

MEMORIA

DESARROLLO DE INTERFACES



Alejandro Roberto Chiralt
2ºDAM

INDICE

INTRODUCCIÓN:.....	3
OBJETIVOS:.....	3
TAREAS A REALIZAR:.....	3
ACTIVIDADES :.....	4
DIAGRAMA DE GANT:.....	8
PROBLEMAS:.....	9
CONCLUSIÓN:.....	9
WEBGRAFIA:.....	9

INTRODUCCIÓN:

La publicación de aplicaciones es un paso crucial en el desarrollo de software, ya que permite distribuir y ejecutar las aplicaciones en los sistemas de los usuarios finales. Este proyecto se centra en la publicación de aplicaciones desarrolladas con .NET MAUI en el sistema operativo Windows. Mediante el uso de Visual Studio, se genera un instalador en formato MSIX junto con un certificado digital y los scripts necesarios para facilitar la instalación y configuración en otros equipos. Este proceso, conocido como *sideloading*, permite distribuir las aplicaciones sin depender de tiendas oficiales como Microsoft Store, garantizando flexibilidad en el despliegue y prueba de los desarrollos.

OBJETIVOS:

Comprender el proceso de publicación de aplicaciones desarrolladas con .NET MAUI para Windows.

1. Aprender a generar instaladores en formato MSIX con Visual Studio.
2. Configurar y utilizar certificados digitales para garantizar la seguridad en la instalación de las aplicaciones.
3. Realizar pruebas de despliegue (*deploy*) en equipos externos mediante el método de *sideloading*.
4. Evaluar alternativas de distribución de aplicaciones fuera de las tiendas oficiales.

TAREAS A REALIZAR:

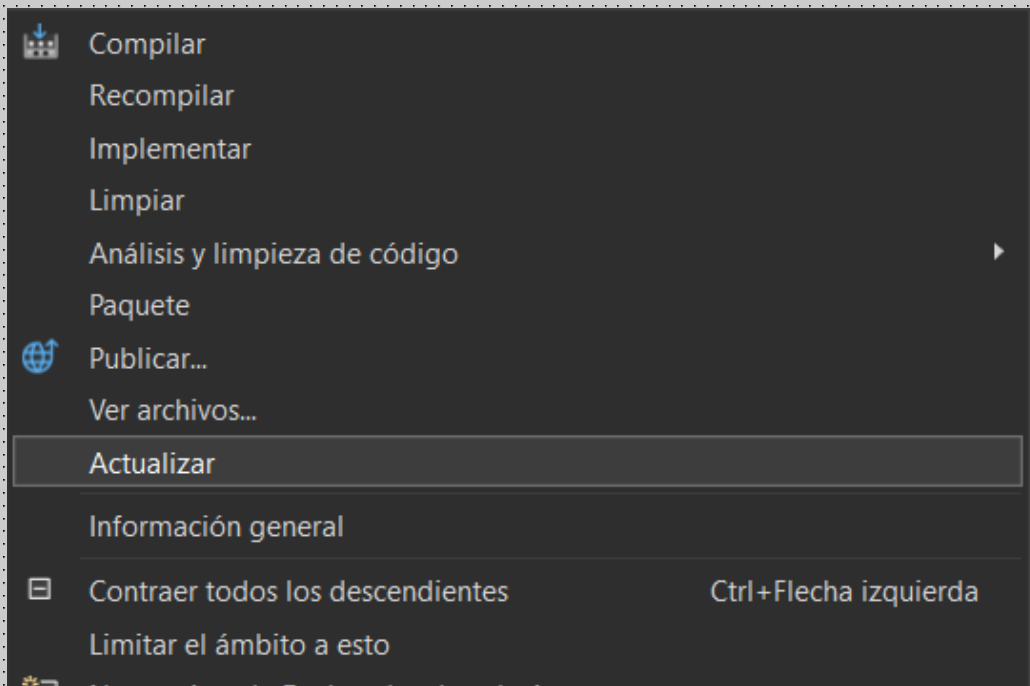
Configurar Visual Studio para habilitar las opciones de publicación de proyectos .NET MAUI.

1. Crear un instalador MSIX, incluyendo la generación de un certificado digital y los scripts necesarios.
2. Desactivar actualizaciones automáticas y seleccionar opciones de empaquetado adecuadas para el proyecto.
3. Probar la instalación en un equipo externo, asegurando que el script de instalación ejecute todos los pasos necesarios.
4. Documentar el proceso completo, desde la configuración inicial hasta la ejecución de la aplicación publicada en Windows.

5. Explorar métodos alternativos de distribución, como la publicación en una página web o la creación de instaladores personalizados.

ACTIVIDADES :

Lo primero que tenemos que hacer es actualizar la app, para evitar problemas de compatibilidad.

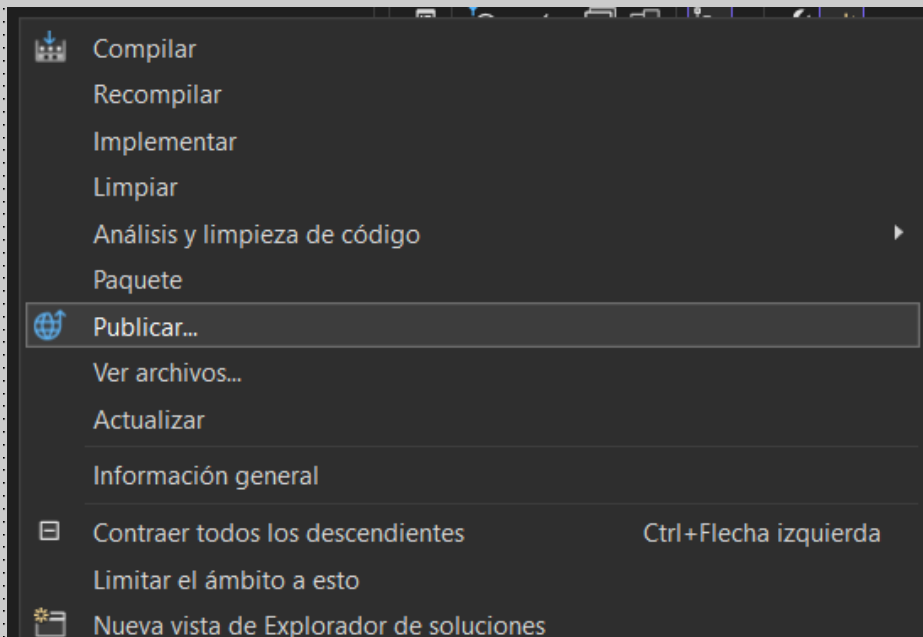


Si no lo tenemos descargado ya, nos va a mandar descargar el UpgradeAssistant, lo descagamos y ya.



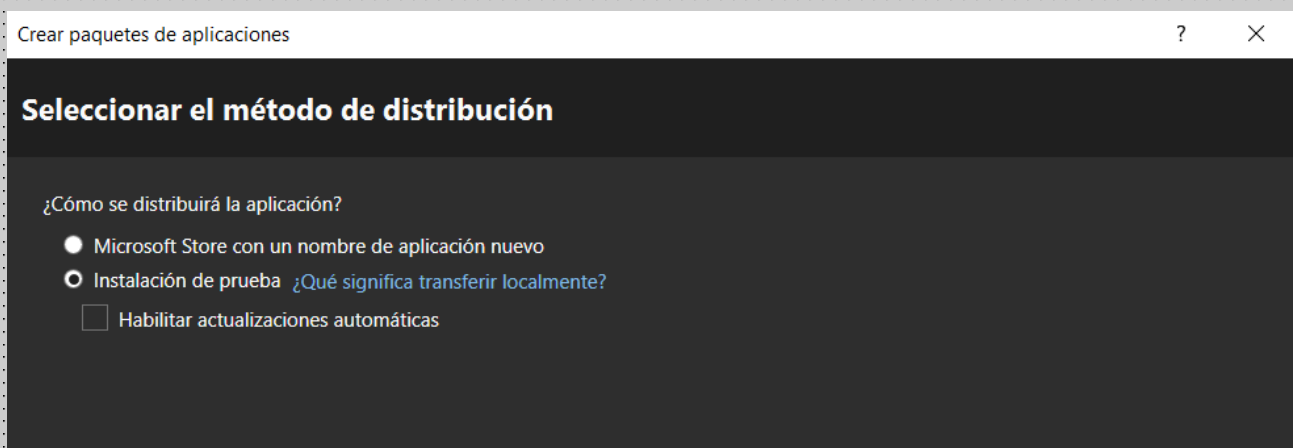
Microsoft.NET.UpgradeAssistant.vsix

Lo siguiente que tenemos que hacer es darle a publicar, que se situa dándole click derecho en el Proyecto y te aparecerá el apartado de Publicar.



Una vez le damos, nos saldrán distintas pestañas.

En esta tenemos que darle a la segunda opción. Para no publicar la app en ninguna plataforma.



Le damos a usar certificado y nos pedirá que lo creemos, lo creamos y lo añadimos.


Crear paquetes de aplicaciones ? X

Seleccionar método de firma

¿Quiere firmar este paquete de aplicación?

☐ No, omitir la firma del paquete.
Los paquetes de la aplicación deben estar firmados para poder instalarse. Tendrá que firmarlos externamente.
[¿Cómo firmo un paquete de la aplicación?](#)

☒ Sí, use el certificado actual:
Asunto: CN=Alejandro
Huella digital: C3D942B4E7A28F2EAEDA807A0976B2483C71EAB1
Fecha de expiración: 16/12/2025 23:49:40

 Visual Studio no puede importar este certificado a un almacén de confianza.
Para confiar en el certificado, impórtelo manualmente.
[Obtenga más información sobre la firma de certificados.](#)

Quitar Detalles

Dirección URL del servidor de marca de tiempo (opcional):

Algoritmo de firma:
SHA256 ▼

En la pestaña esta, tenemos que añadir el perfil de la publicación.

Crear paquetes de aplicaciones ? X

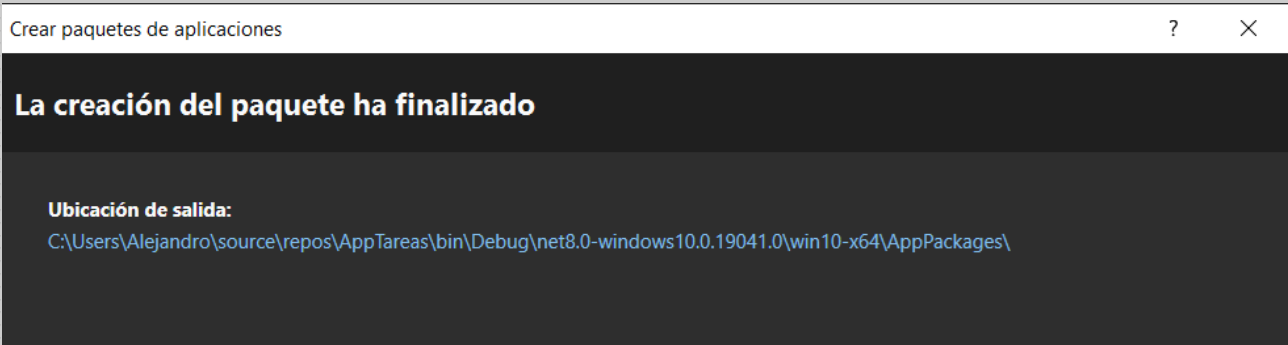
Seleccionar y configurar paquetes

Versión:
0 . 0 . 2 . 0
☒ Incrementar automáticamente
[¿Cómo se usa la numeración de versión del paquete?](#)

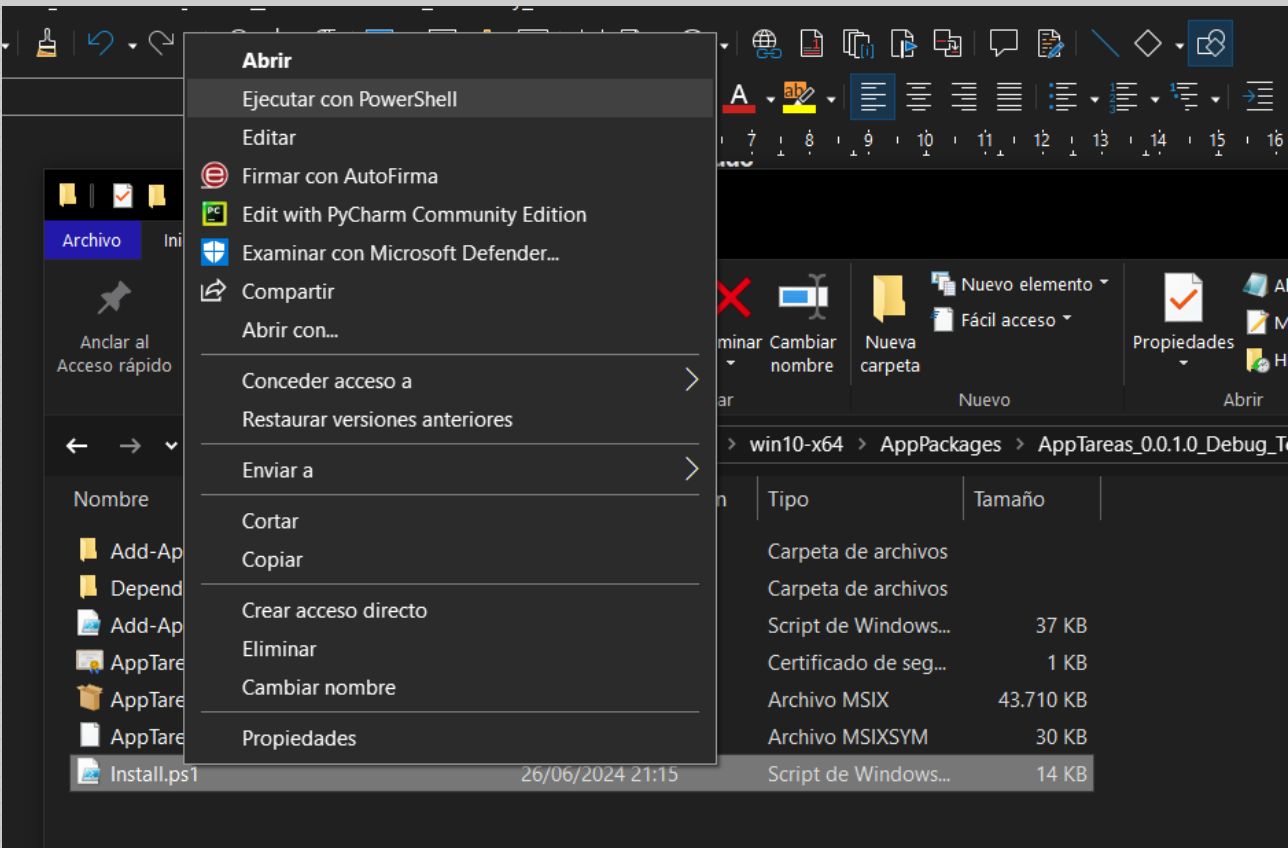
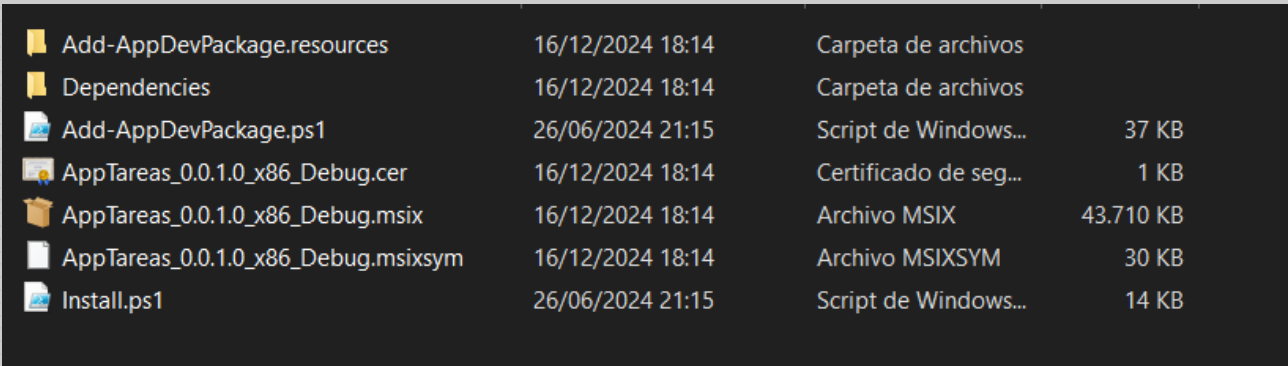
Perfil de publicación:
MSIX-win10-x86.pubxml ▼

☐ Incluir archivos de símbolos públicos, si los hay, para habilitar el análisis de bloqueo de la aplicación. [¿Cómo se usan los símbolos de depuración?](#)

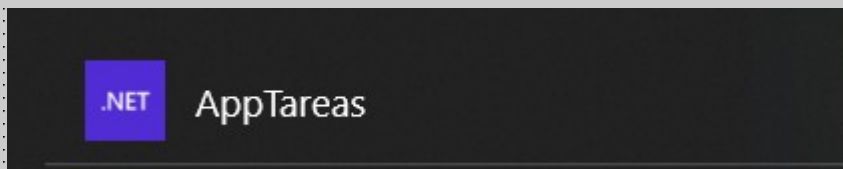
Y una vez le demos a crear, tardará un poco en subir la App, pero una vez termine nos saldrá esta alerta.



Buscamos la app, y le damos al archivo que pone Install , y la abrimos con la powerShell. Y ya se nos habrá descargado la app.



Buscamos la App en el equipo y la ejecutamos.



Aquí se ve la App.

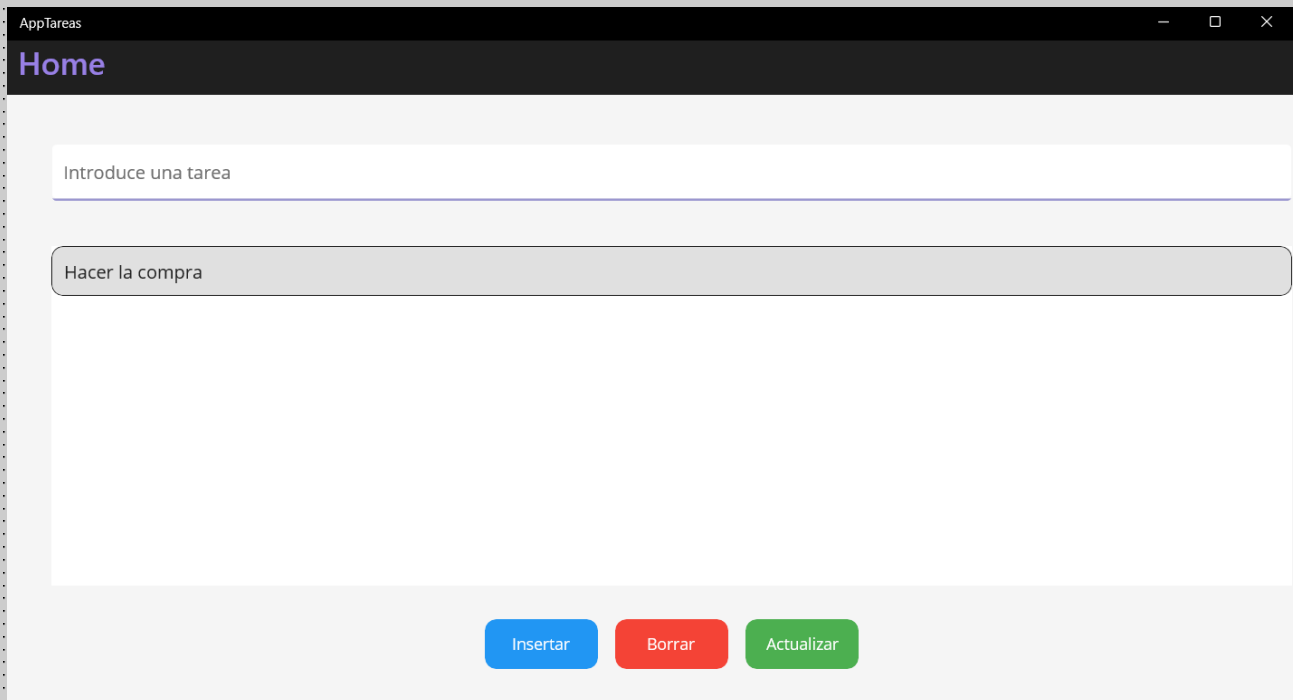


DIAGRAMA DE GANT:



PROBLEMAS:

Los pdf están desactualizado , pero buscando un poco por internet, eres capaz de encontrar las respuestas.

CONCLUSIÓN:

La publicación de aplicaciones desarrolladas con .NET MAUI para Windows mediante Visual Studio ofrece una solución eficiente y estructurada para distribuir software de manera segura y funcional. El uso del formato MSIX, junto con la generación de certificados digitales y scripts automatizados, facilita el despliegue en entornos externos, garantizando una experiencia de usuario profesional y cumpliendo con los estándares de seguridad requeridos.

A través de este proyecto, se ha demostrado la importancia de comprender cada etapa del proceso de publicación, desde la configuración inicial hasta la instalación en otros dispositivos. Además, se ha destacado la flexibilidad del método *sideloading*, que permite distribuir aplicaciones fuera de las plataformas oficiales, adaptándose a las necesidades particulares de desarrolladores y empresas.

En conclusión, dominar estas herramientas y técnicas no solo fortalece las competencias técnicas del desarrollador, sino que también abre posibilidades para explorar formas alternativas de distribución y personalización de las aplicaciones, ampliando su alcance y utilidad en diferentes contextos.

WEBGRAFIA:

<https://learn.microsoft.com/es-es/dotnet/maui/get-started/first-app?view=net-maui-8.0&tabs=vswin&pivots=devices-android>