



Creare app con .NET MAUI

05/03/2025

Chi siamo

- ▶ Giammarco Amadori:
 - sviluppatore senior di OnitSmart
- ▶ Ambra Benvenuti:
 - tirocinante in OnitSmart da giugno 2024
 - frequentante il corso di laurea professionalizzante di Tecnologie dei Sistemi Informatici



Introduzione a .NET MAUI

Cos'è .NET MAUI

- ▶ .NET MAUI (*Multiplatform App UI*) è un framework **open-source** sviluppato da Microsoft che permette di creare applicazioni **multiplatforma** utilizzando lo stesso codice.
- ▶ MAUI offre:
 - un'unica **code base** per tutte le piattaforme
 - **supporto nativo** per ogni sistema operativo, con accesso diretto alle API specifiche.
 - un **sistema avanzato di gestione della UI**, che permette di creare interfacce adattabili e reattive.



MULTIPLATFORM vs NATIVO

I due approcci per lo sviluppo di applicazioni messi a confronto

Linguaggi

Sviluppo nativo

- ▶ Android:
 - Kotlin
- ▶ iOS:
 - Swift

Framework multiplatform

- ▶ Flutter:
 - Dart
- ▶ React Native:
 - JavaScript / TypeScript
- ▶ .NET MAUI:
 - C#
 - XAML



Vantaggi

Sviluppo nativo

- ▶ **prestazioni ottimali**
 - codice scritto appositamente per la piattaforma
- ▶ **accesso completo** alle funzionalità del dispositivo
 - e.g.: fotocamera, sensori, GPS
- ▶ **UX migliore**
 - UI fluida e reattiva, ottimizzata

Framework multiplatform

- ▶ **codice riutilizzabile**
 - minori tempi di sviluppo e costi di manutenzione
- ▶ **manutenzione semplificata**
 - una sola *code base* da gestire, ogni modifica viene applicata a tutte le piattaforme



Svantaggi

Sviluppo nativo

- ▶ **tempi di sviluppo** più lunghi
 - il codice deve essere scritto più volte
- ▶ maggiori **costi di manutenzione**
 - più versioni della stessa app da gestire

Framework multiplatform

- ▶ **limiti tecnici** in presenza di funzionalità avanzate
- ▶ **difficoltà di accesso** a funzionalità hardware o del sistema operativo



Regola generale

- ▶ Lo sviluppo in **nativo** è la scelta migliore se il progetto richiede massime prestazioni e accesso completo alle funzionalità hardware.
- ▶ Al contrario, è meglio scegliere uno sviluppo **multiplatforma** se si vuole risparmiare tempo e l'accesso a funzionalità specifiche del dispositivo non è importante.



Pattern di Programmazione: MVVM

Model – View – ViewModel

Pattern di Programmazione

- ▶ Un pattern di programmazione è una **soluzione generale e riutilizzabile** a un problema comune nello sviluppo software. Non è codice specifico, ma un'idea o un **approccio strutturato** per risolvere determinati problemi di progettazione.
- ▶ Vantaggi dei pattern di programmazione:
 - rendono il codice più **organizzato** e **manutenibile**
 - favoriscono la **riusabilità** e la **scalabilità**
 - aiutano i programmatori a seguire le **best practices**

* **scalabilità**: la capacità di un'applicazione o di un sistema di gestire un aumento del carico di lavoro senza perdere prestazioni o stabilità

* **best practices**: linee guida o strategie consigliate per scrivere codice efficiente, sicuro e manutenibile



MVVM

Il pattern MVVM (*Model-View-ViewModel*) è un'**architettura software** utilizzata per **separare** la logica dell'applicazione dalla sua interfaccia grafica.



Model

rappresenta i dati e la logica di business dell'applicazione



Viewmodel

fa da "ponte" tra Model e View, gestendo dati e logica di presentazione



View

è l'interfaccia grafica che l'utente vede e con cui interagisce (UI)



Ambiente .NET

Breve introduzione ai concetti fondamentali di .NET

Introduzione a .NET

- ▶ .NET è un **framework *cross-platform*** creato da Microsoft, progettato per sviluppare applicazioni.
- ▶ Si basa su un insieme di **librerie** e **strumenti** che permettono agli sviluppatori di scrivere codice garantendo interoperabilità e un'elevata efficienza.
- ▶ **C#** è il linguaggio principale dell'ecosistema.



Componenti di .NET

- ▶ .NET permette lo sviluppo di diversi prodotti grazie alle varie tecnologie che mette a disposizione:



Web front-end

Blazor e .NET Core



Web back-end

.NET Core



Cloud

.NET Aspire



AI e Semantic Kernel

C#, OpenAI e Azure, qDrant



App mobile

.NET MAUI



Gaming

Unity



MAUI nell'ecosistema .NET

MAUI è parte dell'ecosistema .NET e ne condivide molte caratteristiche fondamentali, tra cui:



**Linguaggi di
programmazione
comuni**

utilizzo di C#



**Tooling e Ambiente
di Sviluppo**

integrazione in Visual
Studio e utilizzo di NuGet
per la gestione delle
dipendenze



**Accesso alle API di
.NET**

supporto alla DI e
possibilità di utilizzare le
API standard di .NET



Linguaggi Utilizzati

C# e XAML

C# e XAML

C#

- ▶ è il linguaggio principale di .NET MAUI, usato per scrivere la logica dell'app

```
[RelayCommand]
2 references
public void OnAppearing()
{
    OrdersList.Clear();

    var orders = PreferencesUtilities.GetOrders();
    foreach (var order in orders)
    {
        OrdersList.Add(order);
    }
}
```

XAML

- ▶ (*Extensible Application Markup Language*) è un linguaggio di markup per definire l'interfaccia grafica

```
<Label Text="{Binding Error}"
        IsVisible="{Binding HasError}"
        FontAttributes="Bold"
        TextColor="{AppThemeBinding Default={StaticResource DarkFuchsia}}"
        HorizontalOptions="Center"
        Padding="10"
/>
<Button Text="SALVA"
        FontAttributes="Bold"
        Command="{Binding SaveOrderInPreferencesCommand}"
/>
```



LINQ

Breve introduzione al LINQ e al suo utilizzo

LINQ

- ▶ LINQ (*Language Integrated Query*) è una tecnologia di .NET che permette di scrivere **query** direttamente all'interno del **codice C#** in modo semplice e leggibile.
- ▶ Fornisce un insieme di **metodi ed operatori** che consentono di filtrare, ordinare, trasformare e manipolare dati provenienti da **diverse fonti**, tra cui anche Array e Liste

```
var savedOrders = new List<PizzaOrderModel>();  
var order = savedOrders.Where(x => x.Id == id).First();
```

- ▶ *nell'esempio*: chiedo di selezionare il primo elemento della lista di *PizzaOrderModels* chiamata *savedOrders* che abbia l'*id* richiesto



Navigazione

Breve introduzione al Navigation Stack

Navigation Stack

- ▶ Il **Navigation Stack** è ciò che gestisce la navigazione tra le pagine dell'app.
- ▶ Ha la struttura di una **pila** (*stack*) e segue la logica **LIFO** (*last in, first out*). Questo significa che le pagine vengono aggiunte una sopra all'altra e rimosse dall'alto.
- ▶ Mette a disposizione vari metodi, ma a noi serviranno i due più semplici: *Push* e *Pop*.



Navigazione: *Push* e *Pop*

- ***Push*** aggiunge una nuova pagina in cima alla pila e la visualizza



- ***Pop*** rimuove la pagina in cima alla pila e torna a quella precedente



Pacchetti di Dipendenze

Introduzione ai pacchetti di dipendenze e al loro utilizzo tramite NuGet

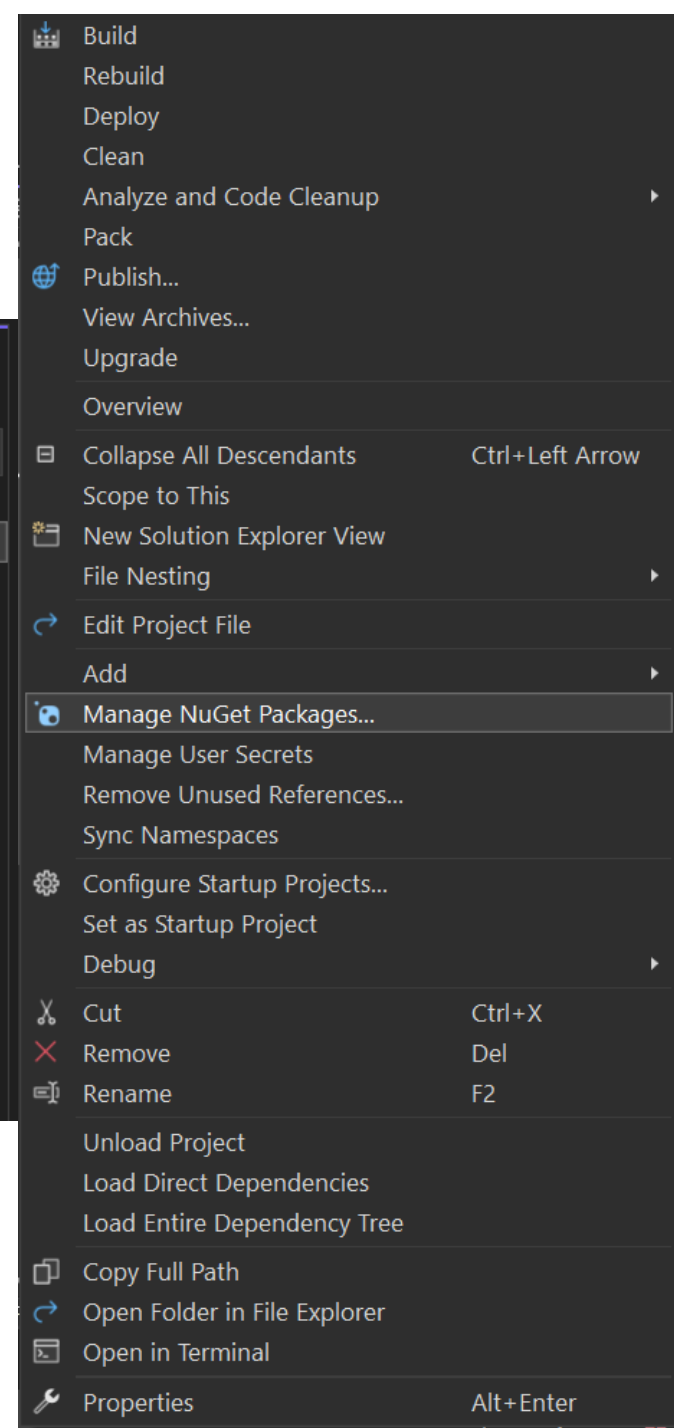
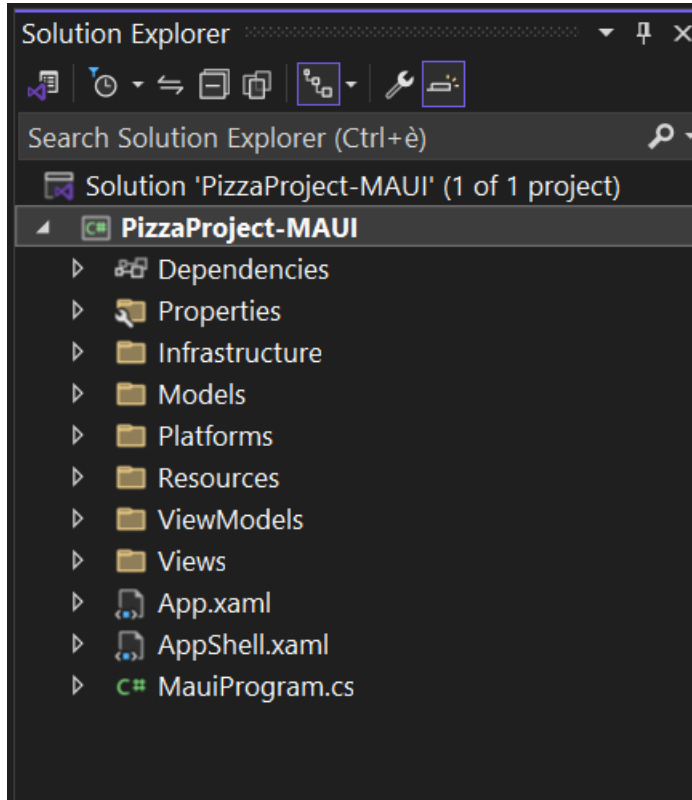
Pacchetti di dipendenze e NuGet

- ▶ I **pacchetti di dipendenze** sono insiemi di codice riutilizzabile (librerie, strumenti, framework) che un'applicazione può utilizzare senza doverli scrivere da zero.
- ▶ **NuGet** è il **gestore di pacchetti ufficiale di .NET**. Consente di scaricare, installare e aggiornare pacchetti di dipendenze direttamente all'interno di un progetto .NET, gestendo automaticamente le versioni e le dipendenze tra i pacchetti.
- ▶ I pacchetti NuGet sono ospitati in un repository online (**nuget.org**) e possono essere aggiunti tramite **Visual Studio** oppure **.NET CLI**.



NuGet Package Manager su Visual Studio

- ▶ Tasto destro del mouse sul nome del progetto (in questo caso *PizzaProject-MAUI*)
- ▶ Selezionare *Manage NuGet Packages* (in italiano *Gestione Pacchetti NuGet*)



NuGet Package Manager su Visual Studio

- Ricorda **sempre** di controllare che la **versione** del pacchetto che si sta installando o aggiornando sia **compatibile** con le tecnologie in utilizzo!
- Nell'esempio: è proposto un aggiornamento per il pacchetto *CommunityToolkit* ma nelle dipendenze è richiesto .NET 9. Poiché il progetto utilizza .NET 8 l'aggiornamento **NON è compatibile** e non si può fare.

CommunityToolkit.Maui nuget.org

Installed: 9.1.1 [Uninstall](#)

Version: Latest stable 11.1.0 [Update](#)

[Package source mapping is off. Configure](#)

[Options](#)

Description

The .NET MAUI Community Toolkit is a collection of Animations, Behaviors, Converters, and Custom Views for development with .NET MAUI. It simplifies and demonstrates common developer tasks building iOS, Android, macOS and Windows apps with .NET MAUI.

Dependencies

- net9.0
 - CommunityToolkit.Maui.Core (>= 11.1.0)
 - Microsoft.Maui.Controls (>= 9.0.30 && < 10.0.0)
- net9.0-android35.0
 - CommunityToolkit.Maui.Core (>= 11.1.0)
 - Microsoft.Maui.Controls (>= 9.0.30 && < 10.0.0)
- net9.0-ios18.0
 - CommunityToolkit.Maui.Core (>= 11.1.0)
 - Microsoft.Maui.Controls (>= 9.0.30 && < 10.0.0)
- net9.0-maccatalyst18.0
 - CommunityToolkit.Maui.Core (>= 11.1.0)
 - Microsoft.Maui.Controls (>= 9.0.30 && < 10.0.0)
- net9.0-tizen10.0
 - CommunityToolkit.Maui.Core (>= 11.1.0)
 - Microsoft.Maui.Controls (>= 9.0.30 && < 10.0.0)
- net9.0-windows10.0.19041
 - CommunityToolkit.Maui.Core (>= 11.1.0)
 - Microsoft.Maui.Controls (>= 9.0.30 && < 10.0.0)



Git e *repository*

Accenni a Git e clonazione di un *repository* su Visual Studio

Cos'è Git

- ▶ **Git** è un **sistema di controllo di versione distribuito (DVCS)** che permette di gestire e tracciare le modifiche ai file di un progetto nel tempo.
- ▶ È particolarmente utile nello sviluppo software, dove più sviluppatori lavorano sugli stessi file **contemporaneamente**.
- ▶ L'archivio del progetto, dove vengono salvati tutti i file e la cronologia delle modifiche, è chiamato **repository** (o **repo**).
- ▶ È buona pratica controllare sempre il file **README.md**, che spiega il contenuto del *repo*.



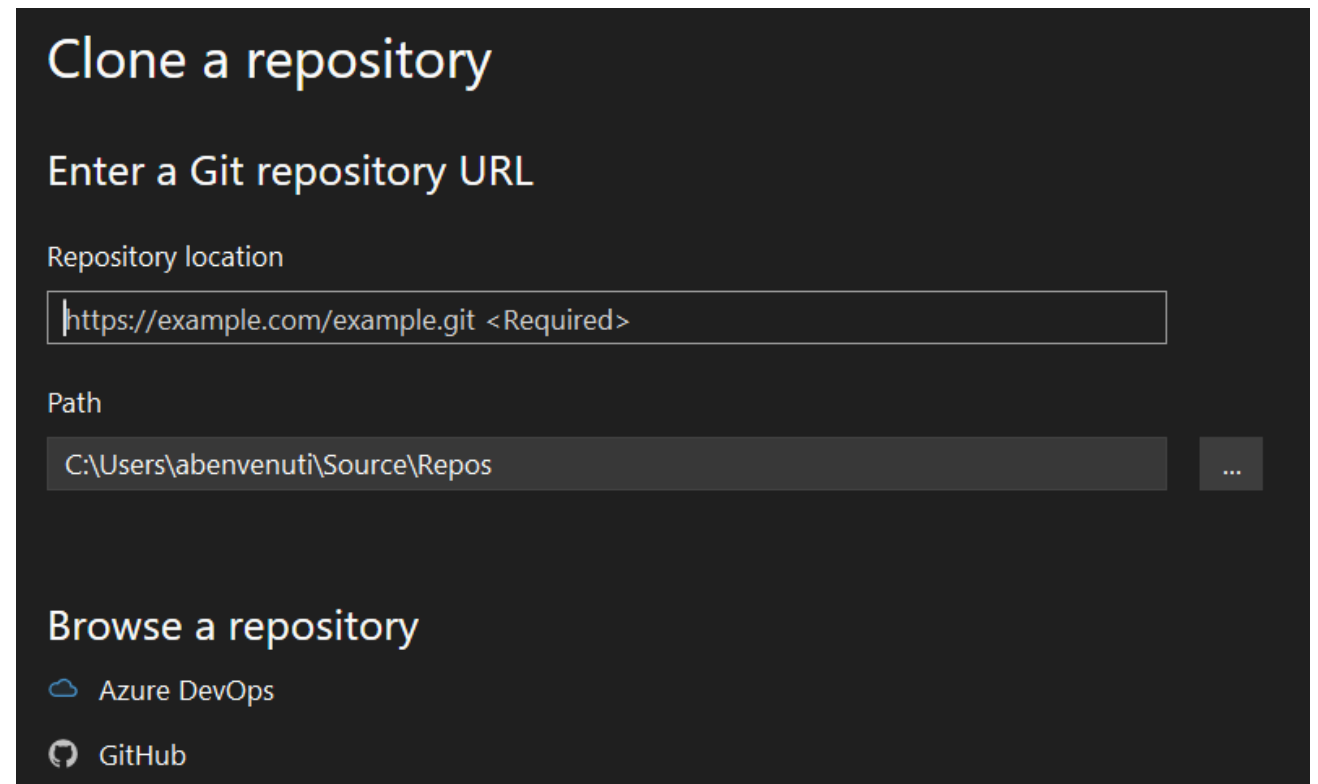
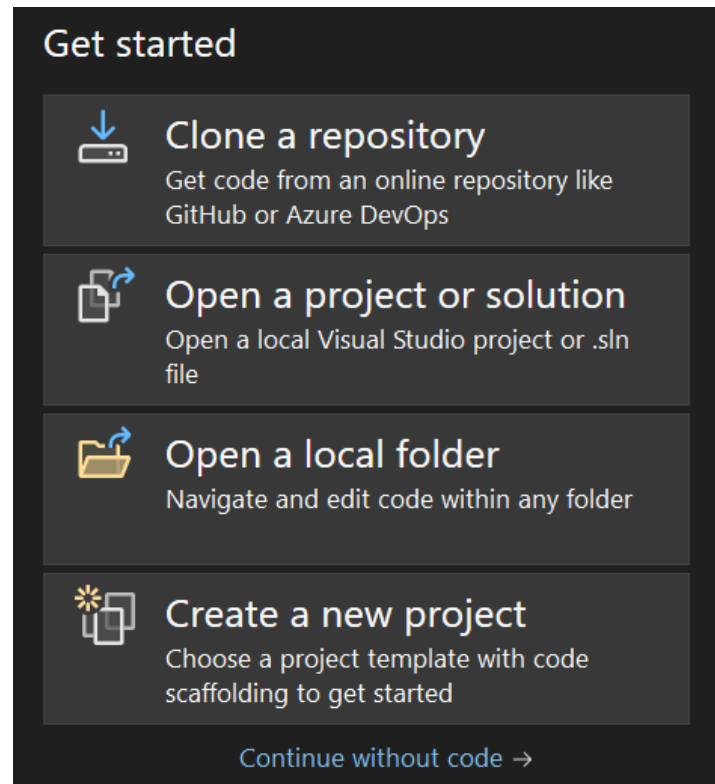
I vantaggi di Git

- ▶ **Traccia le modifiche:** registra ogni modifica ai file, permettendo di tornare a versioni precedenti se necessario.
- ▶ Favorisce il **lavoro di squadra**: più persone possono collaborare sullo stesso progetto senza sovrascrivere il lavoro degli altri.
- ▶ È **distribuito**: ogni sviluppatore ha una copia completa del progetto sul proprio computer, quindi può lavorare anche senza connessione a internet.
- ▶ È **sicuro e veloce**: le operazioni sono rapide e i dati sono protetti da corruzioni accidentali.



Collegare Visual Studio ad un *repo*

- ▶ Per prima cosa aprire Visual Studio e selezionare l'opzione *Clonare un repository*, poi indicare il percorso di una cartella vuota e inserire il link



Documentazione .NET

Accenni alla documentazione di .NET e come usarla

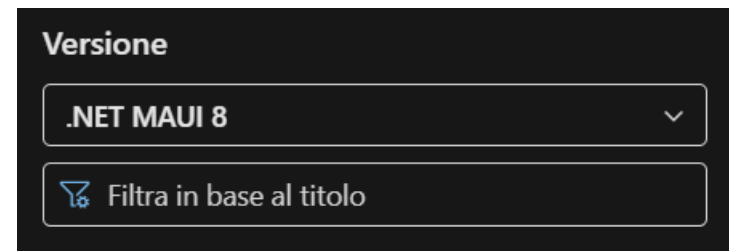
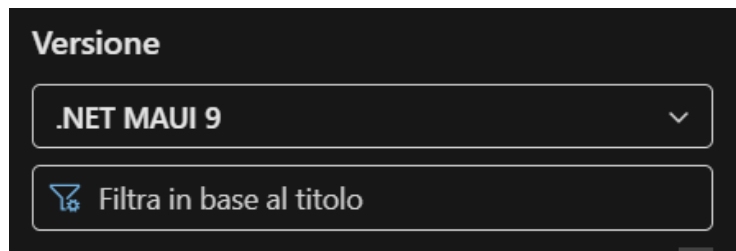
Cos'è la documentazione .NET

- ▶ La documentazione ufficiale di .NET è una raccolta di **guide**, **esempi** e **riferimenti** su tutto l'ecosistema e funge da punto di riferimento per capire come funzionano le **librerie**, i **metodi** e le **funzionalità** di .NET.
- ▶ È **aggiornata costantemente** da Microsoft e dalla community.
- ▶ documentazione generale: docs.microsoft.com/dotnet
- ▶ documentazione MAUI: [.Net MAUI 8](#)



Versione della documentazione .NET

- ▶ Attenzione alla **versione** che si consulta: dobbiamo stare attenti a consultare la documentazione della versione di .NET che stiamo utilizzando
- ▶ Aprendo una pagina di documentazione dai risultati di Google (il caso più solito), questa verrà aperta nell'ultima versione disponibile. Al momento corrisponde a .NET 9, ma per la nostra app useremo **.NET 8**, quindi è necessario ricordarsi di scalare la versione (in alto a sinistra).



Primi step

Guida per creare la prima applicazione con .NET MAUI e pacchetti di dipendenze necessari

Creare l'app

- ▶ Per impostare il progetto in maniera corretta seguiamo la guida della documentazione ufficiale.
 - Link: [Creare la mia prima app](#)
- ▶ **ATTENZIONE:** nel **punto 4** chiederà di selezionare la versione di .NET da utilizzare, non seguite la documentazione ma **scegliete .NET 8 !**



MauiProgram.cs

- ▶ Il file **MauiProgram.cs** è il **punto di ingresso** dell'applicazione, ovvero dove viene costruita e ne vengono configurati vari aspetti, come la gestione delle risorse e l'inizializzazione di componenti esterni.
- ▶ L'applicazione viene creata nel metodo ***CreateMauiApp()*** tramite un *builder* di tipo **MauiAppBuilder**. Il *builder* mette a disposizione una serie di metodi che configurano vari aspetti dell'app.
- ▶ Alla fine della configurazione viene chiamato il metodo ***builder.Build()*** che crea un oggetto di tipo **MauiApp**.



MauiProgram.cs

- ▶ Come prima cosa, nel *builder* viene registrata la classe **App** (definita nel file `App.xaml.cs`) come base dell'applicazione.
- ▶ Successivamente, vengono inizializzati i **pacchetti di dipendenze** che lo necessitano.
- ▶ Infine, sono registrati i **font custom** a partire dai rispettivi file **.ttf** (*TrueType Font*: formato di file utilizzato per archiviare i font).
- ▶ **Attenzione**: ogni metodo del *builder* ritorna il *builder* stesso, di conseguenza i metodi sono da chiamare consecutivamente e non uno alla volta.

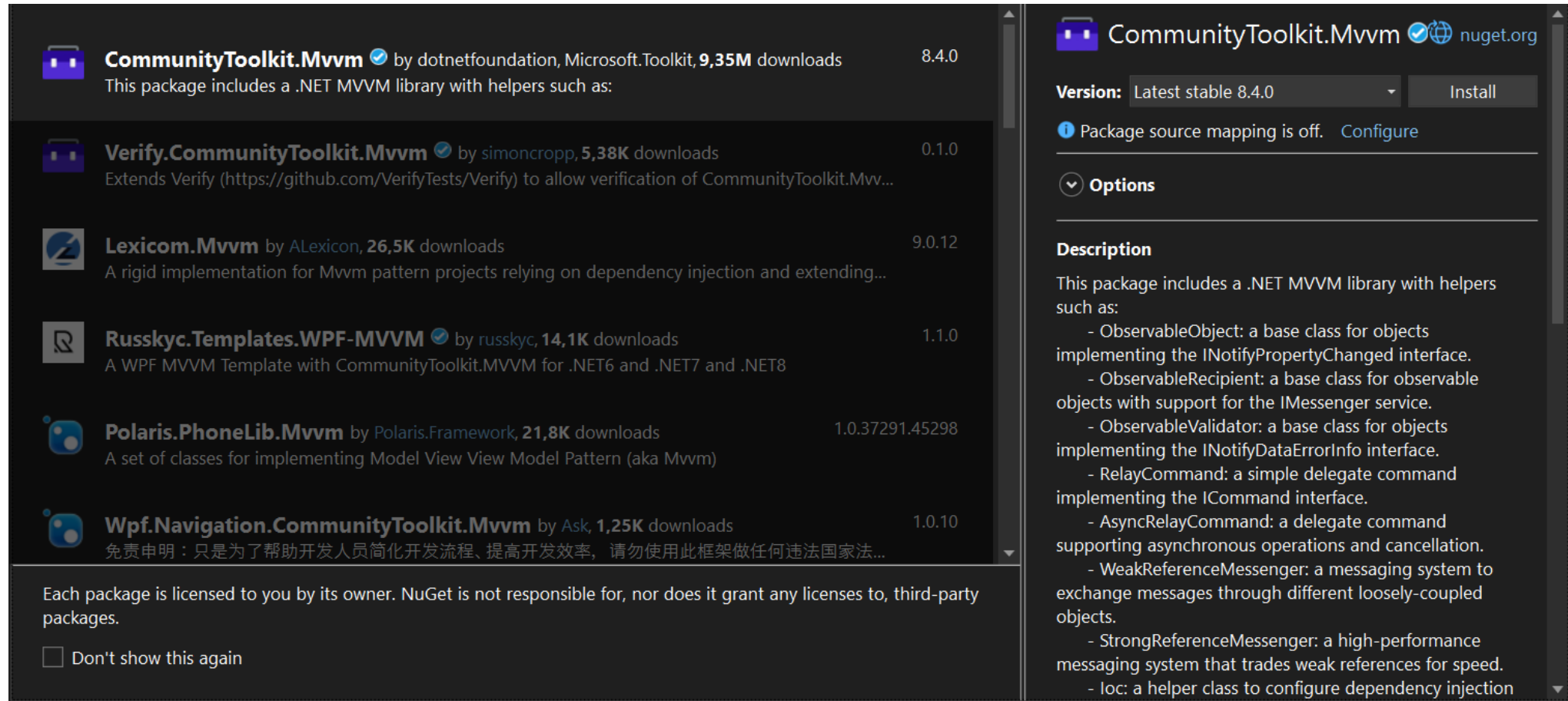


Pacchetti da aggiungere

- ▶ Per il nostro progetto sarà necessario installare alcuni pacchetti di dipendenze:
 - **CommunityToolkit.Mvvm** che mette a disposizione l'oggetto *ObservableCollection* e gli attributi *RelayCommand* e *ObservableProperty*, ciascuno dei quali sarà fondamentale nello sviluppo
 - **CommunityToolkit.Maui** per personalizzare il metodo *OnAppearing* e utilizzare le *GestureRecognizers*
 - **AathifMahir.Maui.MauiIcons.FontAwesome.Solid** per utilizzare le icone di FontAwesome (catalogo icone: [FontAwesomeIcons](https://fontawesome.com/icons))



CommunityToolkit.Mvvm



The screenshot displays the NuGet package manager interface. On the left, a list of search results for 'CommunityToolkit.Mvvm' is shown. The top result is 'CommunityToolkit.Mvvm' by dotnetfoundation, Microsoft.Toolkit, with 9,35M downloads and version 8.4.0. Below it are 'Verify.CommunityToolkit.Mvvm' (5,38K downloads), 'Lexicom.Mvvm' (26,5K downloads), 'Russkyc.Templates.WPF-MVVM' (14,1K downloads), 'Polaris.PhoneLib.Mvvm' (21,8K downloads), and 'Wpf.Navigation.CommunityToolkit.Mvvm' (1,25K downloads). A disclaimer at the bottom states: 'Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.' and includes a checkbox for 'Don't show this again'.

On the right, the details for 'CommunityToolkit.Mvvm' are shown. The version is set to 'Latest stable 8.4.0' with an 'Install' button. A note indicates 'Package source mapping is off.' with a 'Configure' link. Under the 'Options' section, there is a 'Description' section that reads: 'This package includes a .NET MVVM library with helpers such as:' followed by a list of helpers: ObservableObject, ObservableRecipient, ObservableValidator, RelayCommand, AsyncRelayCommand, WeakReferenceMessenger, StrongReferenceMessenger, and loc.

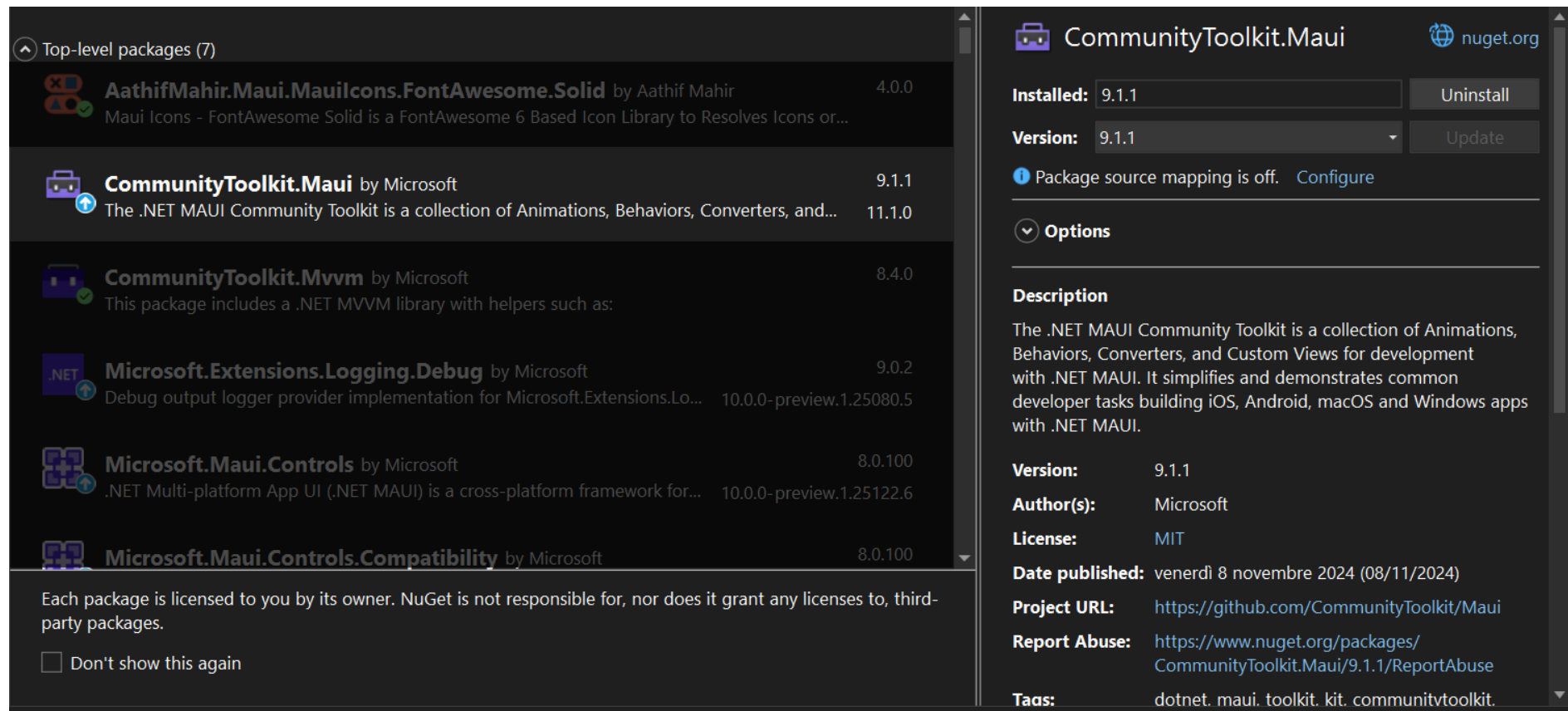
- ObservableObject: a base class for objects implementing the INotifyPropertyChanged interface.
- ObservableRecipient: a base class for observable objects with support for the IMessenger service.
- ObservableValidator: a base class for objects implementing the INotifyDataErrorInfo interface.
- RelayCommand: a simple delegate command implementing the ICommand interface.
- AsyncRelayCommand: a delegate command supporting asynchronous operations and cancellation.
- WeakReferenceMessenger: a messaging system to exchange messages through different loosely-coupled objects.
- StrongReferenceMessenger: a high-performance messaging system that trades weak references for speed.
- loc: a helper class to configure dependency injection

- L'installazione di questo pacchetto **non** richiede setup aggiuntivi.



CommunityToolkit.Maui

► versione 9.1.1



The screenshot displays the NuGet package manager interface. On the left, a list of top-level packages is shown, including 'AathifMahir.Maui.MauiIcons.FontAwesome.Solid', 'CommunityToolkit.Maui' (selected), 'CommunityToolkit.Mvvm', 'Microsoft.Extensions.Logging.Debug', 'Microsoft.Maui.Controls', and 'Microsoft.Maui.Controls.Compatibility'. The right pane shows the details for 'CommunityToolkit.Maui' by Microsoft, version 9.1.1. It includes buttons for 'Uninstall' and 'Update', and a 'Package source mapping is off' message. Below this, the 'Options' section is expanded, showing the 'Description' of the package. The description states: 'The .NET MAUI Community Toolkit is a collection of Animations, Behaviors, Converters, and Custom Views for development with .NET MAUI. It simplifies and demonstrates common developer tasks building iOS, Android, macOS and Windows apps with .NET MAUI.' Other details include the version (9.1.1), author (Microsoft), license (MIT), date published (venerdì 8 novembre 2024 (08/11/2024)), project URL (https://github.com/CommunityToolkit/Maui), report abuse URL (https://www.nuget.org/packages/CommunityToolkit.Maui/9.1.1/ReportAbuse), and tags (dotnet. maui. toolkit. kit. communitvtoolkit).

Top-level packages (7)

- AathifMahir.Maui.MauiIcons.FontAwesome.Solid** by Aathif Mahir 4.0.0
Maui Icons - FontAwesome Solid is a FontAwesome 6 Based Icon Library to Resolves Icons or...
- CommunityToolkit.Maui** by Microsoft 9.1.1
The .NET MAUI Community Toolkit is a collection of Animations, Behaviors, Converters, and... 11.1.0
- CommunityToolkit.Mvvm** by Microsoft 8.4.0
This package includes a .NET MVVM library with helpers such as:
- Microsoft.Extensions.Logging.Debug** by Microsoft 9.0.2
Debug output logger provider implementation for Microsoft.Extensions.Lo... 10.0.0-preview.1.25080.5
- Microsoft.Maui.Controls** by Microsoft 8.0.100
.NET Multi-platform App UI (.NET MAUI) is a cross-platform framework for... 10.0.0-preview.1.25122.6
- Microsoft.Maui.Controls.Compatibility** by Microsoft 8.0.100

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Don't show this again

CommunityToolkit.Maui nuget.org

Installed: 9.1.1 **Uninstall**

Version: 9.1.1 **Update**

Package source mapping is off. [Configure](#)

Options

Description

The .NET MAUI Community Toolkit is a collection of Animations, Behaviors, Converters, and Custom Views for development with .NET MAUI. It simplifies and demonstrates common developer tasks building iOS, Android, macOS and Windows apps with .NET MAUI.

Version: 9.1.1

Author(s): Microsoft

License: MIT

Date published: venerdì 8 novembre 2024 (08/11/2024)

Project URL: <https://github.com/CommunityToolkit/Maui>

Report Abuse: <https://www.nuget.org/packages/CommunityToolkit.Maui/9.1.1/ReportAbuse>

Tags: dotnet. maui. toolkit. kit. communitvtoolkit.




CommunityToolkit.Maui


- ▶ Una volta installato il pacchetto è necessario completare il setup. Seguire le indicazioni nel file *ReadMe.txt*, in alternativa, aprire il [link](#) indicato nel *Project Url* del pacchetto.
- ▶ Come possiamo vedere, viene indicato di aggiornare il *builder* dentro al file **MauiProgram.cs** in questo modo:


```
4 references
public static MauiApp CreateMauiApp()
{
    var builder = MauiApp.CreateBuilder();
    builder
        .UseMauiApp<App>()
        .UseMauiCommunityToolkit()
        .ConfigureFonts(fonts =>
        {
            fonts.AddFont("OpenSans-Regular.ttf", "OpenSansRegular");
            fonts.AddFont("OpenSans-Semibold.ttf", "OpenSansSemibold");
        });
}
```





AathifMahir.Maui.Mauilcons.FontAwesome.Solid

**UraniumUI.Icons.FontAwesome** by enisn, **51K** downloads 2.11.0
Uranium is a Free & Open-Source UI Kit for MAUI.

**AathifMahir.Maui.Mauilcons.FontAwesome** by Aathif_Mahir, **8,74K** downloads 4.0.0
Maui Icons - FontAwesome is a FontAwesome 6 Based Icon Library to Resolves Icons or Font Icon M...

**AathifMahir.Maui.Mauilcons.FontAwesome.Solid** by Aathif_Mahir, **8,14K** downloads 4.0.0
Maui Icons - FontAwesome Solid is a FontAwesome 6 Based Icon Library to Resolves Icons or Font Ic...

**AathifMahir.Maui.Mauilcons.FontAwesome.Brand** by Aathif_Mahir, **6,19K** download: 4.0.0
Maui Icons - FontAwesome Brand is a FontAwesome 6 Based Icon Library to Resolves Icons or Font I...

**Jamjardavies.Icon.Maui.FontAwesome** by Jamjardavies, **1,95K** downloads 1.2.0
Adds FontAwesome icons.

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Don't show this again

**AathifMahir.Maui.Mauilcons.FontAwesome.Solid** nuget.org

Version: Latest stable 4.0.0 Install

 Package source mapping is off. [Configure](#)

 **Options**

Description
Maui Icons - FontAwesome Solid is a FontAwesome 6 Based Icon Library to Resolves Icons or Font Icon Management on .Net Maui by Providing Controls with Complete FontAwesome 6 Icon Collection Built into Library.

Version: 4.0.0
Owner(s): Aathif_Mahir
Author(s): Aathif Mahir
License: MIT
Readme: [View Readme](#)
Downloads: 8,143
Date published: giovedì 17 ottobre 2024 (17/10/2024)
Project URL: <https://github.com/AathifMahir/Mauilcons>
Report Abuse: <https://www.nuget.org/packages/AathifMahir.Maui.Mauilcons.FontAwesome.Solid>



AathifMahir.Maui.MauiIcons.FontAwesome.Solid

- ▶ Anche in questo caso seguiremo le istruzioni della documentazione per il setup post installazione.
- ▶ Come prima, è necessario aggiornare il *builder* dentro al file **MauiProgram.cs**:

```
4 references
public static MauiApp CreateMauiApp()
{
    var builder = MauiApp.CreateBuilder();
    builder
        .UseMauiApp<App>()
        .UseMauiCommunityToolkit()
        .UseFontAwesomeSolidMauiIcons()
        .ConfigureFonts(fonts =>
        {
            fonts.AddFont("OpenSans-Regular.ttf", "OpenSansRegular");
            fonts.AddFont("OpenSans-Semibold.ttf", "OpenSansSemibold");
        });
}
```



Utilizzo in XAML - *namespaces*

- ▶ Per poter accedere ai contenuti delle librerie *CommunityToolkit.Maui* e *FontAwesome.Solid* è necessario aggiungere questi *namespaces* in ogni vista:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              xmlns:toolkit="http://schemas.microsoft.com/dotnet/2022/maui/toolkit"
              xmlns:mi="http://www.aathifmahir.com/dotnet/2022/maui/icons"
              x:Class="ITI_PizzaProject_PROVA.ListPage"
              Title="ListPage">

    <VerticalStackLayout>
        <Label Text="Welcome to .NET MAUI!"
              VerticalOptions="Center"
              HorizontalOptions="Center" />
    </VerticalStackLayout>
</ContentPage>
```



Utilizzo in XAML – *tags*

- ▶ Esempio di utilizzo:

```
<ContentPage.Behaviors>
  <toolkit:EventToCommandBehavior EventName="Appearing"
                                Command="{Binding AppearingCommand}" />
</ContentPage.Behaviors>
```

```
<Button Text="AGGIUNGI PIZZA"
        ImageSource="{mi:FontAwesomeSolid Icon=Plus, IconColor={AppThemeBinding Default={StaticResource White}}}"
        Command="{Binding Source={RelativeSource AncestorType={x:Type views:ListPage}}, Path=NewOrderCommand}"
        Padding="7"
        Margin="20, 10, 20, 20"
        />
```



Link finali

- ▶ Link al *repository* con prototipo di app funzionante e presentazione:
<https://github.com/onitsmart/Project-Work-ITI-MAUI.git>



Contattaci



+39 0547 1956023



info@onitsmart.it



Via dell'Arrigoni, 220
47522 Cesena



www.onit.it