

MAUI

Documentation

Pagina principale per imparare .NET MAUI

<https://dotnet.microsoft.com/en-us/learn/maui>

Esempi guidati di .NET MAUI

<https://learn.microsoft.com/en-us/training/browse/?expanded=dotnet&products=dotnet-maui>

Training - Build mobile and desktop apps with .NET MAUI:

<https://learn.microsoft.com/en-us/training/paths/build-apps-with-dotnet-maui/>

<https://learn.microsoft.com/en-us/events/learn-events/learnlive-mobile-desktop-apps-dotnet-maui/>

.NET MAUI For Beginners (video serie di dotnet – 8 episodi)

<https://youtube.com/playlist?list=PLdo4fOcmZ0oUBAdL2NwBpDs32zwGqb9DY>

.NET MAUI For Beginners (video serie di Montemagno – circa una ventina di episodi):

<https://youtube.com/playlist?list=PLwOF5UVsZWUjNR3roRK79QgBcKLyOX48I>

Video Corso (4 ore): Learn .NET MAUI - Full Course for Beginners | Build cross-platform apps in C#:

https://youtu.be/DuNLR_NJv8U

<https://github.com/dotnet-presentations/dotnet-maui-workshop>

Video serie su MAUI (per approfondimenti) di Daniel Hindrikes

<https://www.youtube.com/playlist?list=PLNfklfxm6YiO7fA0fJNO4rrbFLVik3af>

.NET MAUI Crash Course (Video Serie) di Gerald Versluis:

<https://youtube.com/playlist?list=PLfbOp004UaYVt1En4WW3pVuM-vm66OqZe>

Altri tutorial su MAUI di Gerald Versluis

<https://www.youtube.com/@jfversluis>

Miniserie semplificata su MAUI

<https://jesseliberty.com/2022/06/05/learning-net-maui-posting-0/>

Documentazione ufficiale .NET MAUI

<https://learn.microsoft.com/en-us/dotnet/maui/?view=net-maui-7.0>

<https://github.com/dotnet/maui>

<https://github.com/dotnet/maui/blob/main/.github/DEVELOPMENT.md>

Esempi di codice presenti nella documentazione .NET MAUI:

<https://github.com/dotnet/maui-samples>

Raccolta di risorse per imparare .NET MAUI

<https://github.com/jfversluis/learn-dotnet-maui>

<https://www.andreasnesheim.no/>

MAUI Beginner to Advanced: <https://youtube.com/playlist?list=PL-mV1UKHGpQaWmiVUHPbENNAxExE9TIIJ>

MAUI Android Deeplinking

Fare in modo che quando l'utente clicca su un particolare URL viene aperta una pagina specifica della nostra App

https://youtu.be/SP_CwEmePKE

<http://xamaringuysshow.com/dotnet-maui-deeplinking-android-easy-way/>

Il metodo generale per fare in modo che una App su Android possa rispondere ad un Intent è quello di mettere un Intent Filter sulla MainActivity (in Platforms/android) e poi gestirlo nei metodi OnCreate e OnNewIntent. Si veda ad esempio il progetto <https://github.com/franckbour/Plugin.NFC>

Si può far partire una pagina particolare a seconda dell'Intent ricevuto, usando una sorta di meccanismo di routing interno a MAUI basato su WeakReferenceMessenger (vedere i dettagli qui https://youtu.be/SP_CwEmePKE)

<https://developer.android.com/reference/android/app/Activity>

<https://stackoverflow.com/questions/8357200/how-we-can-use-onnewintent-in-any-activity>

<https://stackoverflow.com/a/43662027>

MAUI App Actions

<https://www.andreasnesheim.no/app-actions-in-net-maui/>

.NET MAUI local database

.NET MAUI local database: SQLite con sqlite-net-pcl

<https://learn.microsoft.com/en-us/dotnet/maui/data-cloud/database-sqlite>

<https://learn.microsoft.com/en-us/training/modules/store-local-data/>

<https://github.com/MicrosoftDocs/mslearn-dotnetmaui-store-local-data>

.NET MAUI local database: SQLite con EF Core (non raccomandato per il momento)

<https://learn.microsoft.com/en-us/ef/core/get-started/xamarin>

<https://stackoverflow.com/questions/72598734/entity-framework-core-in-net-maui-xamarin-forms>

.NET MAUI local database: LiteDb

<https://www.litedb.org/>

<https://youtu.be/m03IjT1b2Hs>

Raccolta di librerie per MAUI

Awesome .NET MAUI

<https://github.com/jsuarezruiz/awesome-dotnet-maui>

.NET Showcase - A curated list of awesome .NET MAUI open-source samples to show how to create goodlooking UI with .NET

<https://github.com/jsuarezruiz/dotnet-maui-showcase>

.NET MAUI & Xamarin.Forms UI Snippets

<https://www.snippets.dev/>

Grafica in C#

C# Data Visualization

<https://swharden.com/csdlv/>

Beautiful, animated, automatically updated, cross-platform, object oriented and easy to use Chart library

<https://lvcharts.com/>

Community Toolkit

<https://github.com/CommunityToolkit/>

<https://github.com/CommunityToolkit/Maui> (CommunityToolkit Sample App)

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/maui/>

Shiny.net

<https://shinylib.net/>

<https://github.com/shinyorg/shiny>

MAUI App Accelerator

<https://marketplace.visualstudio.com/items?itemName=MattLaceyLtd.MauiAppAccelerator>

MediaElement (>= .NET 7)

MediaElement is a control for playing video and audio.

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/maui/views/mediaelement>

<https://github.com/jfversluis/MauiMediaElementSample>

https://youtu.be/_sp4RG0l0x4

Nuget: CommunityToolkit.Maui.MediaElement

<https://www.nuget.org/packages/CommunityToolkit.Maui>

Plugin.Maui.Audio - Play Sound in .NET MAUI with Plugin.Maui.Audio

<https://youtu.be/oIYnEuZ9oew>

<https://github.com/jfversluis/MauiPluginAudioSample>

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/maui/>

Risorse utili

Immagini SVG gratis

<https://www.svgrepo.com/>

Pacchetti utili per applicazioni particolari

NFC

<https://github.com/franckbour/Plugin.NFC>

QR and bar code reader

Pacchetto ZXing:

Repository del pacchetto: <https://github.com/Redth/ZXing.Net.Maui>

Codice di esempio: <https://github.com/jfversluis/MauiZxingBarcodeScannerSample>

Video: https://youtu.be/ostgi2xB_ok

Video: <https://youtu.be/p4BRsdMc7OA>

Pacchetto Camera.Maui:

Repository del pacchetto: <https://github.com/hjam40/Camera.MAUI>

Take Pictures in .NET MAUI with the Camera.MAUI CameraView (video):

<https://youtu.be/nlZSLPf22vI>

Codice di esempio: <https://github.com/jfversluis/MauiCameraMauiSample>

Implement QR Code & Barcode Scanning with Camera.MAUI in .NET MAUI (video):

<https://youtu.be/FuvFrIS9wm0>

Codice di esempio: <https://github.com/jfversluis/MauiCameraMauiBarcodeScanningSample>

Pacchetto BarcodeScanner.Mobile (Powerful barcode scanning library using Google MLKit API):

Repository del pacchetto: <https://github.com/JimmyPun610/BarcodeScanner.Mobile>

Video: <https://youtu.be/WWP2t-B5ADU>

Codice di esempio: <https://github.com/jfversluis/MauiGoogleVisionBarcodeScanningSample>

File Picker in .NET MAUI

<https://youtu.be/Wg1fhr3iwKY>

<https://github.com/jfversluis/MauiFilePickerSample>

Folder Picker in .NET MAUI

Documentazione: <https://learn.microsoft.com/en-us/dotnet/communitytoolkit/maui/essentials/folder-picker>

Video: <https://youtu.be/IOzAWGsWKf4>

.NET MAUI Community Toolkit Repository: <https://github.com/CommunityToolkit/Maui>

Codice di esempio: <https://github.com/jfversluis/MauiToolkitFolderPickerSample>

File Saver in .NET MAUI

Documentazione: <https://learn.microsoft.com/en-us/dotnet/communitytoolkit/maui/essentials/file-saver>

Video: <https://youtu.be/Q9T-dRYq3Ps>

Codice di esempio: <https://github.com/jfversluis/MauiToolkitFileSaverSample>

.NET MAUI Community Toolkit Repository: <https://github.com/CommunityToolkit/Maui>

Blog di David Britch

<https://www.davidbritch.com/>

Blog di Andreas Nesheim

<https://www.andreasnesheim.no/>

Progetti particolari

Building an Audio Recorder and Player App in .NET MAUI

<https://github.com/SyncfusionExamples/dotnet-maui-audio-recorder-and-player->

Implement Speech to Text .NET MAUI App

<https://github.com/jfversluis/MauiSpeechToTextSample>

https://youtu.be/CI-Fx8_0oYo

MAUI Learning path

Cos'è MAUI?

https://youtu.be/Hh279ES_FNQ

.NET MAUI Tutorial, Hello World for Android and Windows in C#

Setup dell'ambiente di sviluppo, Hyper-V, emulatore Android, connessione al telefono fisico mediante debug USB - <https://dotnet.microsoft.com/en-us/learn/maui/first-app-tutorial/intro>

Unità di apprendimento "Creare app per dispositivi mobili e desktop con .NET MAUI"

<https://learn.microsoft.com/en-us/training/modules/build-mobile-and-desktop-apps/>

Learning objectives

In this module, you will:

- Learn the basic architecture of .NET MAUI
- Create .NET MAUI app
- Define a shared UI for the platforms supported by .NET MAUI
- Deploy a .NET MAUI app from Visual Studio
- Dial a number from within the app

Unità di apprendimento "Create a UI in a .NET MAUI app by using XAML"

<https://learn.microsoft.com/en-us/training/modules/create-user-interface-xaml/>

Learning objectives

By the end of this module, you'll be able to describe:

- The benefits of using XAML over defining the UI for a .NET MAUI app in C#
- The types available for defining a .NET MAUI app using XAML, and the properties these types expose
- How to handle UI events and wire them up in XAML

Fino alla sezione "Create your first XAML Page"

Pagine e Navigazione

ContentPage

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/pages/contentpage>

Navigazione gerarchica - NavigationPage

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/pages/navigationpage>

Esempi:

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Navigation/HierarchicalNavigationDemo>

TabbedPage

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/pages/tabbedpage>

Esempi:

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Navigation/TabbedPage>

Con l'esempio della TabbedPage vengono affrontati anche i seguenti argomenti:

- a) Referenziare una pagina XAML all'interno di un'altra pagina XAML. Ad esempio, inserire la MainPage.xaml all'interno di un Tab:
 - 1) Si aggiunge il namespace local come attributo della pagina nella quale si vuole inserire il riferimento ad un'altra pagina
 - 2) Si utilizza <local:MainPage .../> per referenziare la pagina
- b) Fare riferimento a configurazioni specifiche di una piattaforma (Ad esempio, Android), con il riferimento al namespace dell'assembly di MAUI che gestisce queste configurazioni.

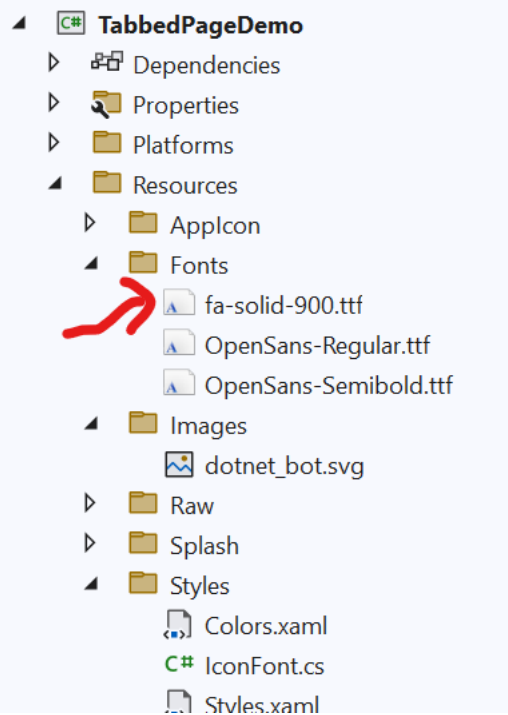
Nell'esempio fatto:

```
<?xml version="1.0" encoding="utf-8" ?>
<TabbedPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:TabbedPageDemo;assembly=TabbedPageDemo"
             xmlns:android="clr-namespace:Microsoft.Maui.Controls.PlatformConfiguration.AndroidSpecific;assembly=Microsoft.Maui.Controls"
             android:TabbedPage.ToolbarPlacement="Bottom"
             x:Class="TabbedPageDemo.TabbedPageApp"
             Title="TabbedPageApp">
    <!--questo è un commento-->
    <ContentPage Title="One" IconImageSource="{StaticResource TabOne}" >
        <ScrollView>
            <VerticalStackLayout>
                <Label Text="Ciao dalla pagina One"/>
                <Label Text="Ciao ciao"/>
            </VerticalStackLayout>
        </ScrollView>
    </ContentPage>
    <ContentPage Title="Two" IconImageSource="{StaticResource TabTwo}" >
        <ScrollView>
            <VerticalStackLayout>
                <Label Text="Ciao dalla pagina Two"/>
                <Label Text="Ciao ciao"/>
            </VerticalStackLayout>
        </ScrollView>
    </ContentPage>
    <!--in questo caso mettiamo una pagina all'interno di una navigation page-->
    <NavigationPage Title="Three" IconImageSource="{StaticResource TabThree}">
        <x:Arguments>
            <local:MainPage Title="Three"/>
        </x:Arguments>
    </NavigationPage>
    <!--in questo caso mettiamo una pagina definita localmente-->
    <local:MainPage Title="Quattro" IconImageSource="dotnet_bot.svg"/>
</TabbedPage>
```

c) Utilizzare FontAwesome per creare icone da inserire nella propria app

- 1) FontAwesome: <https://fontawesome.com/>
- 2) Download fonts: <https://fontawesome.com/download> (scaricare una versione free, ad esempio, per il web, ad esempio <https://use.fontawesome.com/releases/v6.2.1/fontawesome-free-6.2.1-web.zip>)
- 3) Scompattare il file e osservare il contenuto. Per vedere i font si possono aprire i singoli file in formato svg (grafica vettoriale). Come visualizzatore si può usare un browser come Chrome, oppure Edge. Se si vuole avere la possibilità di visualizzare tutti i font, per avere un'anteprima di come verranno "renderizzati", occorre un programma specifico. Si può usare un plugin di Visual Studio Code come Font Preview:
<https://marketplace.visualstudio.com/items?itemName=ctcuff.font-preview>
Bisogna aprire i file nel formato supportato, ad esempio, ttf (true type font).
Si può usare anche una web application come, ad esempio, FontDrop <https://fontdrop.info>, che consente di caricare il file ttf e di visualizzare l'anteprima dei font.
- 4) Importare i font nella propria applicazione MAUI:

a. Aggiungere il file di font all'applicazione MAUI



b. Registrare il file dei font mediante MauiProgram:

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



```

        fonts.AddFont("OpenSans-Semibold.ttf",
"OpenSansSemibold");
        fonts.AddFont("fa-solid-900.ttf", "FontAwesome");
    });

    return builder.Build();
}
}

```

- c. Definire una classe statica che riporti i codici Unicode per i font presenti nel file ttf. Questa classe può essere generata automaticamente, mediante il pacchetto IconFont2Code <https://github.com/andreinitescu/IconFont2Code> , oppure mediante la sua versione live web: <https://andreinitescu.github.io/IconFont2Code/> . Qui basta caricare il file di font in uno dei formati supportati (ad esempio, ttf) e copiare la classe C# generata automaticamente: viene creata la classe statica.

```

// Generated by IconFont2Code: https://andreinitescu.github.io/IconFont2Code
// If you change the class to 'public' in order to use it outside its
assembly
// you should also change the fields from 'const' to 'public static
readonly'
static class IconFont
{
    //altri codici
    public const string Dice = "\uf522";
    public const string DiceFive = "\u2684";
    public const string DiceFour = "\u2683";
    public const string DiceOne = "\u2680";
    public const string DiceSix = "\u2685";
    public const string DiceThree = "\u2682";
    public const string DiceTwo = "\u2681";
    //altri codici
}

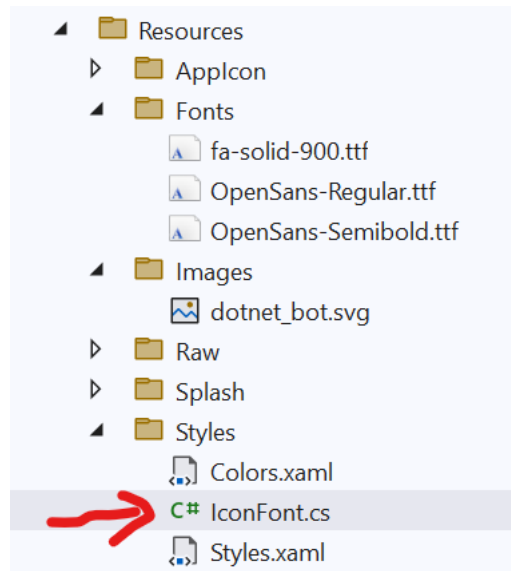
```

La classe può essere inserita in Resource/Styles, lasciando come namespace quello principale dell'applicazione. Ad esempio:

```

namespace TabbedPageDemo
{
    // Generated by IconFont2Code: https://andreinitescu.github.io/IconFont2Code
    // If you change the class to 'public' in order to use it outside its assembly
    // you should also change the fields from 'const' to 'public static readonly'
    static class IconFont
    {
        public const string Exclamation = "\u0021";
        public const string Hashtag = "\u0023";
        public const string DollarSign = "\u0024";
    }
}
//etc

```



- d. Modificare gli stili nel dizionario di risorse degli stili (file Styles.xaml), aggiungendo le parti evidenziate in giallo (namespace e assembly devono corrispondere a quelli del proprio progetto):

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xaml-comp compile="true" ?>
<ResourceDictionary
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:TabbedPageDemo;assembly=TabbedPageDemo">
```

```
<FontImage x:Key="TabOne"
    FontFamily="FontAwesome"
    Size="22"
    Color="{StaticResource Primary}"
    Glyph="{x:Static local:IconFont.DiceOne}"/>
<FontImage x:Key="TabTwo"
    FontFamily="FontAwesome"
    Size="22"
    Color="{StaticResource Primary}"
    Glyph="{x:Static local:IconFont.DiceTwo}"/>
<FontImage x:Key="TabThree"
    FontFamily="FontAwesome"
    Size="22"
    Color="{StaticResource Primary}"
    Glyph="{x:Static local:IconFont.DiceThree}"/>
<FontImage x:Key="TabFour"
    FontFamily="FontAwesome"
    Size="22"
    Color="{StaticResource Primary}"
    Glyph="{x:Static local:IconFont.DiceFour}"/>
<FontImage x:Key="TabFive"
    FontFamily="FontAwesome"
    Size="22"
    Color="{StaticResource Primary}"
    Glyph="{x:Static local:IconFont.DiceFive}"/>
<FontImage x:Key="TabSix"
    FontFamily="FontAwesome"
```

```

        Size="22"
        Color="{StaticResource Primary}"
        Glyph="{x:Static local:IconFont.DiceSix}"/>
<!--le cose sotto sono quelle inserite di default dal template di progetto-->

```

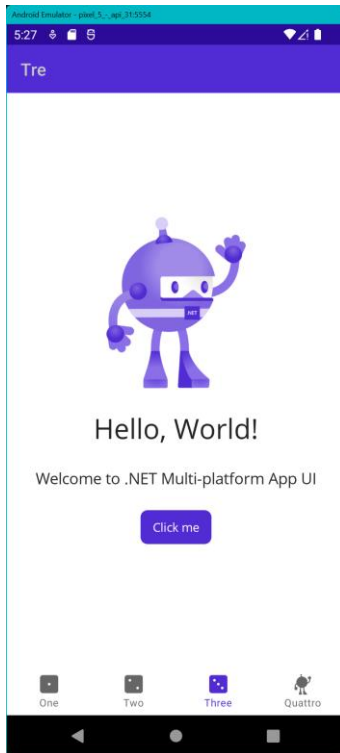
- e. Definire i riferimenti ai FontImage al posto delle immagini nella pagina principale dell'App:

```

<?xml version="1.0" encoding="utf-8" ?>
<TabbedPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:TabbedPageDemo;assembly=TabbedPageDemo"
    xmlns:android="clr-namespace:Microsoft.Maui.Controls.PlatformConfiguration.AndroidSpecific;assembly=Microsoft.Maui.Controls"
    android:TabbedPage.ToolbarPlacement="Bottom"
    x:Class="TabbedPageDemo.TabbedPageApp"
    Title="TabbedPageApp">
    <!--questo è un commento-->
    <ContentPage Title="One" IconImageSource="{StaticResource TabOne}" >
        <ScrollView>
            <VerticalStackLayout>
                <Label Text="Ciao dalla pagina One"/>
                <Label Text="Ciao ciao"/>
            </VerticalStackLayout>
        </ScrollView>
    </ContentPage>
    <ContentPage Title="Two" IconImageSource="{StaticResource TabTwo}" >
        <ScrollView>
            <VerticalStackLayout>
                <Label Text="Ciao dalla pagina Two"/>
                <Label Text="Ciao ciao"/>
            </VerticalStackLayout>
        </ScrollView>
    </ContentPage>
    <!--in questo caso mettiamo una pagina all'interno di una navigation page-->
    <NavigationPage Title="Three" IconImageSource="{StaticResource TabThree}">
        <x:Arguments>
            <local:MainPage Title="Three"/>
        </x:Arguments>
    </NavigationPage>
    <!--in questo caso mettiamo una pagina definita localmente-->
    <local:MainPage Title="Quattro" IconImageSource="dotnet_bot.svg"/>
</TabbedPage>

```

Se tutto è stato fatto correttamente si possono vedere i dadi come icone dei tab:



Con la stessa procedura, seguita per importare i font di FontAwesome, si possono importare anche altri tipi di font. A titolo di esempio, i font di <https://lineicons.com/download/> .

FlyoutPage

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/pages/flyoutpage>

Esempi:

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Navigation/FlyoutPageSample>

Tipi di Layout di Pagina

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts>

Grid

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/grid>

AbsoluteLayout

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/absolutelayout>

FlexLayout

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/flexlayout>

StackLayout, HorizontalStackLayout, VerticalStackLayout

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/stacklayout>

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/horizontalstacklayout>

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/verticalstacklayout>

Esempi:

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Layouts>

Persistenza in MAUI

Salvare in maniera persistente i dati localmente sul dispositivo.

<https://learn.microsoft.com/en-us/training/modules/store-local-data/2-compare-storage-options>

Preferences

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/preferences>

<https://learn.microsoft.com/en-us/dotnet/api/microsoft.maui.storage.preferences>

Esempi:

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Persistence/MauiStorageDemo>

Local File System (File System Helpers)

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers>

Cache directory

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers#cache-directory>

App data directory

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers#app-data-directory>

Bundled files

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers#bundled-files>

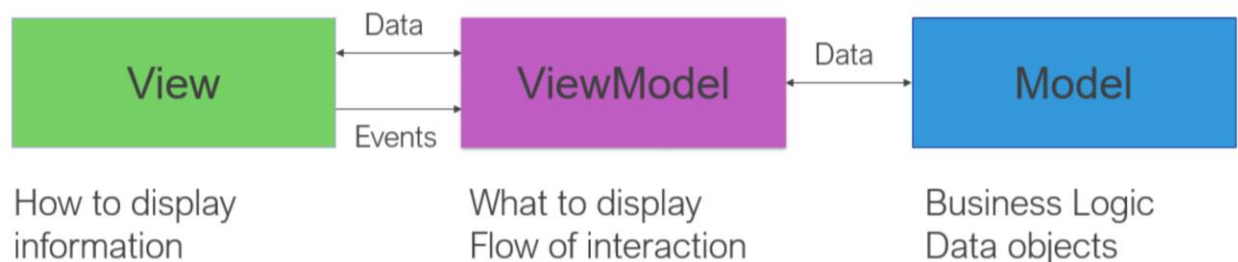
Esempi:

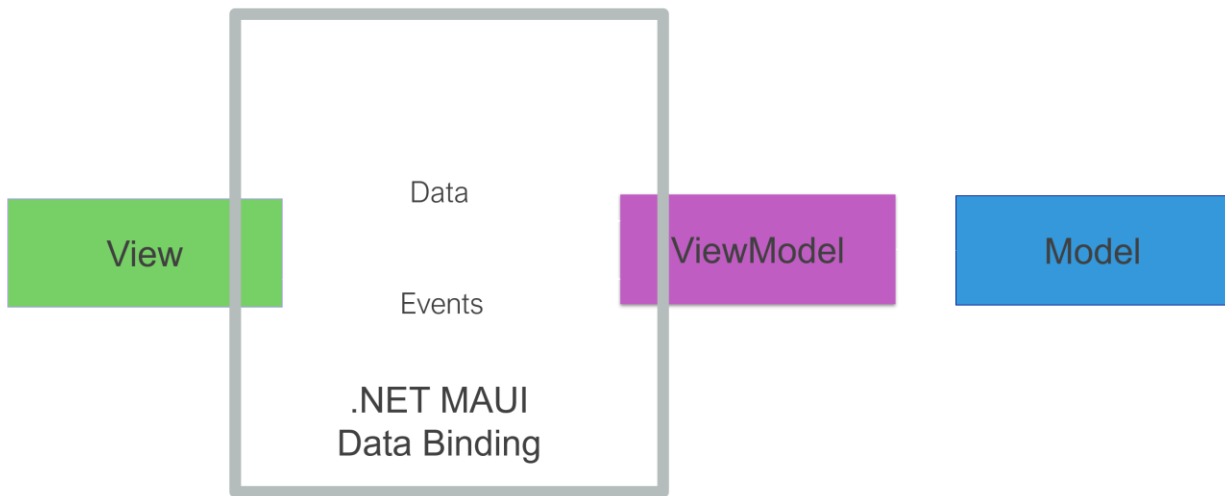
<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Persistence/MauiStorageDemo>

Data Binding

Data Binding and MVVM

<https://learn.microsoft.com/en-us/dotnet/maui/xaml/fundamentals/mvvm>





<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>

Esempi:

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/DataBinding/XamlSamples/XamlSamples>

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/DataBinding/RefreshViewDemo> (con NET 7)

Data Binding (documentazione ed esempi più dettagliati)

<https://learn.microsoft.com/en-us/dotnet/maui/fundamentals/data-binding>

Analisi degli esempi riportati nella documentazione (quasi tutti: il multi-Binding viene mostrato solo con riferimento al StringFormat) con riferimento al codice del progetto che si trova negli esempi riportati di seguito. Gli esempi sul Binding in combinazione con ICommand nella guida e nel corrispondente esempio DataBindingDemos sono abbastanza complessi, si possono capire dopo aver ben compreso gli esempi di Binding presenti in XamlSamples.

Esempi:

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/DataBinding/DataBindingDemos>

Nell'esempio DataBindingDemos sono state aggiunte anche alcune pagine non presenti nella documentazione ufficiale MAUI. Ad esempio, la pagina che mostra come aggiornare continuamente l'orario mostrato in una View, attraverso i concetti di:

`Application.Current.Dispatcher`

`MainThread.BeginInvokeOnMainThread`

MVVM

<https://stackoverflow.com/a/73839274>

<https://stackoverflow.com/a/73917792>

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/appmodel/main-thread>

MVVM Community Toolkit

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/>

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/overview>

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/relaycommand>

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/inotifypropertychanged>

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/observableproperty>

<https://github.com/dotnet-presentations/dotnet-maui-workshop/tree/main/Part%20%20-%20MVVM>

<https://youtu.be/aCxl0z04BN8>

https://youtu.be/DuNLR_NJv8U

<https://github.com/dotnet-presentations/dotnet-maui-workshop>

Esempi con l'uso di CommunityToolkit.Mvvm:

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/DataBinding/CommunityToolkitMVVMDemos/XamlSamples>

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/DataBinding/CommunityToolkitMVVMDemos/RefreshViewDemo>

Shell

<https://learn.microsoft.com/en-us/dotnet/maui/fundamentals/shell>

Introduzione alla Shell con le prime pagine della documentazione Microsoft e con gli esempi riportati di seguito. Gli esempi (ad eccezione di Xaminals) sono stati modificati rispetto a quelli del repository ufficiale degli esempi MAUI.

TabBar

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Navigation/ShellTabBarSample>

Flyout

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Navigation/ShellFlyoutSample>

Flyout e Tab assieme

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Navigation/ShellMixedSample>

Un esempio completo con anche il passaggio dei parametri e la funzione di ricerca (questo è l'esempio che è utilizzato dalle pagine della documentazione Microsoft della Shell):

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Navigation/Shell/Xaminals>

Un esempio completo di App .NET MAUI - MAUI Workshop (.NET 7)

Di James Montemagno:

https://youtu.be/DuNLR_NJv8U

<https://github.com/dotnet-presentations/dotnet-maui-workshop>

Versione alternativa, con spiegazione meno dettagliata, ma più veloce a cura di Gerald Versluis

<https://youtube.com/playlist?list=PLfbOp004UaYVt1En4WW3pVuM-vm66OqZe>

<https://github.com/jfversluis/dotnet-maui-workshop>

Local database in .NET MAUI (SQLite con il pacchetto sqlite-net-pcl)

Parte di teoria da:

<https://learn.microsoft.com/en-us/dotnet/maui/data-cloud/database-sqlite>

con esempi da:

<https://learn.microsoft.com/en-us/samples/dotnet/maui-samples/database-sqlite/>

<https://github.com/dotnet/maui-samples/tree/main/6.0/Data/ToDoSQLite>

Esercitazione guidata:

<https://learn.microsoft.com/en-us/training/modules/store-local-data/>

<https://github.com/MicrosoftDocs/mslearn-dotnetmaui-store-local-data>

Progetti funzionanti nel repository scolastico:

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Persistence/ToDoSQLite>

<https://github.com/GreppiDev/Info4IA2223MAUI/tree/main/Persistence/People>