

# UD 4: Pruebas, documentación y distribución de aplicaciones

*Desarrollo de Interfaces*

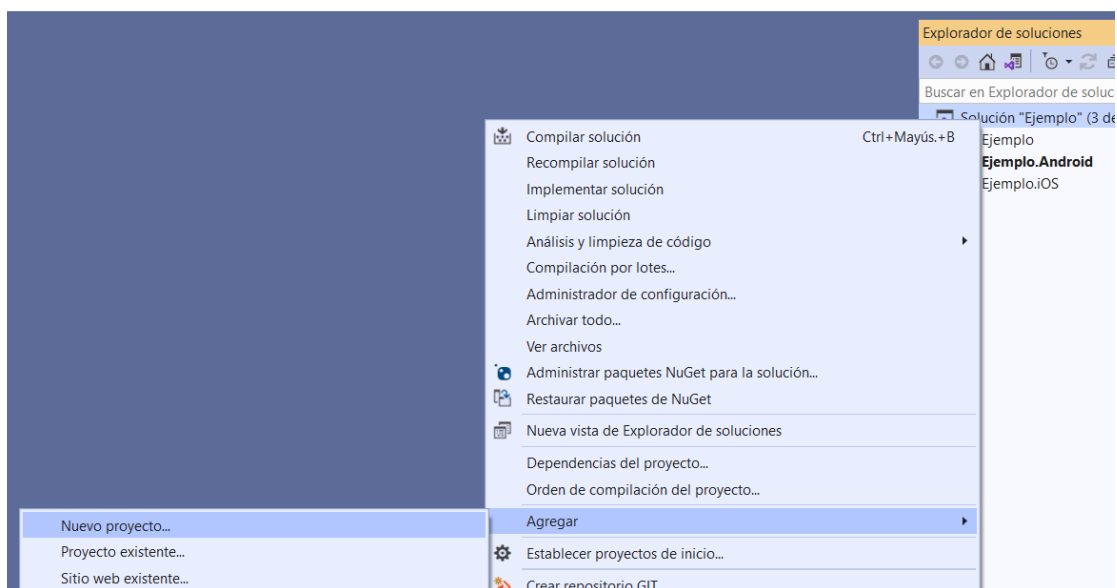
## Índice

1. Pruebas (Xamarin.UITest)	1
2. Documentación de aplicaciones	8
3. Distribución de aplicaciones	10

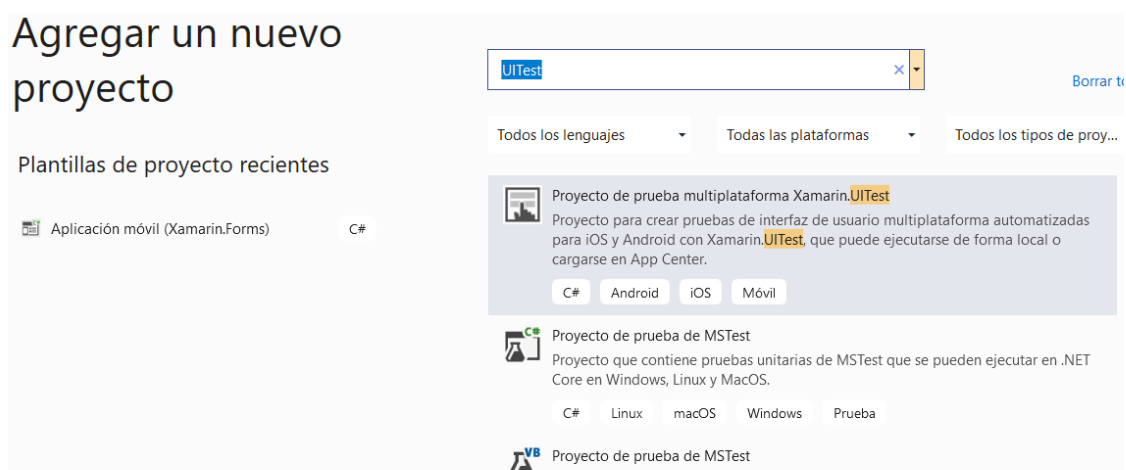
## 1. Pruebas (Xamarin.UITest)

Xamarin.UITest es un marco de pruebas de C# que utiliza NUnit para realizar pruebas de aceptación de la interfaz de usuario en aplicaciones iOS y Android.

Para añadir un proyecto de Xamarin.UITest a nuestra solución de Visual Studio de Xamarin.Forms, haremos click derecho sobre la solución en el explorador de soluciones , Agregar > Nuevo Proyecto.

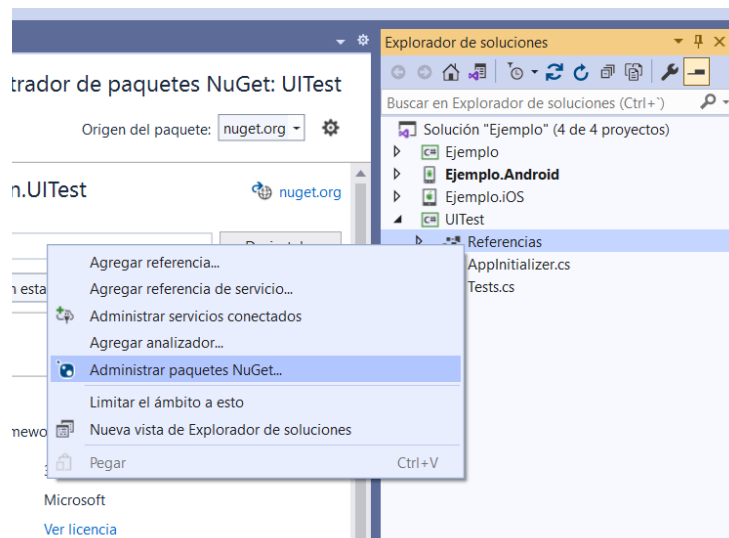


A continuación elegiremos el tipo Xamarin.UITest en la ventana emergente.

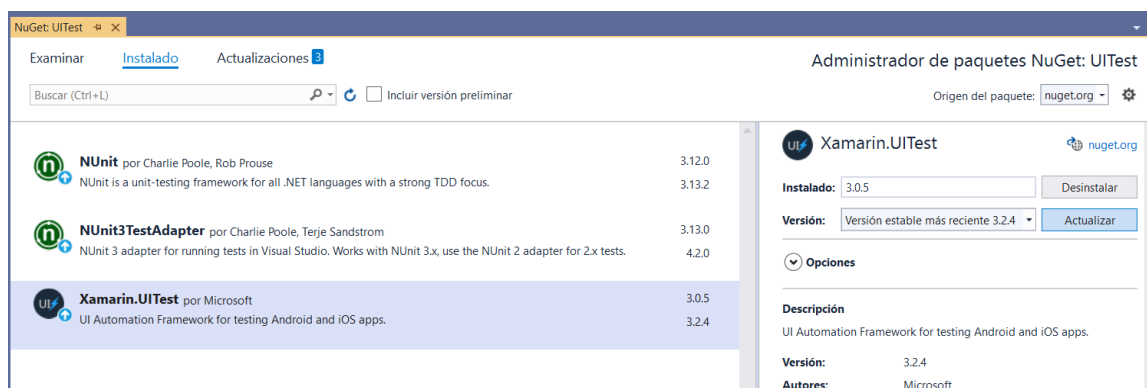


Una vez tengamos añadido el proyecto de pruebas a nuestra solución de visual studio, lo primero que haremos será actualizar a la versión más estable los paquetes NuGet. Para ello

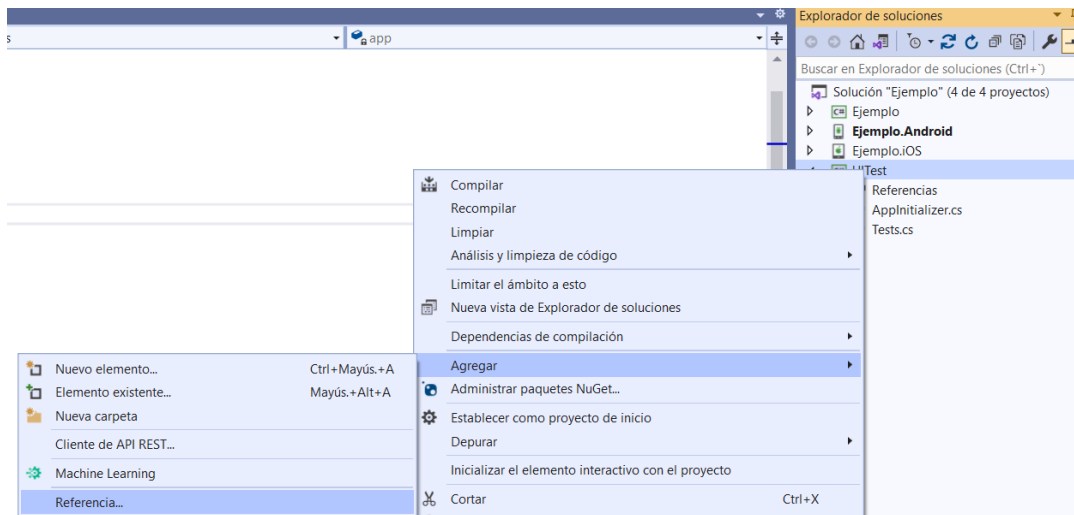
haremos click derecho en el apartado de Referencias del Proyecto de Xamarin.UITest (en el ejemplo se llama “UITest”) y seleccionaremos “Administrar paquetes NuGet”.



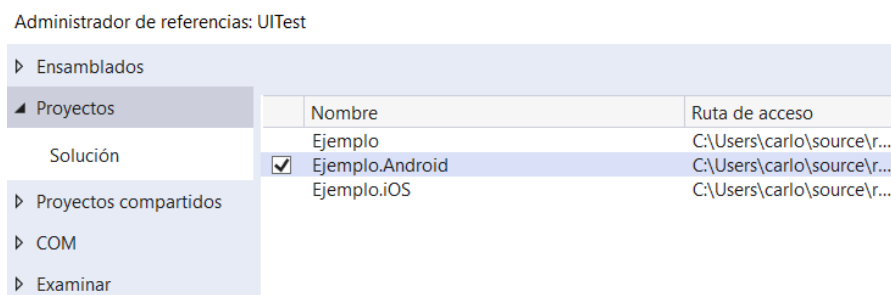
En este punto, se nos abrirá una pestaña dentro del editor de visual studio donde nos aparecerán los tres paquetes que vamos a utilizar. Lo que haremos será seleccionar cada uno de ellos de la parte izquierda y entonces aparecerá la información relativa al paquete seleccionado en la parte derecha. En dicha parte, cambiaremos el valor del desplegable llamado “Versión” y seleccionaremos la “Versión estable más reciente”. Finalmente haremos click en el botón “Actualizar” situado a la derecha de dicho desplegable.



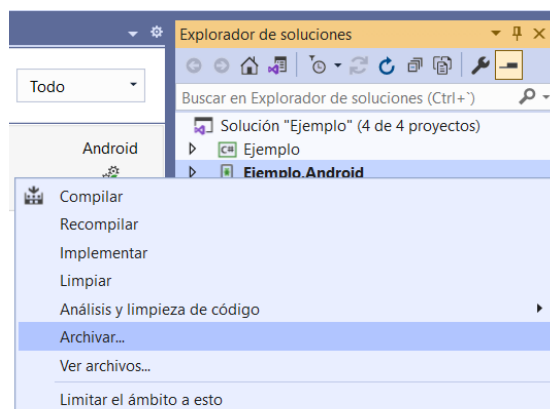
Con todos los paquetes actualizados, simplemente deberemos añadir la referencia al proyecto para poder ejecutar los tests (deberemos añadir en “Referencias” la referencia al proyecto que queramos probar). Haciendo click derecho sobre nuestro proyecto de pruebas, seleccionaremos Agregar > Referencia.



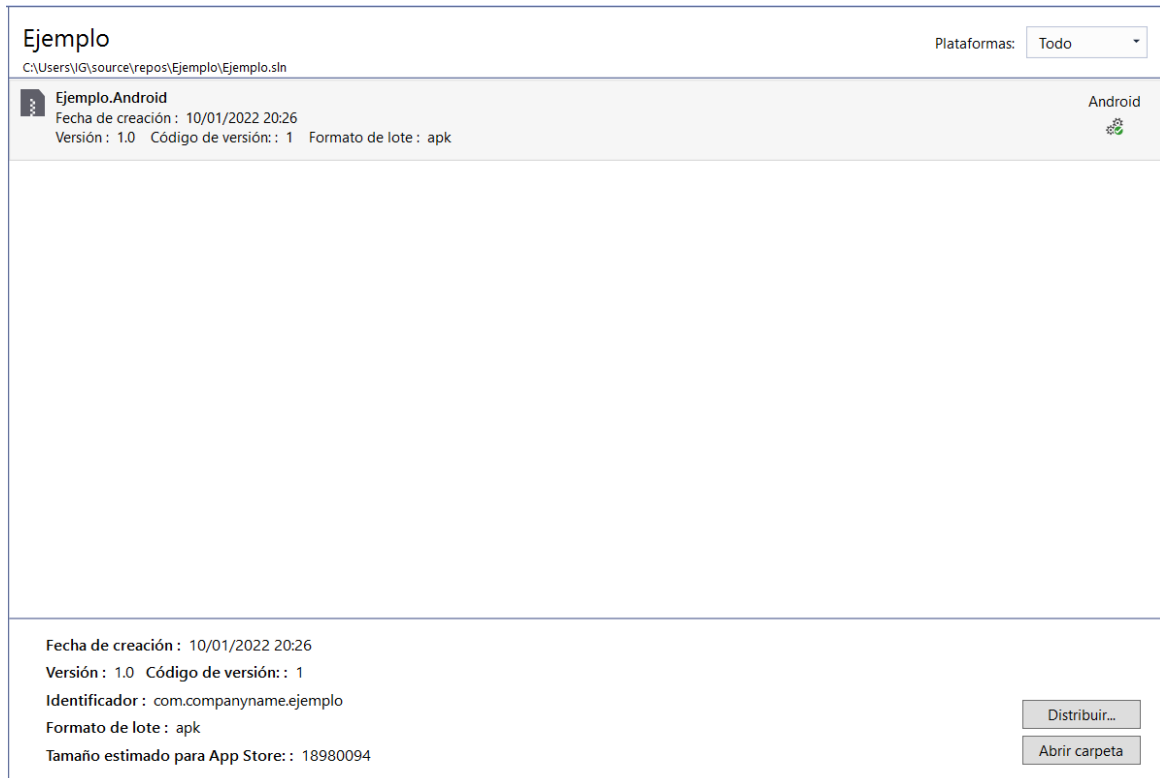
Como todos los proyectos están dentro de la misma solución, nos aparecerán los proyectos que no sean de pruebas. De los otros tres, elegiremos el .Android.



En este punto, ya seríamos capaces de generar nuestros test. Pero para poder ejecutarlos de forma local, primero debemos crear el apk. Para ello, elegiremos el proyecto .Android y haremos click derecho > Archivar.



En ese momento, se nos abrirá una pestaña en el editor de código y se generará el apk. Una vez finalizado el proceso, lo que nos interesa es ir a la parte inferior derecha y hacer click en el botón para ver la carpeta donde se ha generado dicha apk.



Esa ruta es la que debemos utilizar para referenciar a nuestro apk en el fichero “AppInitializer.cs” de nuestro proyecto UITest (este debe ser el aspecto).

```
1 referencia
public class AppInitializer
{
    1 referencia
    public static IApp StartApp(Platform platform)
    {
        if (platform == Platform.Android)
        {
            return ConfigureApp
                .Android
                .EnableLocalScreenshots()
                .ApkFile(@"C:\Users\IG\AppData\Local\Xamarin\Mono for Android\
                .StartApp();
        }

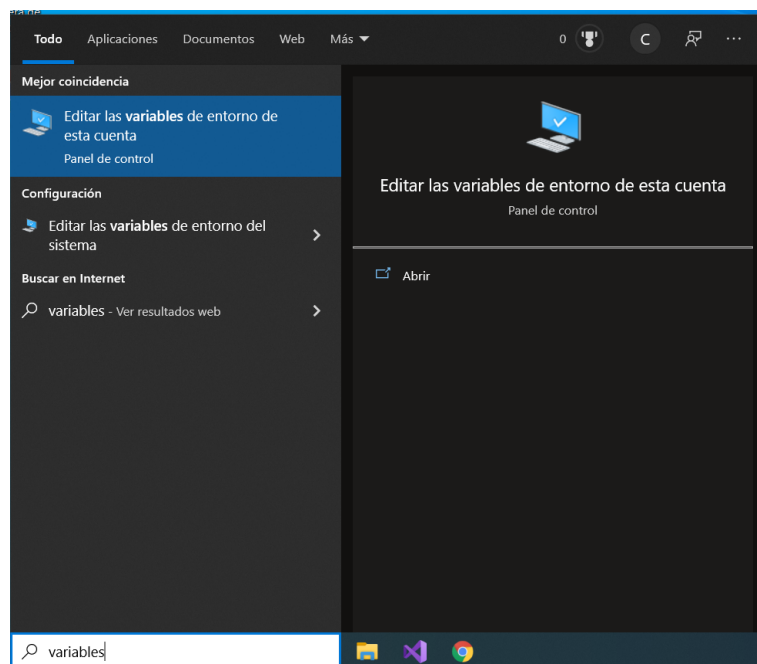
        return ConfigureApp
            .iOS
            .StartApp();
    }
}
```

Para que podáis ver un ejemplo de la ruta, os la facilito aquí (hay que añadir el propio fichero apk al final de la ruta):

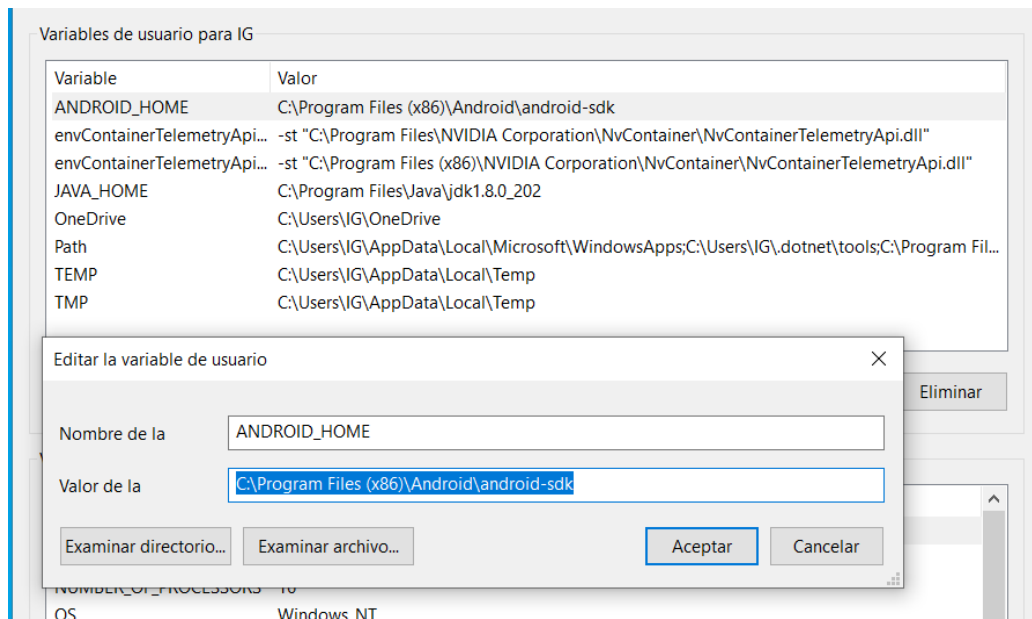
- C:\Users\IG\AppData\Local\Xamarin\Mono for Android\Archives\2022-01-10\Ejemplo.Android 1-10-22 8.26 PM.apkarchive\com.companynome.ejemplo.apk

Así pues, ya podremos escribir nuestros test en el fichero “Test.cs” del proyecto de Xamarin.UITest y ejecutar las pruebas mediante el explorador de pruebas (Ver > Explorador de pruebas).

Para que las pruebas funcionen correctamente, deberemos tener establecidas las variables de entorno de “ANDROID\_HOME” Y “JAVA\_HOME”. La forma más sencilla es buscar en la barra de navegación de Windows la palabra “variables” y seleccionar la opción de “Editar las variables de entorno de esta cuenta”.



Y entonces añadir en el apartado de “Variables de usuario para NOMBRE\_USUARIO” las dos variables mencionadas junto con la ruta donde se encuentran. Os dejo la imagen para que os sirva de guía con las rutas (puede que necesitéis instalar Java).



La forma en la que se generan las pruebas dentro del fichero “Test.cs” es la misma que las pruebas unitarias generadas para C# siguiendo el patrón de las tres A:

- **Organizar (Arrange):** Se inicializan los objetos y se establece el valor de los datos que se pasan al método.
- **Actuar (Act):** Se realizan las acciones a probar.
- **Comprobar (Assert):** Se comprueba si los resultados obtenidos de dichas pruebas son los esperados.

Cada uno de estos métodos generados como prueba unitaria dentro de nuestro proyecto de pruebas tendrá la etiqueta [Test] encima de la definición.

```
[Test]
✓ | 0 referencias
public void LabelText()
{
    Assert.AreEqual("Hello, Xamarin.Forms!", app.Query("MyLabel")[0].Text);
}
```

Para poder referenciar a los elementos presentes en la aplicación que se está probando, cuya referencia dentro de nuestro proyecto de pruebas ya hemos añadido, deberemos utilizar la etiqueta “AutomationId” para darle un identificador que podamos usar en los tests.

```

<StackLayout>
    <Frame BackgroundColor="#2196F3" Padding="24" CornerRadius="0">
        <Label Text="Welcome to Xamarin.Forms!" HorizontalTextAlignment="Center"
    </Frame>
    <Button x:Name="b" AutomationId="MyButton" Text="Click me"/>
    <Label x:Name="l" AutomationId="MyLabel" Text="Hello, Xamarin.Forms!" />
</StackLayout>

```

Como podéis observar, utilizamos el “AutomationId” del Label “MyLabel” como identificador para hacer referencia dentro del “app.Query” y entonces comparamos que el valor de la propiedad “Text” es “Hello, Xamarin.Forms!”.

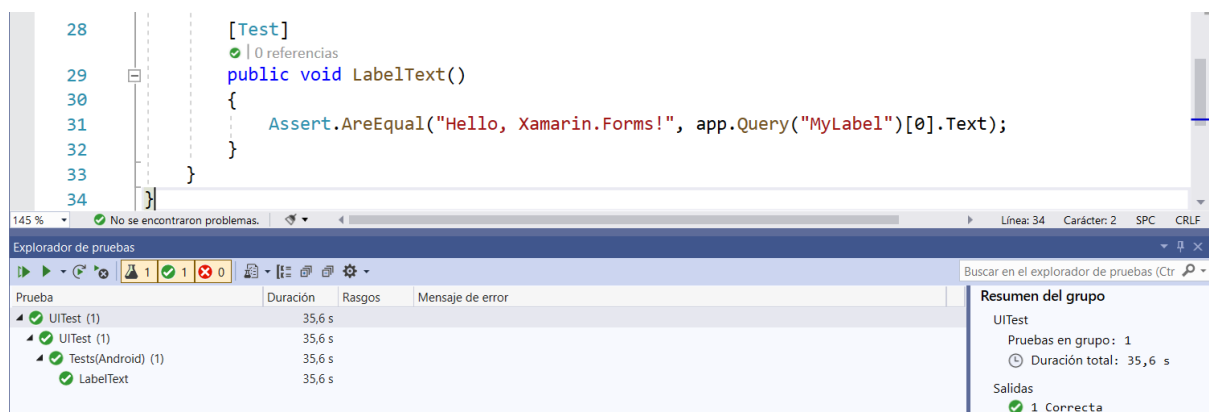
Además comentaremos la línea correspondiente a la ejecución en iOS para evitar problemas y fallos.

```

[TestFixture(Platform.Android)]
//[TestFixture(Platform.iOS)]

```

Con todo esto configurado, podremos ejecutar estos tests mediante el Explorador de pruebas (cuidado con el nombre del APK dado que varía cada día y hora que se cree). También hay que tener el emulador encendido.





## 2. Documentación de aplicaciones

A la hora de generar nuestra aplicación, no basta simplemente con generar el código y el propio programa sino que existen una serie de documentos asociados a este. Estos documentos o manuales asociados a la aplicación son: manual de usuario, guía de referencia, guías rápidas, manuales de instalación, configuración y administración.

El **Manual de Usuario** es el documento técnico que explica el funcionamiento del software o sistema desarrollado como producto resultante del proyecto.

Expone los procesos que el usuario puede realizar con el sistema implantado. Para lograr esto, es necesario que se detallen todas y cada una de las características que tienen los programas y la forma de acceder e introducir información.

Permite a los usuarios conocer en detalle qué actividades deberán desarrollar para la consecución de los objetivos del sistema. Reúne la información, normas y documentación necesaria para que el usuario conozca y utilice adecuadamente la aplicación desarrollada.

Los principales objetivos del manual de usuario son:

- Que el usuario conozca cómo preparar los datos de entrada.
- Que el usuario aprenda a obtener los resultados y los datos de salida.
- Servir como manual de referencia y aprendizaje.
- Definir las funciones que debe realizar el usuario.
- Informar al usuario de la respuesta a cada mensaje de error.
- Definir los diferentes tipos de usuarios
- Definir los módulos en que cada usuario participará.

Los manuales se elaboran en base a las necesidades de los usuarios que van a hacer uso de la aplicación desarrollada. Algunos pueden contener los siguientes apartados:

1. Portada
2. Índice
3. Introducción
4. Instalación del sistema
5. Diagrama general del sistema

6. Diagrama particular detallado
7. Explicación genérica de las fases del sistema
8. Iniciación al uso del sistema
9. Manual de Referencia

El **Manual de Instalación** tiene como objetivo servir de guía en la instalación del sistema. Para ello, en primer lugar, deberá especificar los requerimientos hardware y software necesarios para el correcto funcionamiento del sistema, para posteriormente describir cada uno de los pasos necesarios para la configuración, compilación e instalación del mismo. Además, se deberán incluir las pruebas que se deberán realizar para asegurar que la instalación se ha realizado correctamente, así como el procedimiento de marcha atrás a aplicar en caso de que no haya resultado exitosa la instalación del sistema.

En el caso de las aplicaciones móvil y por el hecho de que estas se distribuyan a través de una plataforma concreta (las tiendas de aplicaciones, generalmente), gran parte de la información presente en estos documentos estará incrustada en la página de nuestra aplicación dentro de dicha tienda y será a través de esta donde se lleve a cabo la instalación y demás aspectos relacionados para su uso. Además, será dentro de la propia aplicación donde esté presente un tutorial interactivo para aprender la operativa y elementos que la conforman.

### **3. Distribución de aplicaciones**

Dentro del desarrollo de aplicaciones móviles y más concretamente en las plataformas donde publicar nuestras aplicaciones, existen dos grandes empresas donde llevar a cabo esta publicación: Google y Apple (aunque existen otras).

Para publicar nuestra aplicación en la Google Play Store, necesitaremos tener una cuenta de Gmail, así como darnos de alta como desarrolladores. Este proceso es muy sencillo y realmente sólo necesitamos registrar nuestra cuenta en Google Play y realizar el pago de la cuota de 25 dólares (pago único para siempre).

Para publicar la propia aplicación dentro de la Play Store, generamos el archivo APK (paquete para el sistema operativo Android similar al formato JAR de JAVA) o el formato promocionado por Google AAB (Android App Bundle). Dicho archivo deberá ser un archivo firmado para que solamente el desarrollador pueda generar actualizaciones del mismo (necesitarás un certificado).

El archivo firmado resultante será el que publiquemos en la Play Store y cuyo proceso de publicación se llevará a cabo a través de la Consola para desarrolladores de Google Play.

En el caso de la Apple App Store, deberemos tener una cuenta de Apple o id de Apple, que se crea de forma gratuita y con cualquier email, tener más de 18 años, realizar un abono anual de 99 dólares y elegir el tipo de desarrollador (individual o empresa). Además esta cuenta debe ser aprobada por Apple y puede tardar un par de días en estar operativa.

Una vez tengamos la cuenta lista para usar, deberemos dar de alta la cuenta en el IDE que utilicemos para desarrollar (por ejemplo Xcode) y posteriormente deberemos empaquetar la aplicación y preparar su subida (las configuraciones que haríamos desde la Consola para desarrolladores de Google Play) pero desde el propio IDE.

#### **Documentos legales**

Cuando desplegamos nuestra aplicación en la correspondiente tienda de aplicaciones (bien sea la Play Store o la App Store entre otras) habrá una serie de elementos que serán necesarios para cumplimentar la sección de nuestra aplicación.

Aparte del resumen y los datos concretos sobre la operativa de nuestra aplicación (así como las imágenes de vista previa), hay que tener listos los documentos sobre la política de privacidad y condiciones del servicio (términos y condiciones).

Estos documentos tienen una base legal importante y para desarrolladores primerizos puede ser un reto rellenarlos de forma correcta. A pesar de ello, tenemos herramientas que nos facilitan tanto obtener información de qué son, para que nos sirven y cómo rellenarlos.

En el siguiente enlace, disponéis de un sitio web con información sobre estos documentos (entre otros) y generadores gratuitos: <https://www.privacypolicies.com/es/>

## **BIBLIOGRAFÍA**

[https://virtual.itca.edu.sv/Mediadores/ads/135\\_manual\\_de\\_usuario.html](https://virtual.itca.edu.sv/Mediadores/ads/135_manual_de_usuario.html)

<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/466>

<https://www.xatakandroid.com/aplicaciones-android/como-crear-publicar-app-google-play>

<https://www.devsdna.com/como-publicar-una-app-en-app-store/>