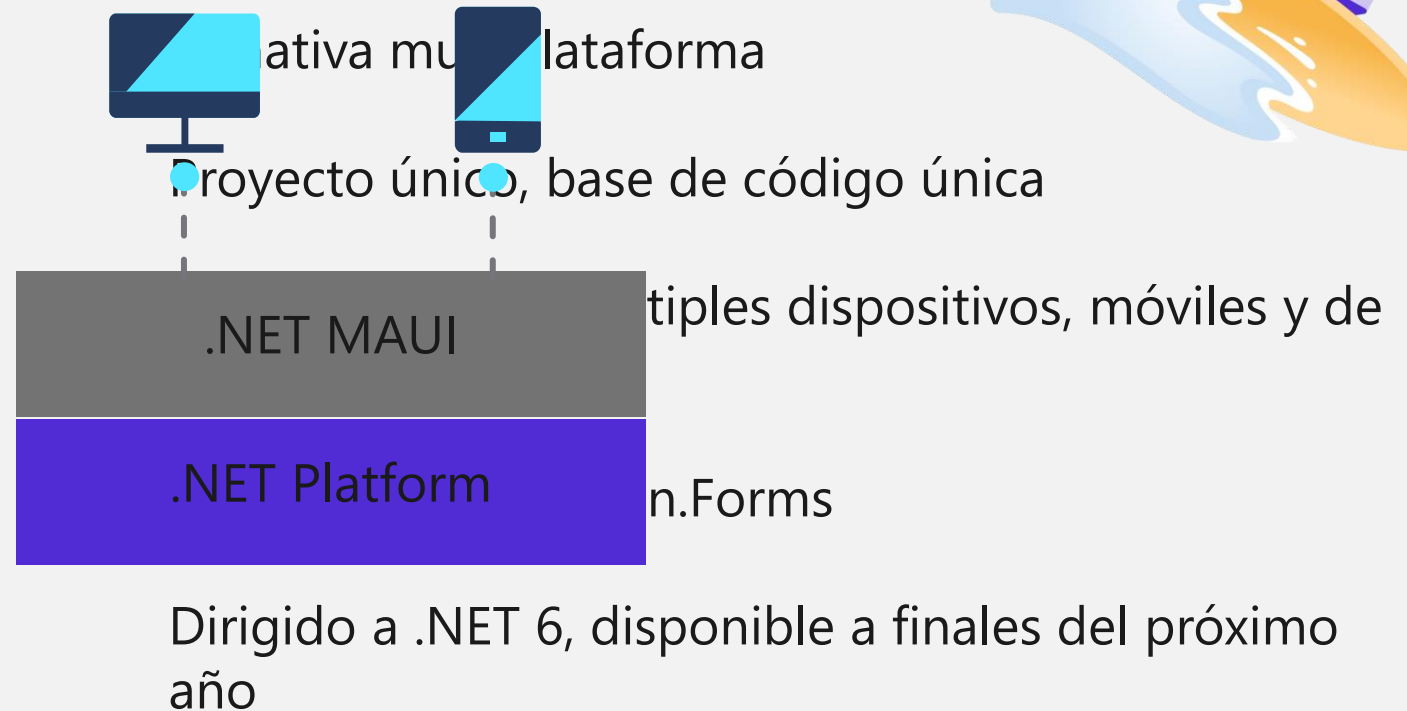


¿Qué es .NET MAUI?

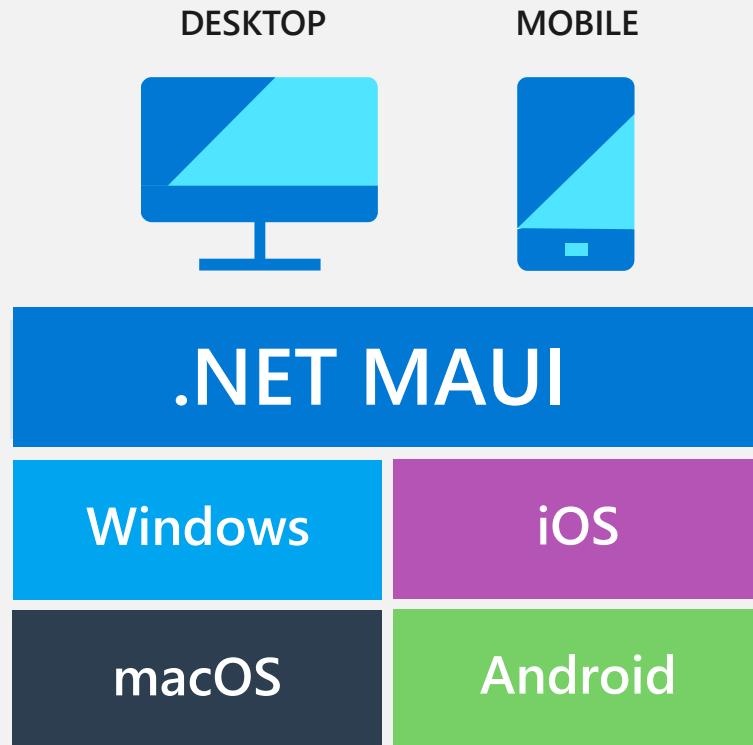
.NET MAUI es la evolución de Xamarin.Forms
extendiendo escenarios de movilidad a
escritorio con un framework de UI realizado
para la búsqueda de rendimiento y
extensibilidad.

La evolución de Xamarin.Forms



Crear interfaces de usuario atractivas para cualquier dispositivo

.NET MAUI



Compatibilidad

- **Cambios mínimos** de API: el objetivo es solo añadir valor (rendimiento, ciclo de desarrollo, etc.)
- Soporte de **migración** para Xamarin y Xamarin.Forms
- Soporte a **Xamarin.Forms** durante un año después del lanzamiento de .NET MAUI

Proyectos SDK-style

Proyecto único

Soporte CLI

.NET 6 BCL

Multi-paradigma

- Interfaz de usuario compatible con XAML y C#
- Listo para Blazor y C# MVU

File | New

- Multi-platform App UI (.NET)

CLI

- `dotnet install maui`
- `dotnet new maui`

Namespaces

- `System.Maui` (previously `Xamarin.Forms`)
- `System.Device` (previously `Xamarin.Essentials`)

Los objetivos de .NET MAUI

Calidad

12 meses de releases con bug fixing y cambios para mejorar la calidad en Xamarin.Forms.

Reorganización de la estructura del proyecto.

Nomenclatura y estructura de clases estandarizadas.

Rendimiento

Arranque inicial más rápido.

Reducir anidación de vistas.

Diseño

Permitir extender controles más fácilmente

Cross-platform templating

Plataformas

Android

iOS

macOS

Windows

Paridad en APIs

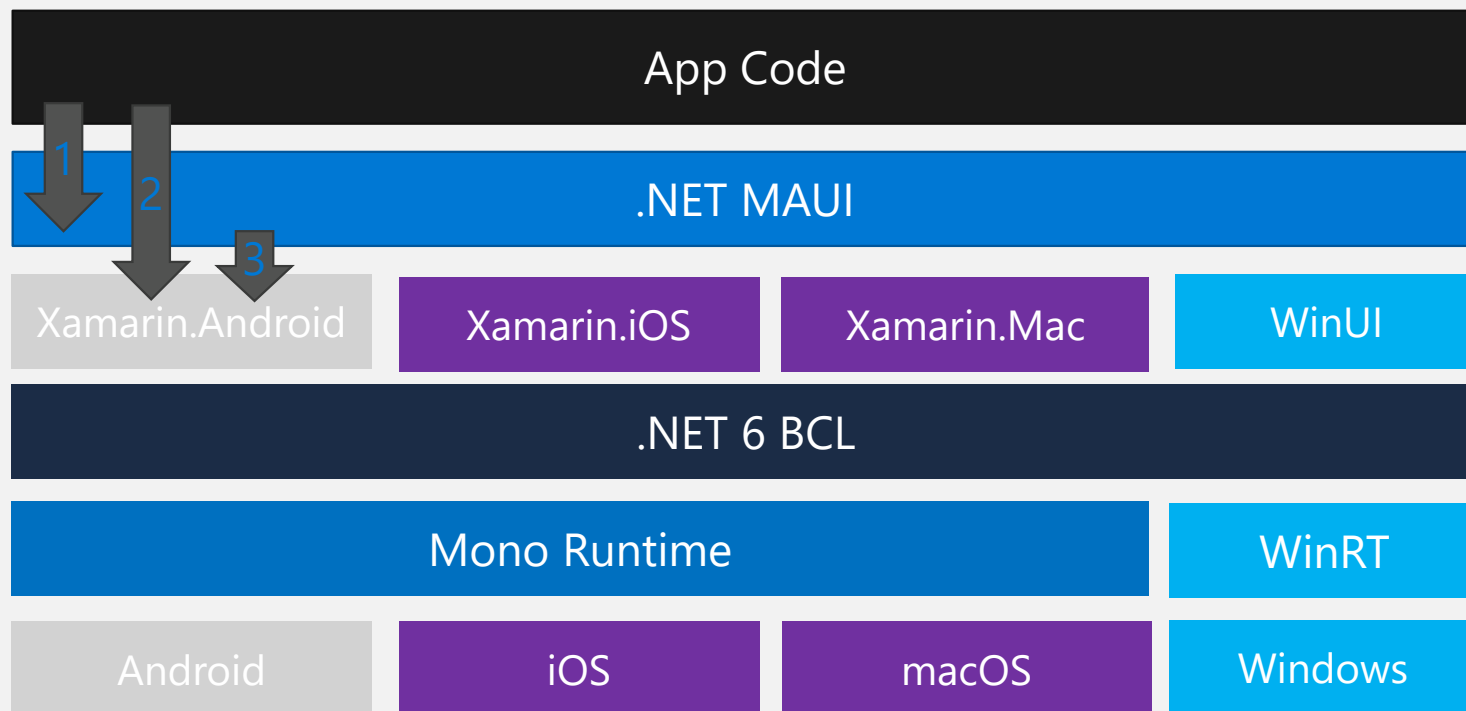
La arquitectura de .NET MAUI

Las API de Android, iOS, macOS y Windows están unificadas en una API abstracta que permite una experiencia de desarrollo que permita escribir una vez y ejecutar en cualquier plataforma, al tiempo que proporciona un acceso total a todos los aspectos de cada plataforma nativa.

App Code interactúa principalmente con .NET MAUI API (1).

Según sea necesario, App Code puede acceder directamente las API de la plataforma (2) a través Handlers, etc.

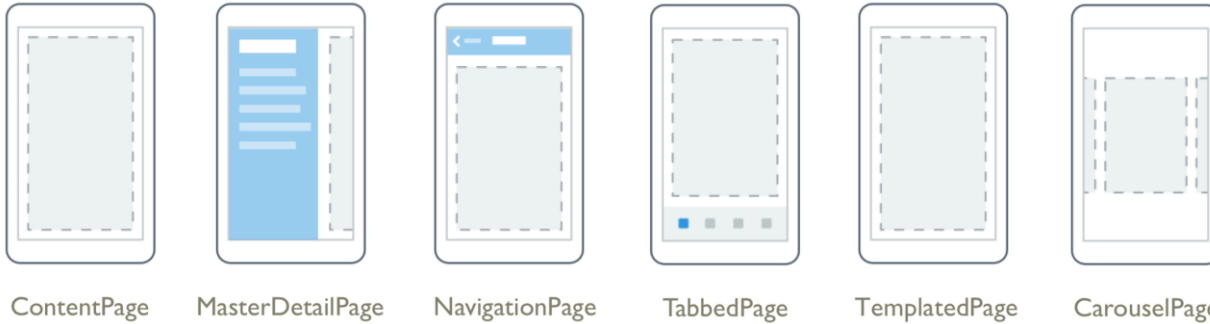
.NET MAUI accede directamente las API de la plataforma nativa (3).



¿Qué incluye .NET MAUI?

- 50+ páginas, layouts, y controles
 - (Se pueden usar desde code behind o XAML)
- Two-way data binding
- Navegación
- Custom handlers
- API de animaciones
- Gestión de dependencias

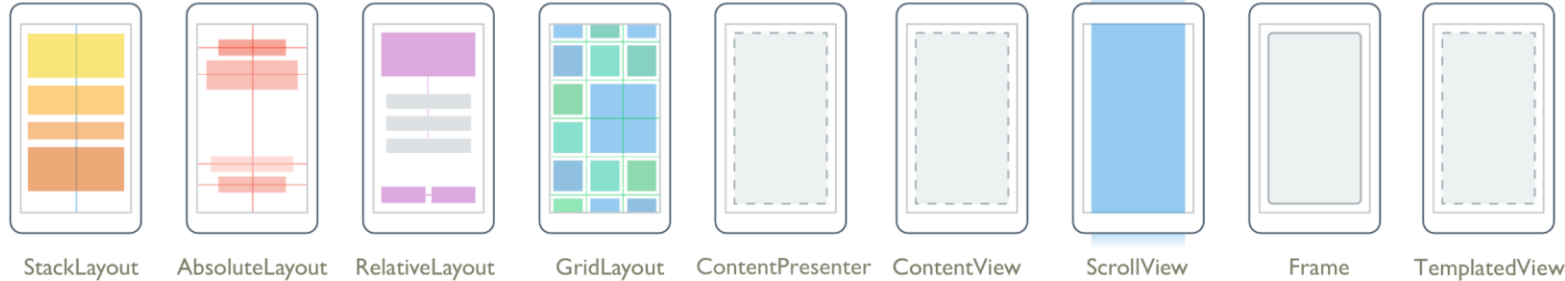
Páginas



Una Page se asigna a un controlador de vista en iOS y una página en Windows. En Android, cada página ocupa la pantalla como una actividad, pero las páginas de .NET MAUI no son actividades.

- **ContentPage:** muestra una sola vista, a menudo un contenedor como StackLayout o ScrollView.
- **MasterDetailPage:** administra dos paneles de información.
- **NavigationPage:** administra la navegación y la experiencia del usuario en una pila de páginas.
- **TabbedPage:** permite la navegación entre páginas secundarias usando pestañas.
- **TemplatedPage:** muestra contenido a pantalla completa con una plantilla de control; la clase base para una página de contenido.
- **CarouselPage:** Permite gestos de deslizamiento entre subpáginas, como una galería.

Layouts



La clase Layout en .NET MAUI es un subtipo especializado de View, que actúa como contenedor para otros Layouts o Views. Por lo general, contiene lógica para establecer la posición y el tamaño de los elementos secundarios en las aplicaciones de .NET MAUI.

- **StackLayout:** coloca elementos secundarios en una sola línea, orientados vertical u horizontalmente. Establece los límites secundarios automáticamente durante un ciclo de diseño. Los límites asignados por el usuario se sobrescribirán y, por lo tanto, el usuario no debe establecerlos en un elemento secundario.
- **AbsoluteLayout:** coloca los elementos secundarios en las posiciones solicitadas absolutas. Los delimitadores y límites asignados por el usuario definen la posición y el tamaño del control.
- **RelativeLayout:** utiliza restricciones para diseñar sus elementos secundarios.
- **Grid:** contiene vistas organizadas en filas y columnas.
- **ContentPresenter:** un administrador de diseño para vistas con plantilla que se usa dentro de un ControlTemplate para marcar dónde aparece el contenido presentado.
- **ContentView:** Un elemento con un solo contenido, que tiene muy poco uso propio. Su propósito es servir como clase base para vistas compuestas definidas por el usuario.
- **ScrollView:** Un elemento capaz de desplazarse si su contenido lo requiere.
- **Frame:** un elemento que contiene un solo elemento secundario con algunas opciones de marco. Este diseño tiene un Padding predeterminado de 20.
- **TemplatedView:** un elemento que muestra contenido con una plantilla de control y la clase base para ContentView.
- **FlexLayout:** se utiliza para organizar vistas horizontal o verticalmente con ajuste.

Views (controles o widgets)

.NET MAUI usa la palabra View para referirse a objetos visuales como botones, texto o cuadros de entrada de texto, que se conocen más comúnmente como controles o widgets.

ActivityIndicator

BoxView

Button

DatePicker

Editor

Entry

Image

Label

ListView

Map

CollectionView

Picker

ProgressBar

SearchBar

Slider

Stepper

TableView

TimePicker

WebView

EntryCell

ImageCell

SwitchCell

TextCell

ViewCell