria del ordenador para que esos datos puedan ser fácilmente recuperados y utilizados de forma eficiente cuando sea necesario posteriormente.
- Conocer las principales estructuras de datos
- Implementar las principales estructuras de datos

## Contenidos

- **Artículo** ¿Qué es una estructura de datos en programación y para qué se utiliza? (<https://blog.soyhenry.com/que-es-una-estructura-de-datos-en-programacion/>)
- **Web** Estructuras de datos - Documentación Python (<https://docs.python.org/es/3/tutorial/datastructures.html>)
- **Artículo** Estructuras de Datos y Algoritmos en Java ([https://programacion.net/articulo/estructuras\\_de\\_datos\\_y\\_algoritmos\\_en\\_java\\_309/4](https://programacion.net/articulo/estructuras_de_datos_y_algoritmos_en_java_309/4))
- **Artículo** Estructuras de datos con Java: un enfoque práctico (<http://hp.fciencias.unam.mx/~alg/estructurasDeDatos/>)
- **Artículo** Colecciones: ARRAY, SET y DICTIONARY en Swift (<https://www.swiftbeta.com/colecciones-array-set-y-dictionary-en-swift/>)
- **Artículo** Colecciones en Swift y su manejo de memoria (<https://medium.com/@grago/colecciones-en-swift-y-su-manejo-de-memoria-61a03e236dd>)

- **Artículo** Estructuras de datos en .NET con C# (<https://www.genesisrrios.com/es/blog/estructuras-de-datos-en-csharp/>)
- **YouTube** Estructuras de Datos | Primeros Pasos (<https://youtu.be/Df-sgxGzyTg>)
- **YouTube** Qué son las estructuras de datos (<https://youtu.be/oQ0Wkldr73E?list=PLTd5ehlj0goMTSK7RRAPBF4wP-Nj5DRvT>)
- **YouTube** ¿Qué son y cómo funcionan los árboles? | Ejemplo de implementación (<https://youtu.be/tBaOQeyXYqg>)
- **YouTube** Aprende: Estructura de Datos con Java ([https://www.youtube.com/watch?v=\\_9ScDWpqhFE](https://www.youtube.com/watch?v=_9ScDWpqhFE))
- **YouTube** Introducción - 1 - Estructuras de Datos en C# ([https://www.youtube.com/watch?v=rqaqJXbauyA&ab\\_channel=nicosiored](https://www.youtube.com/watch?v=rqaqJXbauyA&ab_channel=nicosiored))
- **YouTube** Estructuras de datos con Python en 8 minutos: Listas, Tuplas, Conjuntos y Diccionarios (<https://www.youtube.com/watch?v=v25-m1LOUiU>)
- **YouTube** Programación en Python | Colecciones | Diccionarios (<https://www.youtube.com/watch?v=vAy4IM7NLIQ>)

#### Contenidos Alura:

- **Artículo** Estructura de datos: introducción (<https://www.aluracursos.com/blog/estructura-de-datos-introduccion>)
- **Artículo** Conociendo tuplas en Python (<https://www.aluracursos.com/blog/conociendo-las-tuplas-en-python>)
- **Artículo** Listas en Python: operaciones básicas (<https://www.aluracursos.com/blog/listas-de-python-operaciones-basicas>)
- **Artículo** Estructura de datos: computación práctica con Java (<https://www.aluracursos.com/blog/estructura-de-datos-computacion-practica-con-java>)
- **Artículo** Python: trabajando con diccionarios (<https://www.aluracursos.com/blog/python-trabajando-con-diccionarios>)
- **Artículo** Iterando una lista en Java (<https://www.aluracursos.com/blog/iterando-una-lista-en-java>)
- **Curso** Python: avanzando en el lenguaje (<https://app.aluracursos.com/course/python-avanzando-lenguaje>)
- **Curso** C#: array y tipos genéricos (<https://app.aluracursos.com/course/csharp-array-tipos-genericos>)
- **Curso** Python Collections: listas y tuplas ([https://app.aluracursos.com/course/python-collections-listas-tuplas?utm\\_source=gmarus&utm\\_medium=timeline](https://app.aluracursos.com/course/python-collections-listas-tuplas?utm_source=gmarus&utm_medium=timeline))

- **Curso** Curso de Java: trabajar con listas y colecciones de datos (<https://www.aluracursos.com/curso-online-java-trabajar-listas-colecciones-datos>)
- **Curso** Java y java.util: Colecciones, Wrappers y Lambda expressions (<https://www.aluracursos.com/curso-online-java-util-colecciones-wrappers-lambda-expressions>)

## Nivel 2

### CLR:

- Common Language Runtime (CLR) es el componente de máquina virtual de la plataforma .NET de Microsoft que administra la ejecución de programas .NET.
- Entender cómo funciona CLR
- Entender la gestión de memoria
- Conocer la CIL y el JIT

### Contenidos

- **Web** Introducción a Common Language Runtime (<https://learn.microsoft.com/es-es/dotnet/standard/clr>)
- **Web** Ejemplos de funciones CLR de C# .NET (<https://www.ibm.com/docs/es/db2/11.5?topic=routines-example-c-net-clr-functions>)
- **Web** Optimización y depuración JIT (<https://learn.microsoft.com/es-es/visualstudio/debugger/jit-optimization-and-debugging?view=vs-2022>)
- **Artículo** ¿Qué es el compilador Just-In-Time (JIT) en .NET? (<https://barcelongeeks.com/que-es-el-compilador-just-in-time-jit-en-net/>)
- **YouTube** Curso de C# 001 - ¿Qué es C#, .NET y CLR? (<https://www.youtube.com/watch?v=VWQRFyYPtgc>)

### LINQ:

- LINQ (Language-Integrated Query) es el nombre de un conjunto de tecnologías basado en la integración de recursos de consulta directamente en el lenguaje C#.
- Crear consultas
- Conocer las cláusulas Select y Where
- Consultar colecciones de objetos en memoria
- Mapear la base de datos con LINQ to Sql

## Contenidos

- **Web** Introducción a las consultas LINQ (C#) (<https://learn.microsoft.com/es-es/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries>)
- **Web** Qué puede hacer con LINQ to SQL (<https://learn.microsoft.com/es-es/dotnet/framework/data/adonet/sql/linq/what-you-can-do-with-linq-to-sql>)
- **Web** Procedimiento para rellenar colecciones de objetos de varios orígenes (LINQ) (C#) (<https://learn.microsoft.com/es-es/dotnet/csharp/programming-guide/concepts/linq/how-to-populate-object-collections-from-multiple-sources-linq>)
- **Web** Tutorial: Usar solo procedimientos almacenados (C#) (<https://learn.microsoft.com/es-es/dotnet/framework/data/adonet/sql/linq/walkthrough-using-only-stored-procedures-csharp>)
- **Artículo** LINQ TO SQL, CONEXIÓN A BASES DE DATOS (<https://www.attomgroup.com/linq-to-sql-conexion-a-bases-de-datos/>)
- **YouTube** Consultas Linq en Query Syntax y Method Syntax en .net con c# (<https://www.youtube.com/watch?app=desktop&v=bGborT4e1Fc>)
- **YouTube** 5 Métodos poco conocidos de LINQ en C# .Net (<https://www.youtube.com/watch?v=jhEXqwG0xM8>)
- **YouTube** Uso de LINQPad para Ejecución de Expresiones C# (<https://www.youtube.com/watch?v=g5NZhspKfy8>)

## Contenidos Alura:





- **Curso** Curso C#: List, Lambda, Linq (<https://app.aluracursos.com/course/c-sharp-list-lambda-linq>)

## C# - Serialización:

- Serialización es el proceso de convertir un objeto en un flujo de bytes para almacenar el objeto o transmitirlo a la memoria, a una base de datos o a un archivo. Su objetivo principal es guardar el estado de un objeto para poder recrearlo cuando sea necesario.
- Enviar un objeto a una aplicación remota mediante un servicio web
- Pasar un objeto como una cadena JSON o XML
- Pasar información específica del usuario o de seguridad entre aplicaciones

## Contenidos

- **Web** Serialización en .NET | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/standard/serialization/>)
- **Web** Serialización XML con servicios Web XML (<https://learn.microsoft.com/es-es/dotnet/standard/serialization/xml-serialization-with-xml-web-services>)

- **Artículo** Serializar/deserializar con Json.Serialization (ejemplo para C#)  
(<https://www.elguillemola.com/serializar-deserializar-con-json-serialization-ejemplo-para-c/>)
- **Artículo** #HowTo Serializar un objeto en {} con C# y Newtonsoft.  
(<https://happydevops.com/2022/03/10/howto-serializar-un-objeto-en-con-c-y-newtonsoft/>)
- **YouTube**  APRENDE  que es la  SERIALIZACIÓN  y CÓMO aplicarla en C#  
(<https://www.youtube.com/watch?v=hGPUc49TrC8>)
- **YouTube** Tutorial C# nivel intermedio 17 --- Serialización binaria  
([https://www.youtube.com/watch?v=xq7OGMoC\\_1c](https://www.youtube.com/watch?v=xq7OGMoC_1c))
- **YouTube** Tutorial C# nivel intermedio 19 --- Serialización con composición  
(<https://www.youtube.com/watch?v=7XMj9d2m-F8>)
- **YouTube** Derek Banas: C# Serialization (inglés) (<https://www.youtube.com/watch?v=jbwjbbc5PjI>)

## C# - Red y Sockets:

- El término programación con sockets se refiere a la escritura de programas que se ejecutan en varios ordenadores en los que los dispositivos están todos conectados entre sí utilizando una red.
- Abrir una sesión de comunicación interactiva entre el navegador del usuario y un servidor
- Enviar mensajes a un servidor y recibir respuestas sin consultar al servidor

## Contenidos

- **Web** Sockets en .NET (<https://learn.microsoft.com/es-es/dotnet/fundamentals/networking/sockets/sockets-overview>)
- **Web** Uso de sockets para enviar y recibir datos a través de TCP  
(<https://learn.microsoft.com/es-es/dotnet/fundamentals/networking/sockets/socket-services>)
- **Web** Uso de Visual C# para crear un servidor remoto (<https://learn.microsoft.com/es-es/troubleshoot/developer/visualstudio/csharp/language-compilers/create-remote-server>)
- **Artículo** Escribiendo un servidor WebSocket en C#  
([https://developer.mozilla.org/es/docs/Web/API/WebSockets\\_API/Writing\\_WebSocket\\_server](https://developer.mozilla.org/es/docs/Web/API/WebSockets_API/Writing_WebSocket_server))
- **YouTube** Crear nuestra Conexión al Servidor de datos | C# Nivel Intermedio  
(<https://www.youtube.com/watch?v=VKe3SzeQLuc>)
- **YouTube** Curso de Sockets en C# - Cliente/Servidor Sincrónico [2/?]  
([https://www.youtube.com/watch?v=F\\_cX662Dp1M](https://www.youtube.com/watch?v=F_cX662Dp1M))

## ASP.NET Core:

- ASP.NET Core es un framework de código abierto y multiplataforma para la construcción de aplicaciones basadas en la nube, tales como aplicaciones web, aplicaciones IoT y backends móviles.
- Crear aplicaciones y servicios web
- Mantener una aplicación MVC
- Desarrollar interfaz de usuario web del lado del cliente
- Crear una API web

## Contenidos

- **Web** Información general de ASP.NET Core | Microsoft Learn (<https://learn.microsoft.com/es-es/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-6.0>)
- **Web** Documentación de ASP.NET | Microsoft Learn (<https://learn.microsoft.com/es-es/aspnet/core/?view=aspnetcore-7.0>)
- **Web** Desarrollo de aplicaciones ASP.NET Core MVC | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/architecture/modern-web-apps-azure/develop-asp-net-core-mvc-apps>)
- **Artículo** Ventajas y desventajas de ASP.NET Core (<https://redwerk.es/blog/ventajas-y-desventajas-de-asp-net-core/>)
- **Artículo** Trabajando con servicios web a través de ASP.NET Core y DotVVM (<https://dev.to/dotvvm/trabajando-con-servicios-web-a-traves-de-asp-net-core-y-dotvvm-4c76>)
- **Artículo** Migrar desde ASP.NET .Net Core 2.x web API a .Net Core 3.1 (<https://pablotolentino.medium.com/migrar-desde-asp-net-net-core-2-x-web-api-a-net-core-3-1-25ef8485f6bb>)
- **Artículo** ASP.NET Core 6: Creando una app Multi-tenant (Parte 1) (<https://dev.to/isaacojeda/asp-net-core-6-creando-una-app-multi-tenant-parte-1-3df5>)
- **YouTube** Controladores en ASP NET Core MVC (<https://www.youtube.com/watch?v=HSAp0zrgpHU>)
- **YouTube** Creando proyecto de ASP.NET Core MVC en Mac OS (<https://www.youtube.com/watch?v=448jXb4TQlg>)

## Dapper:

- Dapper es un producto de mapeo objeto-relacional (ORM) para la plataforma Microsoft .NET. Proporciona un framework para asignar un modelo de dominio orientado a objetos a



bases de datos relacionales.

- Realizar consultas a bases de datos como 'select', 'insert', 'update', 'delete'
- Manipular bases de datos

## Contenidos

- **Web** Uso de la biblioteca cliente de bases de datos elásticas con Dapper (<https://learn.microsoft.com/es-es/azure/azure-sql/database/elastic-scale-working-with-dapper?view=azuresql-db>)
- **Artículo** Dapper - SQL (<https://compuarticulos.comotico.com/?p=20>)
- **Artículo** Dapper vs EntityFramework, debo usar un MicroOrm o un Orm? (<http://www.eltavo.net/2016/04/dapper-vs-entityframework-debo-usar-un.html>)
- **Artículo** [C#][SQL] [EF] [Dapper] Uso de Sequence de Sql Server desde .NET (<https://medium.com/jtorrecilla-net/c-sql-ef-dapper-uso-de-sequence-de-sql-server-desde-net-adb2eaa1cdeb>)
- **Artículo** Dapper y Entity Framework Core en .NET 6 (<https://arbems.com/dapper-ef-core-net-6/>)
- **YouTube** Qué es y cómo usar Dapper en .Net para hacer llamadas a bases de datos (<https://www.youtube.com/watch?v=7bfALdBdRTA>)
- **YouTube** Agregando metodo update en c# | Visual Studio 2017 | Angularjs ([https://www.youtube.com/watch?v=M\\_rFCTxrRtY](https://www.youtube.com/watch?v=M_rFCTxrRtY))

## Inyección de Dependencias:

- La inyección de dependencias es un patrón de diseño en el que una clase solicita dependencias de fuentes externas en lugar de crearlas.
- Evitar el alto nivel de acoplamiento de código dentro de una aplicación
- Implementar la inversión de control

## Contenidos

- **Web** Inserción de dependencias .NET | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/core/extensions/dependency-injection>)
- **Web** loc (inversión de control) | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/communitytoolkit/mvvm/ioc>)
- **Artículo** Inyección de dependencias ¿Qué es y para qué sirve? (<https://bravedeveloper.com/2021/04/21/inyeccion-de-dependencias-que-es-y-para-que-sirve/>)
- **Artículo** Inyección de Dependencias(DI) en ASP.NET Core, mejores prácticas para escribir código reutilizable, escalable y desacoplado (<https://dev.to/ebarrioscode/inyeccion-de->)

[dependencias-di-en-asp-net-core-mejores-practicas-para-escribir-codigo-reutilizable-escalable-y-desacoplado-kho\)](#)

- **Artículo** Inyección de Dependencias en ASP.NET Core - III (<https://geeks.ms/jorge/2019/02/09/inyeccion-de-dependencias-en-asp-net-core-iii/>)
- **Artículo** Entity Core 2.0— Inyección de dependencias y Control de inversión (<https://devsebastianvz.medium.com/entity-core-2-0-inyecci%C3%B3n-de-dependencias-y-control-de-inversi%C3%B3n-61b59cc390e9>)
- **YouTube** INYECCIÓN DE DEPENDENCIAS 🖋 - scoped vs transient vs singleton ([https://www.youtube.com/watch?v=1FojQ\\_ZjqOI](https://www.youtube.com/watch?v=1FojQ_ZjqOI))
- **YouTube** Tipos de Inyección de dependencias en ASP .Net | Transient vs Scoped vs Singleton (<https://www.youtube.com/watch?v=srPGwwMwAoA>)
- **YouTube** Inyección de dependencias en ASP.NET Core (<https://www.youtube.com/watch?v=U2TvawZ0Ym4>)

## C# - Multithreading:

- Multithreading es la capacidad de realizar múltiples operaciones al mismo tiempo. Las operaciones con el potencial de retrasar otras operaciones se pueden realizar en subprocesos separados.
- Realizar múltiples tareas simultáneamente
- Entender cómo se ejecutan los hilos
- Aprender cómo hacer que un hilo espere en un punto específico

## Contenidos

- **Web** Uso de subprocesos y subprocesamiento | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/standard/threading/using-threads-and-threading>)
- **Web** Programación asincrónica basada en tareas | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/standard/parallel-programming/task-based-asynchronous-programming>)
- **Artículo** Multithreading: más potencia para los procesadores (<https://www.ionos.es/digitalguide/servidores/know-how/explicacion-del-multithreading/>)
- **Artículo** CONCURRENCIA EN C# - PARTE 1 (<https://fmoralesdev.com/es/2019/01/26/concurrency-in-c-part-1/>)
- **Artículo** Píldoras de C#: Task Parallel Library (TPL) Procesamiento Multihilo y asíncrono (<https://dev.to/ebarrioscode/pildoras-de-c-task-parallel-library-tpl-procesamiento-multihilo-y-asincrono-18mp>)

- **YouTube** Múltiples hilos - 2 - Programación Multihilos C#  
([https://www.youtube.com/watch?v=iJ6NdIpkLFk&list=PLM-p96nOrGca0DgG\\_aRjkBv5awLUoca\\_d&index=2](https://www.youtube.com/watch?v=iJ6NdIpkLFk&list=PLM-p96nOrGca0DgG_aRjkBv5awLUoca_d&index=2))

#### Contenidos Alura:

- **Artículo** Concurrencia o Paralelismo: Threads, Processes, Fibers y Actors  
(<https://www.aluracursos.com/blog/concurrencia-o-paralelismo>)

## Nivel 3

### C# - Delegates y Eventos:

- Delegates son objetos que se usan como punteros de función para referirse a un método asignado a ellos.
- Los eventos son acciones que cambian el estado de un objeto. Los eventos se declaran utilizando delegates - ellos proporcionan una encapsulación a los delegates.
- Entender el concepto de delegate
- Crear una referencia para una función con una cierta lista de parámetros
- Entender el concepto de evento
- Manipular diferentes tipos de eventos

#### Contenidos

- **Web** Introducción a delegados y eventos en C# (<https://learn.microsoft.com/es-es/dotnet/csharp/delegates-overview>)
- **Web** Distinción de delegados y eventos (<https://learn.microsoft.com/es-es/dotnet/csharp/distinguish-delegates-events>)
- **Web** Control y generación de eventos (<https://learn.microsoft.com/es-es/dotnet/standard/events/>)
- **Artículo** Eventos en C# (<https://thatcsharpguy.com/post/events-en-c>)
- **YouTube** EVENTOS en C# - Tutorial y Explicación con ejemplos  
(<https://www.youtube.com/watch?v=wnr5eY1yUjI>)
- **YouTube** Delegados en C# | ¿Cómo funcionan? 🤖 🏠 (<https://www.youtube.com/watch?v=TWbFGoT-7Rc>)
- **YouTube** C# desde Cero - Episodio 06 Enums, Delegados y Eventos  
(<https://www.youtube.com/watch?v=jbeyA84ZlZE>)

## C# - Métodos anónimos y lambda expressions:

- Los métodos anónimos son métodos sin nombre que se pueden definir usando la palabra clave delegate.
- Lambda Expressions se utilizan como funciones anónimas, pero no es necesario especificar el tipo de valor que se escribe, lo que las hace más flexibles de usar.
- Crear funciones anónimas que puedes utilizar para crear delegates
- Crear roles locales que se pueden pasar como argumentos

### Contenidos

- **Web** Expresiones lambda y funciones anónimas | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/operators/lambda-expressions>)
- **Web** Delegados y expresiones lambda | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/standard/delegates-lambdas>)
- **Web** Delegados con métodos con nombre y Métodos anónimos (Guía de programación de C#) | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/csharp/programming-guide/delegates/delegates-with-named-vs-anonymous-methods>)
- **Artículo** Métodos anónimos y expresiones lambda en C# (<https://geeks.ms/jorge/2017/09/21/metodos-anonimos-y-expresiones-lambda-en-c/>)
- **Artículo** LINQ Y LAMBDA EXPRESSIONS (<https://gamecodetips.com/2017/09/17/linq-y-lambda-expressions/>)
- **YouTube** Expresiones Lambda en C# (<https://www.youtube.com/watch?v=Y5q3wse1tDw>)

### Contenidos Alura:

- **Curso** Curso C#: List, Lambda, Linq (<https://app.aluracursos.com/course/c-sharp-list-lambda-linq>)

### Contenedores:

- Los contenedores son paquetes de software que contienen todos los elementos necesarios para ejecutarse en cualquier entorno. La gestión de contenedores es un área crucial en la computación en nube y DevOps, que implica el uso de tecnologías para automatizar el proceso de creación, implementación, escalado y monitoreo de contenedores. Los contenedores son unidades de software estandarizadas que permiten a los desarrolladores empaquetar todas las dependencias de una aplicación (código, bibliotecas, configuraciones, etc.) en un solo paquete. Esto permite que la aplicación se ejecute de forma consistente en cualquier entorno de infraestructura.

- La tecnología de contenedores, como ejemplifica Docker, proporciona un entorno coherente y portátil para el desarrollo, las pruebas y la implementación de aplicaciones, lo que es vital para el trabajo eficiente de la ingeniería de datos. Además, Kubernetes, un sistema de organización de contenedores, permite la gestión, automatización y escalabilidad de aplicaciones basadas en contenedores en entornos de producción. Dominar estos conceptos y tecnologías permite a los ingenieros de datos construir y mantener canalizaciones de datos eficientes y confiables.
- Kubernetes (también conocido como k8s o Kube) es una plataforma de orquestación de contenedores de código abierto que automatiza gran parte de los procesos manuales necesarios para implementar, gestionar y escalar aplicaciones en contenedores.
- Aislar el software para que funcione independientemente
- Implementación de software en clústeres
- Modularizar su sistema en paquetes más pequeños
- Conocer la plataforma Docker
- Conocer Kubernetes

## Contenidos

- **Web** IBM - ¿Qué son los contenedores? (<https://www.ibm.com/mx-es/topics/containers>)
- **Web** Microsoft - ¿Qué es un contenedor? (<https://azure.microsoft.com/es-mx/resources/cloud-computing-dictionary/what-is-a-container>)
- **Artículo** Contenedores y la nube (<https://dev.to/roremdev/contenedores-y-la-nube-3je9>)
- **Artículo** Contenedores: cómo es el ciclo de vida de una aplicación en Kubernetes (<https://dev.to/campusmvp/contenedores-como-es-el-ciclo-de-vida-de-una-aplicacion-en-kubernetes-1ief>)
- **Artículo** ¿Qué diferencia hay entre Docker (Contenedores) y Máquinas virtuales (VMWare, VirtualBox...)? (<https://dev.to/campusmvp/qu-diferencia-hay-entre-docker-contenedores-y-mquinas-virtuales-vmware-virtualbox-4ji3>)
- **Artículo** El potencial de Kubernetes para las empresas (<https://jlcasal.medium.com/el-potencial-de-kubernetes-para-las-empresas-d36f9eece999>)
- **Artículo** ¿Qué es Docker y para que sirve? Explicación ([https://dev.to/prox\\_sea/que-es-docker-y-para-que-sirve-explicacion-5h2n](https://dev.to/prox_sea/que-es-docker-y-para-que-sirve-explicacion-5h2n))
- **YouTube** ¿Que es un contenedor? (<https://www.youtube.com/watch?v=x5zGoICZLnU>)
- **YouTube** ¿Qué es Docker y los contenedores? (<https://www.youtube.com/watch?v=kkfZs0vJFyU>)
- **YouTube** Volúmenes y redes en Docker ([https://www.youtube.com/watch?v=DIdeI70dFI&ab\\_channel=Programaci%C3%B3nnespa%C3%B1ol](https://www.youtube.com/watch?v=DIdeI70dFI&ab_channel=Programaci%C3%B3nnespa%C3%B1ol))

- **YouTube** ¿Qué es KUBERNETES?  Relación con DOCKER y CONTENEDORES   
(<https://www.youtube.com/watch?v=V86eTbswdQo>)

### Contenidos Alura:

- **Artículo** Empezando con Docker (<https://www.aluracursos.com/blog/empezando-con-docker>)
- **Artículo** Creando volúmenes con Docker (<https://www.aluracursos.com/blog/creando-volumenes-con-docker>)

### Arquitectura de Microservicios:

- Los microservicios son un enfoque de arquitectura en el que el software consiste en pequeños servicios independientes que se comunican entre sí y se organizan de acuerdo con sus dominios de negocio.
- Aprender el concepto de arquitectura diseñada para microservicios
- Realizar la comunicación mediante API
- Mejorar la escalabilidad de un sistema

### Contenidos

- **Web** Diseño de una aplicación orientada a microservicios (<https://learn.microsoft.com/es-es/dotnet/architecture/microservices/multi-container-microservice-net-applications/microservice-application-design>)
- **Artículo** Microservicios y arquitectura de microservicios  
(<https://www.intel.la/content/www/xl/es/cloud-computing/microservices.html>)
- **Artículo** Microservicios vs API: Entendiendo la diferencia  
(<https://kinsta.com/es/blog/microservicios-vs-api/>)
- **Artículo** Microservicios Ejemplo de Flujo  
(<https://medium.com/mycodebad/microservicios-ejemplo-de-flujo-f45720a9b278>)
- **Artículo** Microservicios Conceptos (<https://medium.com/mycodebad/microservicios-conceptos-55d19636873e>)
- **YouTube** Patrones fundamentales de la arquitectura microservicios  
(<https://www.youtube.com/watch?v=A7y23uU1NHk>)
- **YouTube** Un ejemplo de microservicios #CafeConRivas (<https://www.youtube.com/watch?v=qAcUGw7HhxM>)

### Reflection y atributos:

- Los objetos de Reflection (reflexión) se utilizan para obtener información del tipo en tiempo de ejecución. Las clases que dan acceso a los metadatos de un programa en

ejecución se encuentran en el espacio de nombres System.Reflection.

- Escribir código que lee la información y metadatos de objetos en tiempo de ejecución
- Obtener nombres de clases en tiempo de ejecución y crear objetos de una clase

## Contenidos

- **Web** Acceso a atributos mediante reflexión | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/csharp/advanced-topics/reflection-and-attributes/accessing-attributes-by-using-reflection>)
- **Web** System.Reflection Espacio de nombres | Microsoft Learn (<https://learn.microsoft.com/es-es/dotnet/api/system.reflection?view=net-7.0>)
- **Web** Reflexión en .NET (<https://learn.microsoft.com/es-es/dotnet/framework/reflection-and-codedom/reflection>)
- **Web** Reflexión - Documentación Python (<https://docs.python.org/es/3.11/c-api/reflection.html>)
- **Web** Inmersión al modo interactivo - Python (<https://entrenamiento-python-basico.readthedocs.io/es/3.7/leccion2/interactivo.html#introspeccion-en-python>)
- **Artículo** ¿Qué es la Metaprogramación en JavaScript? En español, por favor. (<https://www.freecodecamp.org/espanol/news/que-es-la-metaprogramacion-en-javascript-en-espanol-por-favor/>)
- **Artículo** Reflection en Java (<https://jarroba.com/reflection-en-java/>)
- **YouTube** Ejemplo Java Reflection (<https://www.youtube.com/watch?v=3ZGYEDV9PIA>)
- **YouTube** Tutorial C# nivel Avanzado 46 - Reflection/Reflexión (<https://www.youtube.com/watch?v=9kAd4WI0hgQ>)

## MAUI:

- .NET Multi-platform App UI (.NET MAUI) es un framework multiplataforma para crear aplicaciones nativas móviles y de escritorio con C# y XAML.
- Crear aplicaciones móviles y de escritorio nativas con C# y XAML.
- Desarrollar aplicaciones multiplataforma.
- Compartir el diseño y la interfaz de usuario entre plataformas.

## Contenidos

- **Web** ¿Qué es .NET MAUI? (<https://learn.microsoft.com/es-es/dotnet/maui/what-is-maui>)
- **Web** Introducción a XAML (<https://learn.microsoft.com/es-es/dotnet/maui/xaml/fundamentals/get-started>)



- **Artículo** Qué es .NET MAUI y cuáles son sus ventajas (<https://www.linkedin.com/pulse/qu%C3%A9-es-net-maui-y-cu%C3%A1les-son-sus-ventajas-jordi-parareda/?originalSubdomain=es>)
- **Artículo** .NET MAUI: La apuesta de Microsoft para el desarrollo de Apps multiplataforma (<https://www.zimaltec.es/blog/net-maui-la-apuesta-de-microsoft-para-el-desarrollo-de-apps-multiplataforma>)
- **Artículo** Desarrollo de aplicaciones multiplataforma con Microsoft .NET MAUI (<https://www.integratecnologia.es/la-innovacion-necesaria/desarrollo-de-aplicaciones-multiplataforma-con-microsoft-net-maui/>)
- **YouTube** Creando un proyecto de .NET MAUI con Visual Studio - .NET MAUI (<https://www.youtube.com/watch?v=ld6WBvhg5vo>)
- **YouTube** Ejercicio creando tu primera página con XAML - .NET MAUI (<https://www.youtube.com/watch?v=FLDFJbqgVF4>)

## Habilidad Auxiliar: Infraestructura

### Git y GitHub - Fundamentos:

- Git es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia.
- GitHub es un servicio de hosting para el desarrollo de software y el control de versiones mediante Git.
- Crear un repositorio
- Clonar un repositorio
- Comprometerse, empujar y tirar hacia y desde el repositorio
- Revertir un commit
- Crear de ramas y Pull requests
- Manejar fusiones y conflictos

### Contenidos

- **Web** Git: Libro de Consulta (<https://git-scm.com/book/es/v2>)
- **Web** GitHub Documentación (<https://docs.github.com/es>)
- **Web** Github Pages Documentación (<https://docs.github.com/es/pages/getting-started-with-github-pages/about-github-pages>)
- **Web** W3Schools: Git Tutorial (<https://www.w3schools.com/git/default.asp?remote=github>)
- **Web** Git School - Visualizing Git (<https://git-school.github.io/visualizing-git/>)



- **Web** Dangit, Git!?! (<https://dangitgit.com/es>)
- **Artículo** Git and Github Quickstart Tutorial (<https://medium.com/@prashantramnyc/git-and-github-quickstart-tutorial-654a71594dca>)
- **YouTube** ¿Qué es Git y cómo funciona? (<https://www.youtube.com/watch?v=jGehuhFhtnE>)
- **YouTube** Git y Github | Guía Práctico de Git y Github Desde Cero (<https://www.youtube.com/watch?v=HiXLkL42tMU>)

## Contenidos Alura:

- **Artículo** Git y Github: que son y primeros pasos (<https://www.aluracursos.com/blog/git-y-github-que-son-y-primeros-pasos>)
- **Artículo** Guía sobre cómo instalar Git en diferentes sistemas operativos (<https://www.aluracursos.com/blog/guia-sobre-como-instalar-git-en-diferentes-sistemas-operativos>)
- **Artículo** Iniciando un repositorio con Git (<https://www.aluracursos.com/blog/iniciando-repositorio-con-git>)
- **Artículo** Comenzando con Git: aprendiendo a versionar (<https://www.aluracursos.com/blog/comenzando-con-git>)
- **Artículo** Creando un repositorio remoto en GitHub (<https://www.aluracursos.com/blog/creando-repositorio-remoto-en-github>)
- **Artículo** Clonando un repositorio con Git y GitHub (<https://www.aluracursos.com/blog/clonando-un-repositorio-remoto>)
- **Artículo** Paso a Paso para activar tu proyecto en GitHub Pages. (<https://www.aluracursos.com/blog/github-pages>)
- **Artículo** Cómo escribir un README increíble en tu Github (<https://www.aluracursos.com/blog/como-escribir-un-readme-increible-en-tu-github>)
- **Artículo** Buenas practicas en git: evitando errores (<https://www.aluracursos.com/blog/como-evitar-errores-en-git>)
- **Artículo** GIT: Errores de comandos y directorios (<https://www.aluracursos.com/blog/git-errores-de-errores-de-comandos-y-directorios>)
- **Artículo** GIT: errores de commits (<https://www.aluracursos.com/blog/git-errores-de-commits>)
- **Artículo** GIT: Errores de fusión (<https://www.aluracursos.com/blog/errores-de-fusion>)
- **Artículo** GIT: Errores con el remoto (<https://www.aluracursos.com/blog/errores-con-el-remoto>)
- **YouTube** Git y GitHub para Principiantes #AluraMás ([https://www.youtube.com/watch?v=-LmFK6skG7s&ab\\_channel=AluraLatam](https://www.youtube.com/watch?v=-LmFK6skG7s&ab_channel=AluraLatam))

- **YouTube** Git y GitHub: Herramientas Esenciales para el Control de Versiones y Colaboración en Desarrollo (<https://youtu.be/dw04N616Abw>)
- **YouTube** ¿Puedo subir mi proyecto a Github sin líneas de comando? - Git y Github para principiantes ([https://www.youtube.com/watch?v=Yfm16Tlpcwk&ab\\_channel=AluraLatam](https://www.youtube.com/watch?v=Yfm16Tlpcwk&ab_channel=AluraLatam))
- **YouTube** Git y GitHub: Herramientas Esenciales para el Control de Versiones y Colaboración en Desarrollo (<https://www.youtube.com/watch?v=dw04N616Abw>)
- **YouTube** ¿Puedo subir mi proyecto a Github sin líneas de comando? - Git y Github para principiantes (<https://www.youtube.com/watch?v=Yfm16Tlpcwk>)
- **Curso** Git y GitHub: repositorio, commit y versiones (<https://app.aluracursos.com/course/git-github-repositorio-commit-versiones>)
- **Curso** Git y Github: estrategias de ramificación, conflictos y Pull Requests (<https://www.aluracursos.com/curso-online-git-github-estrategias-ramificacion-conflictos-pull-requests>)

## HTTP - Fundamentos:

- HTTP significa Protocolo de transferencia de hipertexto. La comunicación entre las computadoras cliente y los servidores web se realiza mediante el envío de solicitudes HTTP y la recepción de respuestas HTTP.
- Comprender la diferencia entre los verbos HTTP
- Probar solicitudes y verificar los códigos de estado en el navegador
- Aprendiendo a hacer una solicitud HTTP en la línea de comando con WGET
- Descargar una imagen con WGET
- Realización de una POST

## Contenidos

- **Web** W3Schools: ¿Qué es HTTP? ([https://www.w3schools.com/whatis/whatis\\_http.asp](https://www.w3schools.com/whatis/whatis_http.asp))
- **Web** MDN Web Docs: Una descripción general de HTTP (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>)
- **Web** MDN Web Docs: Métodos de solicitud HTTP (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>)
- **Web** MDN Web Docs: HTTP (<https://developer.mozilla.org/es/docs/Web/HTTP>)
- **Web** MDN Web Docs: Métodos de petición HTTP (<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>)
- **Web** MDN Web Docs: Mensajes HTTP (<https://developer.mozilla.org/es/docs/Web/HTTP/Messages>)
- **Web** HTTP Cats (<https://http.cat/>)

- **Web** HTTP Dogs (<https://http.dog/>)
- **YouTube** Peticiones, Métodos Http y Códigos de estado. (<https://www.youtube.com/watch?v=gBK-Mfa0lw8>)
- **YouTube** SSL, TLS, HTTPS, HTTP - Explicado Fácilmente ([https://www.youtube.com/watch?v=6HJAWFenYx8&ab\\_channel=ProfeSang](https://www.youtube.com/watch?v=6HJAWFenYx8&ab_channel=ProfeSang))
- **YouTube** ★ Protocolo HTTP 🖥 Requests y Responses con: GET, POST, PUT, PATCH y DELETE | Desarrollo web 🌐 (<https://www.youtube.com/watch?v=I2MihYAj0lw>)
- **YouTube** REST y los Verbos de HTTP (<https://www.youtube.com/watch?v=OHBHeAPoZ8E>)

### Contenidos Alura:

- **Artículo** HTTP: Desmitificando el protocolo Web (<https://www.aluracursos.com/blog/http-desmitificando-el-protocolo>)
- **Artículo** ¿Cual es la diferencia entre HTTP y HTTPS? (<https://www.aluracursos.com/blog/cual-es-la-diferencia-entre-http-y-https>)
- **Artículo** HTTP: Diferencias entre GET y POST (<https://www.aluracursos.com/blog/diferencias-entre-get-y-post>)
- **Artículo** Métodos de petición HTTP (<https://www.aluracursos.com/blog/metodos-de-peticion-http>)
- **Curso** HTTP: La base de internet (<https://app.aluracursos.com/course/http-base-internet>)

### JSON:

- JSON significa Notación de objetos de JavaScript. Es un formato de texto para almacenar y transportar datos.
- Crear un objeto
- Transformar un objeto en una cadena
- Transformar una cadena en un objeto
- Manipular un objeto

### Contenidos

- **Web** MDN Web Docs: JSON ([https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global\\_Objects/JSON](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/JSON))
- **Web** MDN Web Docs: Trabajando con JSON (<https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>)
- **YouTube** ¿Que és JSON y cómo funciona? ([https://www.youtube.com/watch?v=z8qk7T\\_2sWg&ab\\_channel=SoyDalto](https://www.youtube.com/watch?v=z8qk7T_2sWg&ab_channel=SoyDalto))

- **YouTube** ★ ¿Qué es JSON? ¿Cuál es su SINTAXIS? 🖥️ ¿Cómo crear un archivo JSON? | DESARROLLO WEB 🌐 ([https://www.youtube.com/watch?v=RhxOTqFbl5Q&ab\\_channel=TodoCode](https://www.youtube.com/watch?v=RhxOTqFbl5Q&ab_channel=TodoCode))

### Contenidos Alura:

- **Artículo** ¿Que es Json? (<https://www.aluracursos.com/blog/que-es-json>)
- **Artículo** ¿JSON y Objeto JavaScript son lo mismo? (<https://www.aluracursos.com/blog/json-y-objeto-javascript-son-lo-mismo>)
- **Artículo** Simulando una API REST con json-server (<https://www.aluracursos.com/blog/simulando-una-api-rest-con-json-server>)
- **Artículo** ¿Qué es JSON Web Token? (<https://www.aluracursos.com/blog/que-es-json-web-token>)
- **Curso** JS en la Web: CRUD con JavaScript asíncrono (<https://www.aluracursos.com/curso-online-js-web-crud-javascript-asincrono>)

### Línea de Comando - Fundamentos:

- CLI es un programa de línea de comandos que acepta la entrada de texto para ejecutar funciones del sistema operativo.
- Conocer los comandos más importantes

### Contenidos

- **Web** W3Schools: What is Command Line Interface (CLI)? ([https://www.w3schools.com/whatis/whatis\\_cli.asp](https://www.w3schools.com/whatis/whatis_cli.asp))
- **Web** Uso de argumentos de la línea de comandos para Terminal Windows (<https://learn.microsoft.com/es-es/windows/terminal/command-line-arguments?tabs=windows>)
- **Artículo** Interfaz de línea de comandos o CLI (<https://www.computerweekly.com/es/definicion/Interfaz-de-linea-de-comandos-o-CLI>)
- **Artículo** El Manual de Comandos de Linux (<https://www.freecodecamp.org/espanol/news/comandos-de-linux/>)
- **YouTube** Consola vs Terminal vs Shell vs CLI 🖥️ ¿Qué es la terminal? ([https://www.youtube.com/watch?v=1YxHXBsVNGQ&ab\\_channel=ProgramadorX](https://www.youtube.com/watch?v=1YxHXBsVNGQ&ab_channel=ProgramadorX))
- **YouTube** Aprende la linea de comandos en un mac - bash scripting ([https://www.youtube.com/watch?v=vfwA3pUnVOg&ab\\_channel=Datademia](https://www.youtube.com/watch?v=vfwA3pUnVOg&ab_channel=Datademia))
- **YouTube** freeCodeCamp.org: Command Line Crash Course (<https://www.youtube.com/watch?v=yz7nYlnXLfE>)

- **YouTube** Traversy Media: Command Line Crash Course For Beginners - Terminal Commands (<https://www.youtube.com/watch?v=uwAqEzhyjtw>)
- **YouTube** Comandos Básicos e Intermedios CMD (<https://youtu.be/erKosEQaaFc>)
- **YouTube** Terminal MAC tutorial en Español - Cómo usar la terminal en MAC (<https://www.youtube.com/watch?v=TmP3y7Z2kk4>)

### Contenidos Alura:

- **Artículo** CMD: Sugerencias para trabajar en el prompt de Windows (<https://www.aluracursos.com/blog/consejos-para-trabajar-en-el-indicador-de-windows>)
- **Artículo** Como usar el terminal integrado de Visual Studio Code (<https://www.aluracursos.com/blog/como-usar-el-terminal-integrado-de-visual-studio-code>)
- **Curso** Linux 1: conociendo y utilizando la terminal (<https://app.aluracursos.com/course/linux-1-conociendo-utilizando-terminal>)

### Cloud - Fundamentos:

- La computación en nube, o cloud computing, es la distribución de servicios informáticos a través de Internet mediante un modelo de tarificación de pago por uso. Una nube se compone de varios recursos informatizados, desde los propios ordenadores (o instancias, en terminología de nube) hasta las redes, el almacenamiento, las bases de datos y todo lo que les rodea. En otras palabras, todo lo que normalmente se necesita para montar el equivalente a una sala de servidores, o incluso un centro de datos completo, estará listo para usar, configurar y ejecutar.
- Conocer la diferencia entre IaaS, PaaS y SaaS
- Conocer los mayores proveedores de nube
- Especializarse en un proveedor específico de su preferencia

### Contenidos

- **Web** ¿Qué es la informática en la nube? | Microsoft Azure (<https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-cloud-computing/>)
- **Web** Amazon AWS: ¿Qué es la computación en nube? (<https://aws.amazon.com/en/what-is-cloud-computing/>)
- **Web** Tipos de computación en la nube (<https://aws.amazon.com/es/types-of-cloud-computing/>)
- **Web** ¿Cómo funciona Azure? (<https://learn.microsoft.com/es-es/azure/cloud-adoption-framework/get-started/what-is-azure>)

- **Web** ¿Qué es el almacenamiento en la nube? (<https://aws.amazon.com/es/what-is/cloud-storage/>)
- **Web** ¿Qué es la seguridad en la nube? (<https://cloud.google.com/learn/what-is-cloud-security?hl=es-419>)
- **Web** Arquitectura sin servidor (<https://learn.microsoft.com/es-es/dotnet/architecture/serverless/serverless-architecture>)
- **Web** ¿Qué es Docker? (<https://aws.amazon.com/es/docker/>)
- **Artículo** Guía para principiantes sobre los fundamentos de la computación en nube (<https://scientya.com/a-beginners-guide-to-the-basics-of-what-cloud-computing-is-about-e8b3b7f25a30/>)
- **Artículo** Cloud Computing para principiantes (<https://medium.com/hackernoon/cloud-computing-for-beginners-85d168959afb/>)
- **Artículo** ¿Qué es Google Cloud y para qué sirve? (<https://www.incentro.com/es-ES/blog/que-es-google-cloud-platform>)
- **YouTube** ¿Qué es computación en la nube? | ¿Qué es cloud computing? | Explicado en 4 minutos (<https://youtu.be/MCKdahh2ISo>)
- **YouTube** 📌 CLOUD COMPUTING ¿Qué es IaaS, PaaS y SaaS? | Modelos de Servicio Cloud (<https://youtu.be/VR8aXePkQ5M>)
- **YouTube** ¿Qué es AWS? (<https://www.youtube.com/watch?v=x2vrg7HuM6g>)
- **YouTube** ¿Cómo empiezo con Google Cloud? (Hablemos en Cloud) (<https://www.youtube.com/watch?v=OiDWqu0oQfo>)
- **YouTube** Introducción a la infraestructura de Google Cloud (<https://www.youtube.com/watch?v=209DGQCism4>)
- **YouTube** ¿Qué es la Computación en la Nube? | AWS desde cero - Parte 1: Introducción (<https://www.youtube.com/watch?v=IciVhWQ8npw>)

## Contenidos Alura:

- **Artículo** ¿Qué es Cloud y sus principales servicios? (<https://www.aluracursos.com/blog/que-es-cloud-y-sus-principales-servicios>)
- **Artículo** Conociendo Terraform (<https://www.aluracursos.com/blog/conociendo-terraform>)
- **Artículo** Empezando con Docker (<https://www.aluracursos.com/blog/empezando-con-docker>)
- **Artículo** Heroku, Vercel y otras opciones de cloud como plataforma (<https://www.aluracursos.com/blog/heroku-vercel-y-otras-opciones-de-cloud-como-plataforma>)

- **YouTube** Fundamentos del OCI | Contenidos ONE (<https://youtu.be/rEgSc0UqX-g>)
- **Curso** Curso Oracle Cloud Infrastructure: implementación de una aplicación en la nube (<https://app.aluracursos.com/course/oracle-cloud-infrastructure-implementacion-aplicacion-nube>)
- **Curso** Curso Oracle Cloud Infrastructure: base de datos e infraestructura como código (<https://app.aluracursos.com/course/oracle-cloud-infrastructure-base-datos-infraestructura-codigo>)
- **Curso** Curso Deploy en Amazon EC2: Alta disponibilidad y escalabilidad de una aplicación (<https://app.aluracursos.com/course/deploy-amazon-ec2-alta-disponibilidad-escalabilidad>)
- **Curso** Curso Amazon Lightsail: Simplificando la nube (<https://app.aluracursos.com/course/amazon-lightsail-simplificando-nube>)

## SQL - Fundamentos:

- Conocer los comandos más comunes de SQL
- Usar SELECT para consultar una tabla
- Usar INSERT para insertar datos en una tabla
- Usar UPDATE para actualizar una tabla
- Usar DELETE para eliminar datos de una tabla
- Usar JOIN para conectar los datos de múltiples tablas
- Conocer las cláusulas (FROM, ORDER BY, etc.)

## Contenidos

- **Artículo** Amazon: ¿Qué es SQL? ([https://aws.amazon.com/es/what-is/sql/?nc1=h\\_ls](https://aws.amazon.com/es/what-is/sql/?nc1=h_ls))

## Contenidos Alura:

- **Artículo** MySQL: desde la descarga e instalación hasta su primera tabla (<https://www.aluracursos.com/blog/mysql-desde-la-descarga-e-instalacion-hasta-su-primera-tabla>)
- **Artículo** Bases de datos relacionales (<https://www.aluracursos.com/blog/base-de-datos-relacional>)
- **Artículo** ¿Qué es SQL? (<https://www.aluracursos.com/blog/que-es-sql>)
- **Artículo** Normalización en base de datos - Estructura (<https://www.aluracursos.com/blog/normalizacion-en-base-de-datos>)
- **Artículo** En SQL, null es null, vacío está vacío (<https://www.aluracursos.com/blog/en-sql-null-es-null-vacio-es-vacio>)



- **Artículo** SELECT, INSERT, UPDATE y DELETE en SQL: aprende a utilizar cada uno (<https://www.aluracursos.com/blog/select-insert-update-delete-sql>)
- **Artículo** Funciones de agregación con GROUP BY en SQL, ¿cómo utilizarlas? (<https://www.aluracursos.com/blog/funciones-de-agregacion-con-group-by-en-sql-como-utilizarlas>)
- **Artículo** SQL JOIN: Aprenda INNER, LEFT, RIGHT, FULL e CROSS (<https://www.aluracursos.com/blog/sql-join-aprenda-inner-left-right-full-e-cross>)
- **Artículo** select count(\*), count(1) y count(nombre): batalla de los counts de SQL (<https://www.aluracursos.com/blog/select-count-count1-e-countnome-la-batalla-de-los-counts-de-sql>)
- **YouTube** Descomplicando Base de Datos | #Aluramás ([https://www.youtube.com/watch?v=G1cDRqKuxpg&t=6s&ab\\_channel=AluraLatam](https://www.youtube.com/watch?v=G1cDRqKuxpg&t=6s&ab_channel=AluraLatam))
- **YouTube** ¿Qué es SQL y NoSQL? (<https://www.youtube.com/watch?v=cLLKVd5CNLc>)
- **YouTube** Banco de Datos MySQL (<https://www.youtube.com/watch?v=8J0AoPZMVxA>)
- **Curso** SQL con MySQL (<https://app.aluracursos.com/formacion-sql-con-mysql>)
- **Curso** Curso Introducción a SQL con MySQL: Manipule y consulte datos (<https://app.aluracursos.com/course/introduccion-sql-mysql-manipule-consulte-datos>)
- **Curso** SQL Server: consultas avanzadas con Microsoft SQL Server 2019 (<https://app.aluracursos.com/course/sql-server-consultas-microsoft-sql-server-2019>)
- **Curso** Formación Modelado de datos (<https://www.aluracursos.com/formacion-modelado-de-datos>)
- **Curso** Formación SQL con Microsoft SQL Server (<https://www.aluracursos.com/formacion-sql-con-microsoft-sql-server-2019>)

## Habilidad Auxiliar: Buenas prácticas

### SOLID:

- Solid tiene cinco principios considerados como buenas prácticas en el desarrollo de software que ayudan a los programadores a escribir los códigos más limpios, dividiendo las responsabilidades, disminuyendo los acoplamientos, facilitando la refactorización y estimulando el reaprovechamiento del código. Propuesto por Robert C. Martin, SOLID propicia el desarrollo de un código limpio, legible y comprobable.

### Contenidos

- **Web** Los principios SOLID de programación orientada a objetos (<https://www.freecodecamp.org/espanol/news/los-principios-solid-explicados-en-espanol/>)



- **Web** Principios SOLID: Qué son, cuáles, y qué beneficios aporta usarlos (<https://devexperto.com/principios-solid/>)
- **Podcast** SOLID - los androides (<https://open.spotify.com/episode/47eElzhTPr1RluVcxeVal4?si=4df70e899c034283>)
- **YouTube** Los principios SOLID, ¡explicados! (<https://www.youtube.com/watch?v=2X50sKeBAcQ>)

## Design Patterns:

- En ingeniería de software, un "patrón de diseño" (Design Pattern en inglés) es una solución general y reutilizable para un problema que ocurre normalmente dentro de un determinado contexto de diseño de software.
- Conocer y aplicar los principales patrones de diseño.

## Contenidos

- **Web** Patrones de Diseño (<https://refactoring.guru/es/design-patterns>)
- **Web** Designer Patterns (<https://github.com/FernandoCalmet/design-patterns>)
- **Web** ¿Qué son los patrones de diseño de software? (<https://profile.es/blog/patrones-de-diseno-de-software/>)
- **Web** Design Patterns: conoce los diferentes tipos que existen y sus beneficios (<https://www.hostgator.mx/blog/design-patterns-que-debes-saber/>)
- **Podcast** Patrones Diseño, ventajas y desventajas (<https://open.spotify.com/episode/3VjQHnPVusU6zz5PyIMVFu?si=a613cb4c157e4055>)
- **Podcast** Patrones Diseño (<https://open.spotify.com/episode/6QO1HYdAgzrMGLVpz2kn0C?si=c95296f387c4490a>)
- **YouTube** ♦ Patrones de diseño software: Repaso completo en 10 minutos (<https://www.youtube.com/watch?v=6BHOeDL8vls>)

## Contenidos Alura:

- **Artículo** Design Patterns: Breve introducción a los patrones de diseño (<https://www.aluracursos.com/blog/design-patterns-introduccion-a-los-patrones-de-diseno>)

## Clean Architecture:

- Clean Architecture (Arquitectura Limpia) es una forma de desarrollar software, de tal forma que solo mirando el código fuente de un programa, debes ser capaz de decir lo que el programa hace.

## Contenidos

- **Web** ¿Qué es Clean Architecture? (<https://clean-architecture-python.readthedocs.io/en/latest/introduccion/index.html>)
- **Artículo** ¿Por qué utilizo Clean Architecture? (<https://xurxodev.com/por-que-utilizo-clean-architecture-en-mis-proyectos/>)
- **YouTube** Revisando Clean code, vale la pena leerlo? | review clean code (<https://www.youtube.com/watch?v=uQfm6YaJTJI>)
- **YouTube** Hexagonal architecture, qué es y qué diferencias tiene contra Clean Architecture? - PT 1 (<https://www.youtube.com/watch?v=NOWU4K6piwo>)
- **YouTube** Desarrollo ágil con Arquitectura limpia Hexa3 (<https://medium.com/@dariopalminio/desarrollo-%C3%A1gil-de-ecosistemas-de-aplicaciones-hexagonales-3-capas-hexa3l-d6370bf11db0>)
- **Podcast** DevTalles - 125: Arquitectura Limpia (<https://open.spotify.com/episode/3ftlJfucj7Jx4ZO2cDSghx?si=e6b2c5607ead43b5>)

## Contenidos Alura:

- **Artículo** Design Patterns: Breve introducción a los patrones de diseño (<https://www.aluracursos.com/blog/design-patterns-introduccion-a-los-patrones-de-diseno>)

## Clean Code:

- Aplicar técnicas sencillas para facilitar la escritura y la lectura de un código.
- Refactorizar el código para que quede más limpio.

## Contenidos

- **Web** Clean Code: código limpio, ¿qué es? (<https://www.hostgator.mx/blog/clean-code-codigo-limpio/>)
- **Web** Clean Code: 35 claves para dominarlo (<https://devexperto.com/clean-code/>)
- **Podcast** ¿Que es Clean Code? - Programando Podcast (<https://open.spotify.com/episode/0xF40axFOYG1pdD2oaEMo0?si=b48fbfed5ec949c4>)
- **Artículo** Cómo escribir código limpio. Nombrando elementos (<https://medium.com/@latteandcode/como-escribir-codigo-limpio-nombrando-elementos-8caa6639602c>)
- **YouTube** Aprende a programar más limpio | The Clean Coder | Reseña del libro de Robert C. Martin (<https://www.youtube.com/watch?v=bZgWdELGBoc>)

## Contenidos Alura:

- **Artículo** Clean Code: qué es, casos de uso, ejemplo de código limpio (<https://www.aluracursos.com/blog/clean-code-que-es-ejemplo>)

## Diseño Orientado a Dominio - DDD:

- El Diseño Orientado a Dominio (DDD) es un enfoque de diseño y desarrollo de software que se informa principalmente por los requisitos de negocio. Los componentes del programa (objetos, clases, matrices, etc.) indican la industria, sector o dominio empresarial en que opera el negocio.
- Modelar dominios de manera efectiva.
- Basar proyectos complejos en modelos de dominio.
- Conocer los bloques de construcción de DDD.

## Contenidos

- **Web** Domain Driven Design: principios, beneficios y elementos — Primera Parte (<https://medium.com/@jonathanloscalzo/domain-driven-design-principios-beneficios-y-elementos-primera-parte-aad90f30aa35>)
- **Web** Domain Driven Design: principios, beneficios y elementos — Segunda Parte (<https://medium.com/@jonathanloscalzo/domain-driven-design-principios-beneficios-y-elementos-segunda-parte-337d77dc8566>)
- **Web** Introducción a Domain Drive Design - DDD (<https://refactorizando.com/introduccion-domain-drive-design/>)
- **Podcast** TDD y DDD, fortalezas y cuando usarlo (<https://open.spotify.com/episode/7iioBOv7ySQZttPyw2GshP?si=a114f642ecc94682>)
- **YouTube** Aprendiendo el Dominio (DDD en Español) (<https://www.youtube.com/watch?v=BilfTrk954A>)
- **YouTube** Domain Driven Design - Píldoras de conocimiento - Autentia (<https://www.youtube.com/watch?v=vccTIFGSiHA>)