
+
•
◦

Distribución de aplicaciones

Desarrollo de interfaces

2º DAM

Objetivos

- Empaquetar los componentes que requiere la aplicación.
- Personalizar el asistente de instalación.
- Empaquetar la aplicación para ser instalada de forma típica, completa o personalizada.
- Generar paquetes de instalación.
- Preparar el paquete de desinstalación.
- Preparar la aplicación para ser descargada desde un servidor web y ejecutada



Componentes de una aplicación y empaquetado



Tras implementar una aplicación llega el momento de su distribución, para ello será necesario realizar un proceso de instalación que requiere de un paso previo: el empaquetado



Para realizar un correcto empaquetado se deben agrupar todos los componentes necesarios para el despliegue de una aplicación

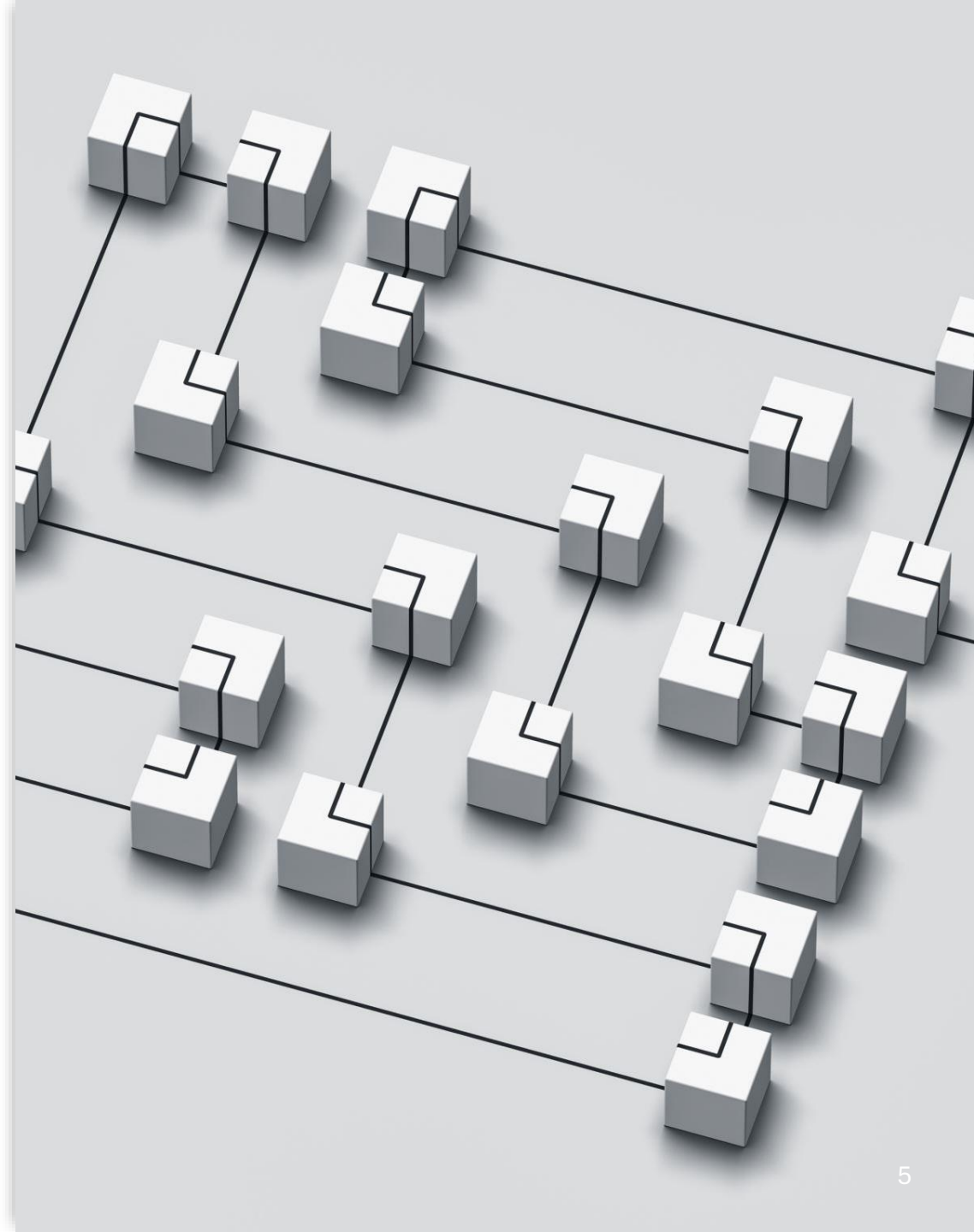
Componentes principales

- Las librerías y bibliotecas
- Los ficheros ejecutables
- Los elementos multimedia




Instaladores y paquetes autoinstalables

- Para desplegar una aplicación software es necesario disponer de instaladores que lleven a cabo cada uno de los pasos necesarios del proceso de instalación.
- Este tipo de programas realizan una serie de operaciones sobre los archivos contenidos en el paquete de distribución que permite la instalación de cualquier software de forma automática.




En la actualidad existen multitud de instaladores, pero entre los más conocidos:

- MSI Studio
- Install Builder
- Windows Installer.



Cada sistema operativo tiene diferentes características y por ello es necesario que todos los instaladores sigan una secuencia de pasos, que consiste en:

1. Comprobar las especificaciones de software y hardware del equipo.
 2. Comprobar la autenticidad del software.
 3. Construir los directorios necesarios para el despliegue de la aplicación.
 4. Extraer los ficheros del paquete de distribución.
 5. Compilar las librerías necesarias.
 6. Definir las variables de entorno.
- 



Windows

Normalmente en el caso de Windows los más utilizados son: EXE o MSI.

- EXE
 - ✓ Este tipo de instaladores es uno de los más comunes entre los usuarios y se trata de un archivo binario ejecutable.
- MSI
 - ✓ A diferencia del anterior, los paquetes MSI permiten llevar a cabo una instalación de forma predefinida. Utilizando este tipo de instaladores al usuario final se le mostrará la opción de siguiente y paso a paso será capaz de instalar la aplicación de manera preconfigurada. El desarrollador podrá seleccionar algunos aspectos de la interfaz de instalación que se le mostrará al usuario, como el texto o el icono. En el caso de los EXE serán completamente personalizables.

[Recurso: instaladores en Windows y diferencias](#)


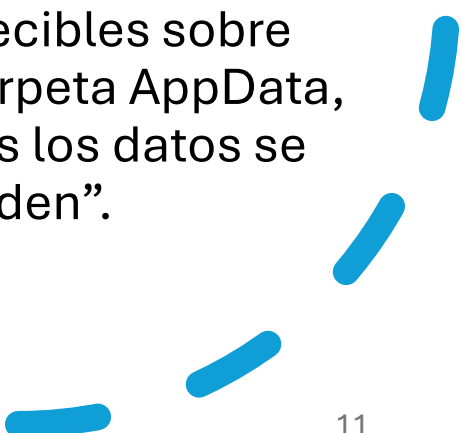
AppX

- ✓ Los instaladores AppX son usados para aplicaciones universales de Windows y cuentan con algunas características de los MSI. Son instaladores muy simples y directos, ofrecen pocas opciones al usuario. Pero cuentan con un proceso de actualización más fácil de versiones antiguas de software a versiones recientes. También permiten realizar una desinstalación más limpia y eficiente.
- ✓ Los instaladores AppX confían en la tecnología de contenedores, de manera que están aislado del resto de Windows por seguridad. Los paquetes AppX solo pueden ser usados en Windows 10, es decir, en un inicio las versiones anteriores de Windows son incompatibles.



MSIX

- ✓ Un archivo con extensión .msix es un instalador de aplicaciones de Windows. En otras palabras, es una alternativa a los archivos .exe, .msi y .appx presentado por Microsoft.
- ✓ Desde la perspectiva del usuario, un MSIX se instala como un archivo MSI, pero por dentro funciona como un archivo AppX. Sin embargo, es importante confiar en la fuente y verificarla muy bien antes de ejecutar este tipo de archivos.

- 
- ✓ Basado en la tecnología de contenedores este formato queda aislado del resto del sistema y permite realizar desinstalaciones y actualizaciones sin problemas.
 - ✓ Cuando instalas un programa usando MSI o EXE, el programa puede hacer cambios en el registro y crear archivos y carpetas en todo su sistema. Cuando los desinstalas, a menudo hay archivos y claves de registro que permanecen, dejando “desorden” en el sistema.
 - ✓ Con MSIX, los programas se instalan en un contenedor y todos los archivos necesarios permanecen dentro de ese contenedor o siguen reglas precisas y predecibles sobre dónde pueden estar esos archivos (en la carpeta AppData, por ejemplo). Cuando los desinstalas, todos los datos se van con el programa, no deja ningún “desorden”.
- 


- ✓ Otro punto importante es que es posible la distribución de MSIX fuera de la Microsoft Store. Con ello es mucho más sencillo coger programas antiguos y volver a empaquetarlos bajo MSIX.
- ✓ Microsoft lanzó un SDK para mejorar la compatibilidad multiplataforma. Como vemos en su página de GitHub, el soporte es posible para iOS, MacOS, Android, Linux e incluso versiones más antiguas de Windows. El proceso es simple. Los desarrolladores colocan instrucciones especiales en los archivos MSIX para permitirle identificar el sistema operativo y los pasos a seguir.
- ✓ La herramienta de creación de paquetes MSIX Packaging Tool, está disponible para Windows 10 versión 1809 o superiores incluyendo versiones Insider, mientras que el proyecto y código fuente está alojado en GitHub.

Linux

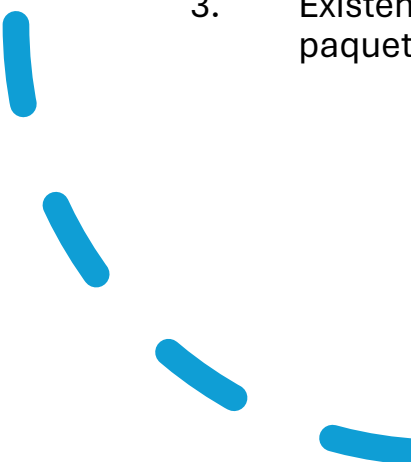
- Este sistema operativo libre permite empaquetar y distribuir aplicaciones con diferentes formatos.
- En Linux se utilizan algunos tipos de paquetes que requieren de operaciones específicas a través de línea de comandos para su creación.
- Entre los más destacables se encuentran:
 - ✓ DEB: usado en aquellas distribuciones que están basadas en Debian, como Ubuntu o Kubuntu. Una de las ventajas de este tipo de paquetes, a diferencia de TAR o TGZ, es que los de tipo deb pueden ser instalados directamente, mientras que los otros han de ser extraídos en primer lugar.
 - ✓ RPM: Redhat Package Manager.
 - ✓ TGZ: específico de UNIX. Se trata de paquetes TAR con compresión a través de GUNZIP.
 - ✓ TAR: paquetes sin compresión.






Interacción con el usuario

- ✓ A la hora de diseñar asistentes de instalación es necesario conocer el conjunto de pautas que debe seguir el desarrollador.
- ✓ Lo más importante es comenzar con un análisis del tipo de interacción que se va a dar entre la aplicación y el usuario antes de comenzar con su diseño.



Después, se tendrá en cuenta el tipo de menús y diálogos que contendrá el asistente de instalación para la configuración de la misma siguiendo el siguiente criterio:

1. Al inicio, si la aplicación se ha desarrollado para varios idiomas, se muestra al usuario un menú para llevar a cabo la elección del idioma deseado.
 2. En la actualidad es cada vez más común que para continuar con el proceso de instalación el asistente muestre la licencia de uso de la aplicación que el usuario ha de aceptar.
 3. Existen aplicaciones que permiten al usuario seleccionar todas o solo algunas de las herramientas contenidas en el paquete. Se modela, por tanto, un menú que permite la selección de las mismas.
- 

- 
- 
- 
- 
- 
4. Después, se selecciona la ruta en la que se van a situar los archivos de la aplicación. Habitualmente, en función del sistema operativo, se utiliza una ruta por defecto, pero el usuario debe poder escoger una nueva.
 5. Durante todo el proceso de instalación se muestra algún tipo de indicador del porcentaje completado sobre el total.
 6. Por último, cuando el proceso concluye se le ha de notificar al usuario. En función del tipo de herramienta, puede ser necesario reiniciar el sistema operativo, en este caso se ha de preguntar al usuario si desea hacerlo en ese momento o más tarde. Del mismo modo, también suele ser habitual habilitar una opción para iniciar la ejecución de la aplicación tras finalizar el proceso de instalación.

Ficheros firmados digitalmente

- ✓ Hoy en día es necesaria la firma en la distribución de software dado que en muchas ocasiones la descarga de nuevas aplicaciones se realiza a través de Internet. Por lo que será necesario utilizar mecanismos que garanticen la autenticidad del software.
- ✓ Se debe conocer que las firmas digitales están formadas por una clave pública y una privada.
- ✓ La clave pública representa la identidad de la entidad que posee la clave privada asociada a esa clave pública.
- ✓ La clave privada está pensada para que siempre esté bajo el control del firmante.

Herramientas para firmar digitalmente

- ✓ Existen ciertas herramientas específicas que permiten el firmado digital de ficheros y que son las más aconsejadas, por ejemplo, AutoFirma utilizado para ficheros PDF, Ksi Secure o XolidoSign que permiten la firma de cualquier tipo de archivo.



Actividad

Descarga la aplicación [XolidoSign](#) y firma un documento de tu propiedad.

SignTool. Firma Setup.exe y MySetup.msi

- ✓ Antes de colocar Setup.exe y MySetup.msi en el servidor web, debe firmar los archivos con el certificado digital y la clave privada, Mycert.cer y Mycert.pvk, mediante la utilidad SignTool.
- ✓ La herramienta [Firmar \(SignTool\)](#) es una herramienta de la línea de comandos que firma archivos digitalmente, comprueba firmas en archivos o archivos con marcas de tiempo.
- ✓ Esta herramienta se instala automáticamente con Visual Studio.
- ✓ [Ejemplo de instalación](#)
- ✓ [Ejemplo de actualización](#)




Instalación de aplicaciones desde un servidor web

- ✓ Hasta hace unos años, la distribución de aplicaciones software se realizaba mediante CD-ROM o archivos comprimidos almacenados en USB.
- ✓ Pero hoy en día gracias a los hostings o servidores web, los paquetes software pueden quedar alojados en estos servidores a los que se podrán acceder en cualquier momento para realizar la descarga a través de un conjunto de hipervínculos.
- ✓ [Distribución de una app de Windows 10 desde un servidor IIS](#)
- ✓ [Distribución de una app de Windows 10 desde una aplicación web de Azure](#)
- ✓ [Distribución de una app de Windows 10 desde un servicio web de AWS](#)



WIX Toolset

- ✓ El conjunto de herramientas de WiX permite a los desarrolladores crear instaladores para Windows Installer, el motor de instalación de Windows. Es de código abierto y parte de la Fundación .NET.
- ✓ El núcleo de WiX es un conjunto de herramientas de compilación que crean paquetes de Windows Installer usando los mismos conceptos de compilación que el resto de su producto: el código fuente se compila y luego se vincula para crear ejecutables; en este caso, los paquetes de instalación .exe, los paquetes de instalación .msi, los módulos de fusión .msm y los parches .msp. Las herramientas de construcción de línea de comandos de WiX funcionan con cualquier sistema de construcción automatizado. Además, MSBuild es compatible desde la línea de comandos, Visual Studio y Team Build.
- ✓ WiX incluye varias extensiones que ofrecen funcionalidad más allá de la de Windows Installer. Por ejemplo, WiX puede instalar sitios web de IIS, crear bases de datos de SQL Server y registrar excepciones en el Firewall de Windows, entre otros.

- 
- ✓ El WiX SDK incluye bibliotecas administradas y nativas que facilitan la escritura de código que funciona con Windows Installer, incluidas acciones personalizadas en C # y C ++.

<https://medium.com/el-acordeon-del-programador/tagged/wix-toolset?p=2eb1ebc186c8>