



# C# Basics

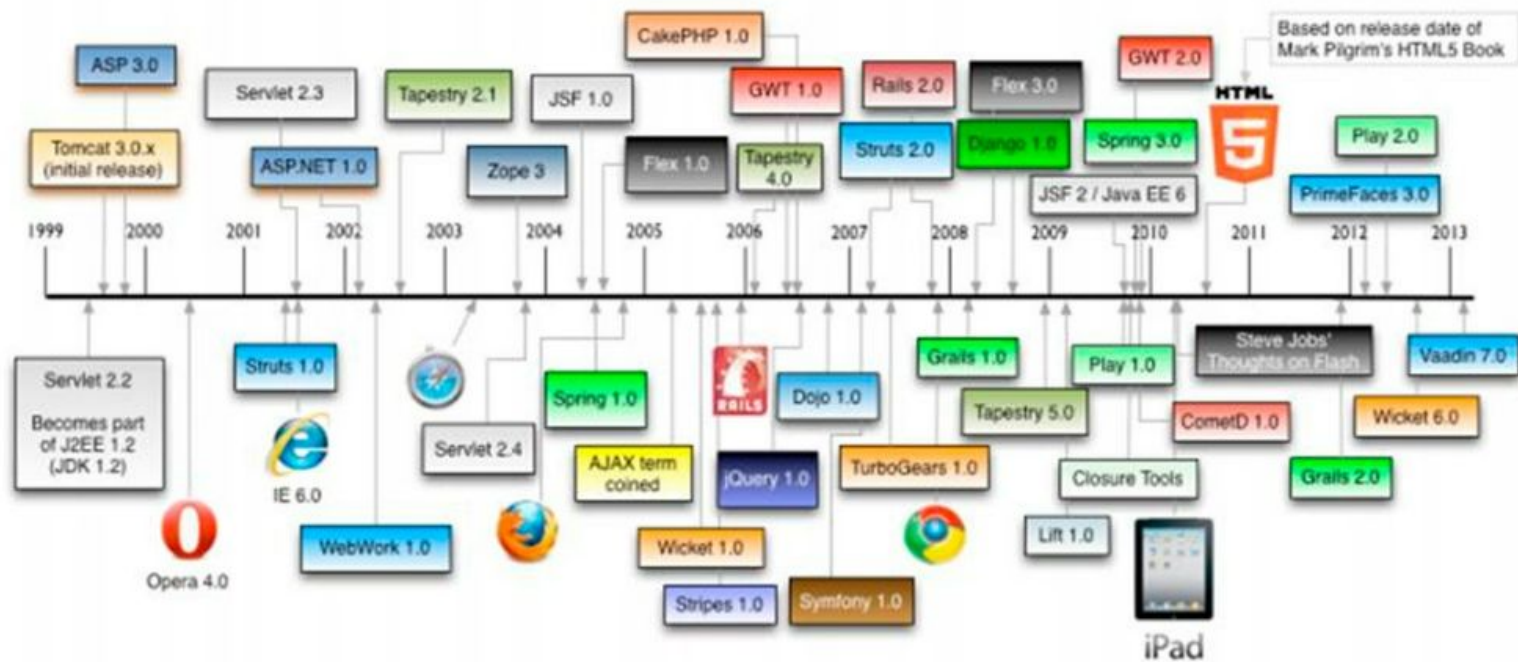
# Conceptos acerca de .NET

**Framework:** Un framework es un **esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos**, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.

COM y ATL: Component Object Model / Active Template Library. Nos permiten crear librerías y compartir código entre lenguajes.

Framework de .NET:

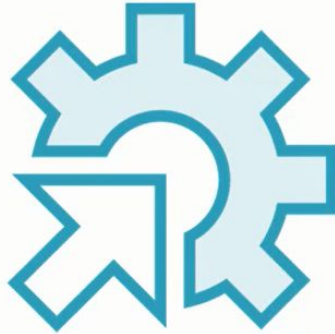
- CLR: Common language runtime
- Assembly: Cuando tenemos un programa escrito en un lenguaje de .NET y lo compilamos se genera el assembly. El assembly contiene el programa compilado en lo que conocemos como CIL y también información sobre todos los tipos que se utilizan en el programa.
- CIL: Common intermediate language.
- JIT: compilación justo a tiempo.
- SDK: software development kit



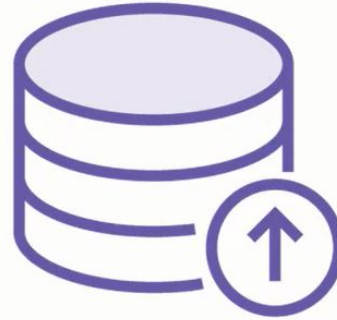
# Construyendo una aplicación



User interface  
layer



Business logic  
layer



Data access  
layer



Common code

## Web y nube (4)



## Desarrollo de ASP.NET y web

Compila aplicaciones web con ASP.NET Core, ASP.NET, HTML/JavaScript y contenedores, e incluye compatibilida...



## Desarrollo de Python

Edición, depuración, desarrollo interactivo y control de código fuente de Python.



## Desarrollo de Azure

Proyectos, herramientas y SDK de Azure para desarrollar aplicaciones en la nube y crear recursos mediante .NET y...



## Desarrollo de Node.js

Compile aplicaciones de red escalables con Node.js, un entorno de ejecución JavaScript controlado por eventos...



## Móviles y de escritorio (5)



## Desarrollo de la interfaz de usuario de aplicaciones mu...

Cree aplicaciones de Android, iOS, Windows y Mac desde un único código base con C# mediante .NET MAUI.



## Desarrollo de escritorio de .NET

Compila WPF, Windows Forms y aplicaciones de consola mediante C#, Visual Basic y F# con .NET y .NET Framework...



## Desarrollo para el escritorio con C++

Cree aplicaciones modernas de C++ para Windows con...



## Desarrollo de aplicaciones de Windows

Cree aplicaciones para la plataforma Windows con WinUI...



## Ubicación

C:\Program Files\Microsoft Visual Studio\2022\Community

Si continúa, indica que acepta la [licencia](#) de la edición de Visual Studio que ha seleccionado. También puede descargar otro software con Visual Studio. Este software tiene una licencia aparte, como se explica en los [avisos de terceros](#) o en la licencia que incluye el software. Si continúa, indica que también acepta esas licencias.

## Detalles de la instalación

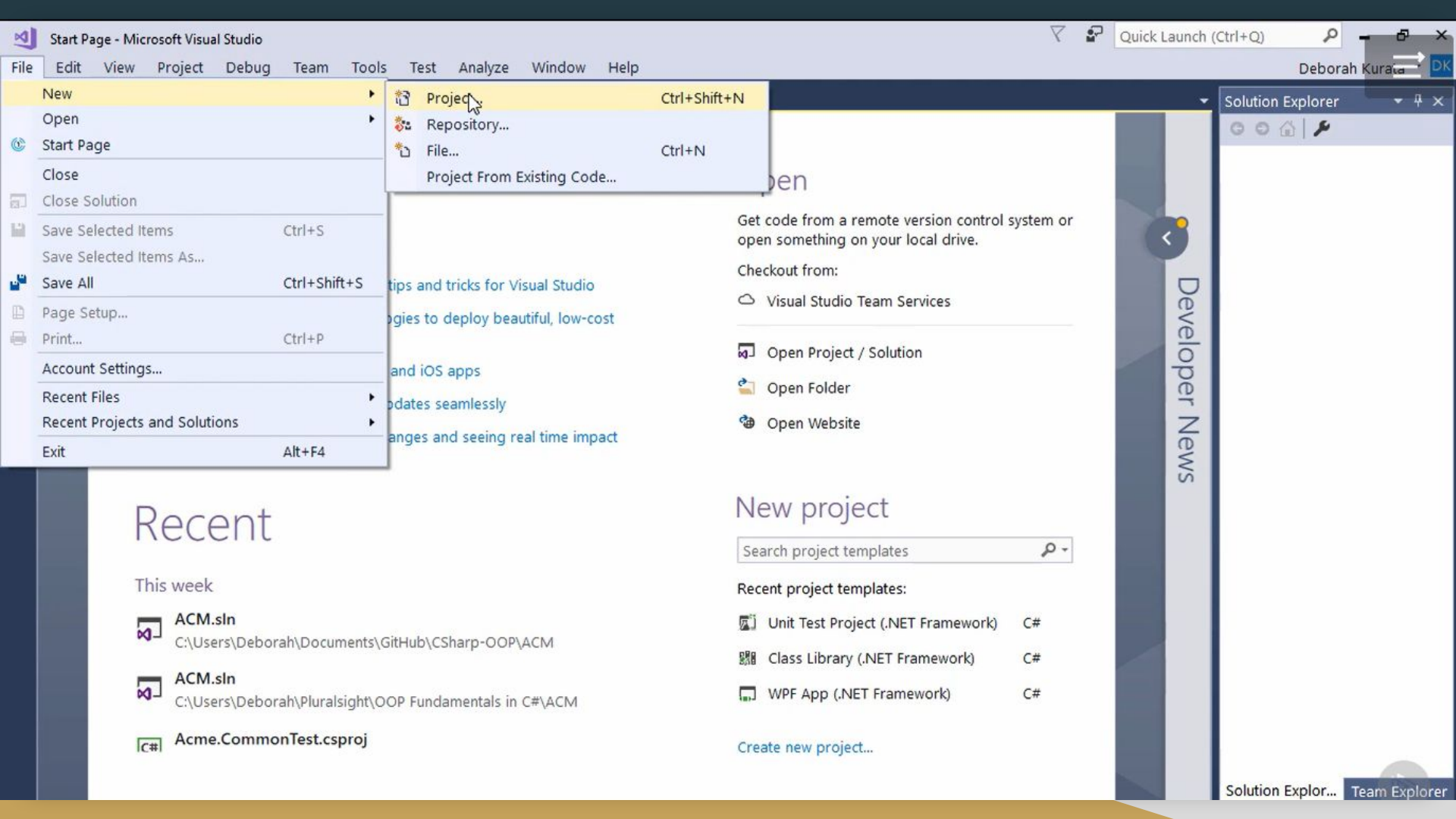
- ☒ Herramientas de Entity Framework 6
- ☒ Herramientas para generación de perfiles...
- ☒ IntelliCode
- ☒ Depurador Just-In-Time
- ☒ Live Share
- ☒ ML.NET Model Builder
- ☒ GitHub Copilot
- ☒ Blend for Visual Studio
- ☐ Compatibilidad con el lenguaje de escritor...
- ☐ PreEmptive Protection - Dotfuscator
- ☒ Herramientas de desarrollo de .NET Frame...
- ☐ Paquete de compatibilidad de bibliotecas...
- ☐ Windows Communication Foundation
- ☒ SQL Server Express 2019 LocalDB
- ☐ MSIX Packaging Tools
- ☒ Diagnóstico de JavaScript
- ☒ Plantillas de C# del SDK de Aplicaciones p...
- ☒ Herramientas de desarrollo de .NET Frame...

Quitar componentes que no son compatibles

Espacio total necesario 0 B

Instalar durante la descarga

Cerrar



- New
  - Open
  - Start Page
  - Close
  - Close Solution
  - Save Selected Items Ctrl+S
  - Save Selected Items As...
  - Save All Ctrl+Shift+S
  - Page Setup...
  - Print... Ctrl+P
  - Account Settings...
  - Recent Files
  - Recent Projects and Solutions
  - Exit Alt+F4
- Project... Ctrl+Shift+N
  - Repository...
  - File... Ctrl+N
  - Project From Existing Code...

## Recent

### This week

- ACM.sln  
C:\Users\Deborah\Documents\GitHub\CSharp-OOP\ACM
- ACM.sln  
C:\Users\Deborah\Pluralsight\OOP Fundamentals in C#\ACM
- Acme.CommonTest.csproj

Get code from a remote version control system or open something on your local drive.

Checkout from:

Visual Studio Team Services

Open Project / Solution

Open Folder

Open Website

## New project

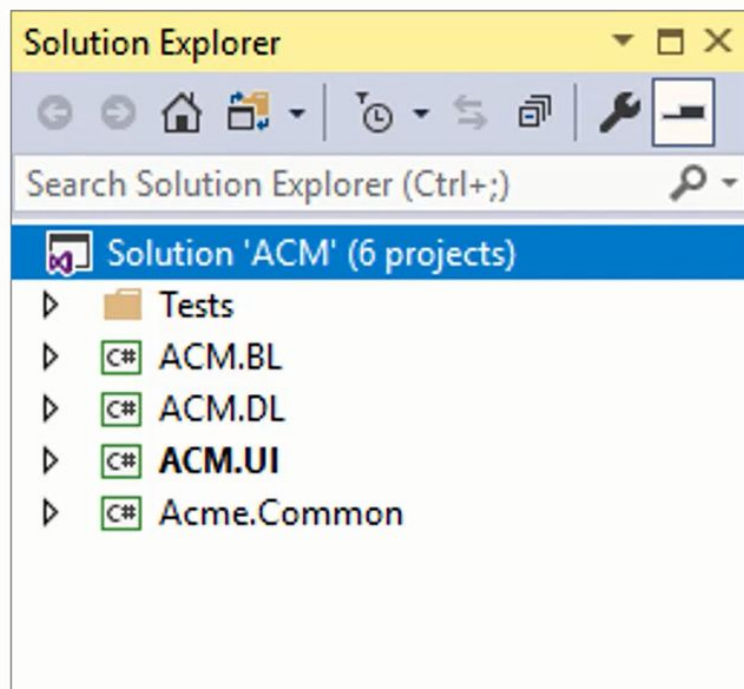
Search project templates

Recent project templates:

- Unit Test Project (.NET Framework) C#
- Class Library (.NET Framework) C#
- WPF App (.NET Framework) C#

Create new project...

# Visual Studio Solution



# DEMO

## Customer

- Name
- Email address
- Home address
- Work address
- Validate()
- Retrieve()
- Save()



# Testing the code

- Unit Testing (Test Unitario)
- Integration Test (Test de Integración)

# Testing Our Code

```
public class Customer
{
    public int CustomerId { get; private set; }

    public string EmailAddress { get; set; }

    public string FirstName { get; set; }

    public string FullName
    {
        get
        {
            return LastName + "," + FirstName;
        }
    }

    private string _lastName;
    public string LastName
    {
        get
        {
            return _lastName;
        }
        set
        {
            _lastName = value;
        }
    }
}
```

Our class

```
[TestClass]
public class CustomerTest
{
    [TestMethod]
    public void FullNameTestValid()
    {
        //-- Arrange
        Customer customer = new Customer();
        customer.FirstName = "Bilbo";
        customer.LastName = "Baggins";

        string expected = "Baggins, Bilbo";

        //-- Act
        string actual = customer.FullName;

        //-- Assert
        Assert.AreEqual(expected, actual);
    }
}
```

Test for our class

# DEMO

## Unit test

# Creando Objetos

```
Customer customer = new Customer();
```

```
var customer = new Customer();
```

# Accediendo a Propiedades

```
var customer = new Customer();
```

```
customer.LastName = "Baggins";  
customer.FirstName = "Bilbo";
```

```
var actual = customer.FullName;
```

```
public class Customer  
{  
    public int CustomerId { get; private set; }  
    public string EmailAddress { get; set; }  
    public string FirstName { get; set; }  
    public string FullName  
    {  
        get  
        {  
            string fullName = LastName;  
            if (!string.IsNullOrEmpty(FirstName))  
            {  
                if (!string.IsNullOrEmpty(fullName))  
                {  
                    fullName += ", ";  
                }  
                fullName += FirstName;  
            }  
            return fullName;  
        }  
    }  
}  
  
private string _lastName;  
public string LastName  
{  
    get  
    {  
        return _lastName;  
    }  
    set  
    {  
        _lastName = value;  
    }  
}
```

# Los objetos son Tipos de Referencia

```
int i1;  
i1 = 42;  
  
int i2 = i1;  
i2 = 2;
```

What is i1?

```
var c1 = new Customer();  
c1.FirstName = "Bilbo";  
  
var c2 = c1;  
c2.FirstName = "Frodo";
```

What is c1.FirstName?

# Modificador static

```
public static int InstanceCount { get; set; }
```

```
Customer.InstanceCount += 1;
```

## Métodos de la clase

```
public Customer Retrieve(int customerId)
{
    // Code that retrieves the defined customer

    return new Customer();
}
```



## Metodos (términos)

- Signature (firma)
- Overloading (sobrecarga)
- Contract (contrato)

# Constructores

- Es un método particular que lleva el mismo nombre de la clase. Es usado para inicializar un objeto.

# Ejercicio

## Product

- Product name
- Description
- Current price
- Validate()
- Retrieve()
- Save()

## Order

- ~~• Customer~~
- Order date
- ~~• Shipping address~~
- ~~• Order items~~
- Validate()
- Retrieve()
- Save()

## Order Item

- Product
- Quantity
- Purchase price
- Validate()
- Retrieve()
- Save()