

Ciclo Formativo DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Módulo 7

Desarrollo de interfaces

Unidad Formativa 2

Quedan rigurosamente prohibidas, sin la autorización escrita de los titulares de «Copyright», bajo las sanciones establecidas en las leyes, la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, y la distribución de ejemplares de ella mediante alquiler o préstamo públicos. Dirijase a CEDRO (Centro Español de Derechos Reprográficos, <http://www.cedro.org>) si necesita fotocopiar o escanear algún fragmento de esta obra.

INICIATIVA Y COORDINACIÓN

IFP Innovación en Formación Profesional

Supervisión editorial y metodológica:

Departamento de Producto de Planeta Formación

Supervisión técnica y pedagógica:

Departamento de Enseñanza de **IFP** Innovación en Formación Profesional

Módulo: Desarrollo de interfaces

UF 2 / Desarrollo de Aplicaciones Multiplataforma

© Planeta DeAgostini Formación, S.L.U.

Barcelona (España), 2017

MÓDULO 7

Unidad Formativa 2

Desarrollo de interfaces

Esquema de contenido

1. FICHEROS DE AYUDA. FORMATOS

- 1.1. PDF (O CUALQUIER FORMATO OFIMÁTICO)**
- 1.2. WINHELP (.HLP)**
- 1.3. CHM**
- 1.4. MICROSOFT ASISTENCIA MARKUP LANGUAGE (MAML)**
- 1.5. FICHEROS DE AYUDA EN LINUX**

2. HERRAMIENTAS DE GENERACIÓN DE AYUDAS

- 2.1. HELPMAKER**
- 2.2. MICROSOFT HTML HELP WORKSHOP**
- 2.3. DOCBUILDER**
- 2.4. WINHELP COMPILER**

3. AYUDA GENÉRICA Y SENSIBLE AL CONTEXTO

4. TABLAS DE CONTENIDOS, ÍNDICES, SISTEMAS DE BÚSQUEDA, ENTRE OTROS

- 4.1. ARCHIVOS HHC (CONTENIDO)**
- 4.2. ARCHIVOS HHK (ÍNDICE)**
- 4.3. ARCHIVOS H (DECLARACIÓN DE ID DE CONTEXTO)**
- 4.4. ARCHIVOS HTM (CUERPOS DE LA AYUDA)**

5. TIPOS DE MANUALES: MANUAL DE USUARIO, GUÍA DE REFERENCIA, GUÍAS RÁPIDAS, MANUALES DE INSTALACIÓN, CONFIGURACIÓN Y ADMINISTRACIÓN, DESTINATARIOS Y ESTRUCTURA

- 5.1. TIPOS DE DOCUMENTACIÓN**
- 5.2. LA GUÍA TÉCNICA**
- 5.3. LA GUÍA DE USUARIO**
- 5.4. LA GUÍA DE INSTALACIÓN**

6. CONFECCIÓN DE TUTORIALES MULTIMEDIA

- 6.1. ¿ENTIENDE LAS NECESIDADES DE SU TARGET?**

- 6.2. SEA BREVE, LA ATENCIÓN DISMINUYE RÁPIDO
- 6.3. ESPIAR A LA COMPETENCIA
- 6.4. PREPÁRESE ANTES DE APRETAR EL BOTÓN DE GRABAR
- 6.5. CONCLUSIÓN
- 7. **HIERRAMIENTAS DE CAPTURA DE PANTALLAS Y SECUENCIAS DE ACCIONES**
 - 7.1. ZSCREEN
 - 7.2. GREENSHOT
 - 7.3. LIGHTSHOT
 - 7.4. PRSRC
 - 7.5. SCREENSHOT CAPTOR
 - 7.6. JET SCREENSHOT
 - 7.7. SCREENPRESSO
- 8. **HERRAMIENTAS PARA LA CONFECCIÓN DE TUTORIALES INTERACTIVOS, SIMULACIÓN**
 - 8.1. VERSIÓN ESCRITORIO
 - 8.2. VERSIONES WEB
- 9. **COMPONENTES DE UNA APLICACIÓN. EMPAQUETADO**
 - 9.1. LINUX
 - 9.2. WINDOWS
 - 9.3. JAVA
- 10. **INSTALADORES. PAQUETES DE INSTALACIÓN. HERRAMIENTAS PARA CREAR PAQUETES DE INSTALACIÓN. PARÁMETROS DE LA INSTALACIÓN.**
 - 10.1. WINDOWS
 - 10.2. LINUX
 - 10.3. PAQUETES .DEB Y .RPM
- 11. **ASISTENTES DE INSTALACIÓN Y DESINSTALACIÓN. PAQUETES AUTOINSTALABLES**
 - 11.1. WINDOWS
 - 11.2. LINUX
- 12. **FICHEROS FIRMADOS DIGITALMENTE**
 - 12.1. ESTRUCTURA DE LA FIRMA

12.2. ¿DÓNDE SE GUARDA EL DOCUMENTO ORIGINAL?

12.3. FIRMAS CON MÚLTIPLES USUARIOS

12.4. FIRMAS LONGEVAS Y SELLO DE TIEMPO

12.6. SELLO DE TIEMPO

12.7. RESELLADO

**12.8. HERRAMINETAS GRATUITAS PARA LA FIRMA ELECTRÓNICA
BASADAS EN CERTIFICADO DE DOCUMENTOS**

**13. DESCARGA, INSTALACIÓN Y EJECUCIÓN DE APLICACIONES
UBICADAS EN SERVIDORES WEB.**

13.1. SISTEMAS BASADOS EN PAQUETES BINARIOS

13.2. SISTEMAS DE METAPAQUETES

13.3. SISTEMAS PRIORITARIOS

13.4. GESTIÓN DE PAQUETES INCRUSTADA EN APLICACIONES

INTRODUCCIÓN

Un sistema pobremente documentado carece de valor aunque haya funcionado bien en alguna ocasión. En el caso de programas pequeños y poco importantes que solo se utilizan durante un corto período de tiempo, unos cuantos comentarios en el código podrían ser suficientes. No obstante, la mayoría de los programas cuya única documentación es el código, se quedan obsoletos rápidamente y es imposible mantenerlos. El que la dedicación de un poco de esfuerzo a la documentación sea recompensada incluso dentro de los límites de un pequeño proyecto, constituye una sorpresa para la mayoría de los novatos.

A menos que usted o su equipo sean infalibles y vivan en un mundo en el que nada cambia, tendrán que volver a consultar el código que ya está escrito, y pondrán en duda decisiones que tomaron durante el desarrollo del mismo. Si no documentaron sus decisiones, se verán siempre cometiendo los mismos errores y tratando de comprender lo que pudieron haber descrito fácilmente en una ocasión. La falta de documentación no sólo genera trabajo adicional, sino que también tiende a dañar la calidad del código. Si no posee una nítida caracterización del problema, es imposible que desarrolle una solución clara.

Aprender a documentar software es una tarea complicada y exige un criterio de ingeniería maduro. Documentar de forma concisa es un error habitual, pero el otro extremo puede resultar igual de perjudicial: si escribe documentaciones extensas, éstas atosigarán al lector y constituirán una carga a la hora de conservarlas. Es esencial documentar solo los asuntos correctos. La documentación no sirve de ayuda para nadie si su extensión desanima a la gente a la hora de leerla.

Los principiantes tienen la tentación de centrar sus esfuerzos en temas sencillos, ya que estos les resultan más fáciles de documentar. Esto es una pérdida de tiempo; no se aprende nada del esfuerzo y se termina escribiendo una documentación que es cualquier cosa excepto útil. Los principiantes también tienden a mostrarse reacios con los problemas de documentación. Esto trae consigo poca visión de futuro: si usted sabe que algún aspecto de su diseño no es del todo correcto, que alguna parte del problema no se ha aclarado o que es posible que parte del código tenga errores, dígalos. Hará que el lector ahorre tiempo dándole vueltas a algo que aparentemente es erróneo, se acordará de dónde tiene que mirar si encuentra problemas y acabará teniendo una documentación más útil y honesta.

Otro asunto es cuándo documentar. Aunque algunas veces es conveniente posponer la tarea de la documentación mientras se realizan experimentos, los programadores con experiencia suelen documentar de forma metódica incluso el código provisional, los análisis de un problema inicial y los borradores de un diseño. Ellos creen que esto hace que la experimentación sea más productiva. Además, dado que han tomado la documentación como hábito, les resulta normal documentar a medida que van avanzando.

1. FICHEROS DE AYUDA. FORMATOS

Un fichero de ayuda, como su nombre indica, es un documento sobre papel o digital cuya finalidad es servir de guía de referencia, manual o, como su nombre indica, ayuda a los diferentes usuarios de una aplicación.

Es importante tener en cuenta que cuando nos referimos a futuros usuarios, también incluimos a posibles desarrolladores que puedan en un futuro modificar dicha aplicación, con lo cual deberán en principio conocer su uso.

Los formatos de ayuda más destacados, centrándonos en soporte digital, pueden ser:

1.1. PDF (O CUALQUIER FORMATO OFIMÁTICO)

Son los más habituales y fáciles de realizar. Pueden ser abiertos por cualquier visor pdf en el primer caso o bien herramientas ofimáticas de carácter general en el segundo caso.

1.2. WINHELP (.HLP)

Los archivos con extensión hlp son los archivos de ayuda de Windows (el nombre de la extensión proviene de Help) u otros programas. Al hacer doble clic sobre un archivo con extensión hlp este se abrirá con la Ayuda de Windows gracias al programa winhlp32.exe.

El formato de archivo se basa en el formato de texto enriquecido (RTF). Fue la plataforma de ayuda más popular desde Windows 3.0 a Windows XP. Posteriormente, WinHelp fue eliminado en Windows Vista para fomentar el uso de formatos de ayuda más recientes como chm.

Un archivo de WinHelp tiene como ".hlp" sufijo. Puede ir acompañado por una tabla de contenido opcional (.cnt). Cuando Windows abre un archivo WinHelp, se crea un archivo gid en el mismo directorio, que contiene información sobre el archivo hlp entre la cual está el tamaño de la ventana y la ubicación en la pantalla. Si el usuario hace clic en la ficha.

"Buscar" permite la indexación de palabras clave.

1.3. CHM

Archivo de Ayuda de HTML Compilado (Microsoft Compiled HTML Help en inglés) es un formato privativo de ayuda en línea desarrollado por Microsoft. Se publicó por primera vez en 1997 como sucesor del sistema de ayuda

Winhelp. Se popularizó con Windows 98 pero, sobre todo, se usó considerablemente hasta el sistema operativo Windows XP.

El archivo chm consiste en un índice, una tabla de contenidos y un conjunto de páginas en HTML hiperenlazadas a la tabla, que se compilan para generar el archivo de ayuda.

Aplicaciones como HTML Help Workshop, de Microsoft, permiten compilar estos archivos. En 2003, Microsoft anunció que debido a fallos de seguridad que presentaba, no lo iba a usar a partir de Windows Vista en adelante; sin embargo, aún aparece en muchas aplicaciones que corren en Windows7.

El archivo chm consiste en un índice, una tabla de contenidos y un conjunto de páginas en HTML hiperenlazadas a la tabla, que se compilan para generar el archivo de ayuda. Aplicaciones como HTML Help Workshop, de Microsoft, permiten compilar estos archivos. Los Archivos de Ayuda de HTML Compilado pueden contener páginas web con código malicioso y ejecutarlas posteriormente, por lo que representan una amenaza a la seguridad. El formato de archivo de Microsoft Reader, lit, es una derivación del chm.

Los archivos chm a veces se utilizan como e-books.

1.4. MICROSOFT ASISTENCIA MARKUP LANGUAGE (MAML)

Es un lenguaje de marcado basado en XML desarrollado para proporcionar asistencia al usuario ("ayuda en línea") para el sistema operativo Microsoft Windows Vista y sucesivos. MAML también se utiliza para proporcionar información de ayuda para los cmdlets de PowerShell, módulos y funciones avanzadas. Fue sustituto para los archivos de ayuda de Windows desde la versión Vista.

MAML se aparta de todos los tipos anteriores de asistencia a los usuarios de los sistemas operativos de Windows. Anteriormente, la asistencia al usuario para los sistemas operativos Windows utiliza los archivos creados con una línea de comandos de compilador (hhc.exe).

Este compilador se utiliza para compilar una .hhp (proyecto) de archivos, hhc (tabla de contenido) de archivos, hhk archivo (índice) y una colección de archivos HTML y los temas relacionados con los recursos (CSS, JavaScript y archivos de imagen) en uno solo archivo chm.

El aspecto más significativo de MAML es que cambia la producción de la ayuda al usuario para el concepto de autoría estructurada (algo similar a DITA o DocBook). Con MAML, se hace hincapié en el contenido y las tareas que un usuario realiza con un ordenador, no las características del software. La presentación se gestiona como parte del motor de renderizado cuando un usuario solicita un tema (como un motor de búsqueda).

La estructura de autoría MAML se divide en segmentos relacionados con un tipo de contenido:

- Preguntas frecuentes conceptual.
- Glosario.
- Procedimiento de referencia.
- Contenido reutilizable.
- Tarea y solución de problemas.
- Tutorial.

En la actualidad Microsoft ha lanzado herramientas automatizadas para ayudar en la creación de MAML.

Hay tres niveles de transformación se producen cuando un tema se muestra: estructura, presentación y representación:

- La estructura contiene contenido reutilizable y aplica la lógica condicional para determinar la estructura que debe tener el contenido cuando se muestra, y el contenido del texto en sí.

- La presentación permite que el contenido creado en MAML utilizar muchos formatos diferentes, incluyendo DHTML, XAML, RTF y material impreso.
- La representación se aplica hojas de estilo y visualiza el contenido final a los usuarios.

Desafortunadamente aún no existe una herramienta que permita la automatización total de la generación de las ayudas con MAML.

1.5. FICHEROS DE AYUDA EN LINUX

En cuanto a Linux los primeros ficheros de ayuda fueron los generados con el comando “man” para conocer el uso y opciones de los comandos.

Por otro lado, están los comandos “whatis” que permite una breve descripción de los comandos y “apropos” que nos permite conocer los comandos relacionados con un determinado tema.

Por ejemplo, si escribimos “apropos password” nos mostrará en línea una serie de comandos relacionados con el término “password”. Además la opción comando “-help” también nos permite acceder a las características de dicho comando.

Finalmente, podemos navegar en nuestro sistema de archivos hasta “/usr/share/doc” y buscar más información como, changelogs, readmes específicos de distribución, archivos de ejemplo, etc., de la aplicación y/o utilidad que estemos estudiando. Es en esta sección donde la mayoría de los programas utilizados en Linux guardan sus documentos de ayuda.

2. HERRAMIENTAS DE GENERACIÓN DE AYUDAS

Si las ayudas están generadas en formatos pdf u otro formato ofimático se utilizarían los editores de texto o documentos habituales.

En el caso de que los formatos sean otros entonces tenemos:

2.1. HELPMAKER

Es una aplicación para crear archivos de ayuda para programas. Permite la creación de archivos de ayuda enteros; en diferentes formatos, tales como: WinHelp, RTF (texto enriquecido) y HTML-Help. El programa permite crear la estructura de la ayuda, pudiendo cambiarla, ampliarla, editarla, añadir vínculos (incluso de una zona a otra de la misma ayuda), etc. También cuenta con un corrector ortográfico, que te vendrá muy bien si haces las ayudas también en inglés.

2.2. MICROSOFT HTML HELP WORKSHOP

Permite crear ficheros de ayuda de windows (HLP) y páginas web que utilicen controles de navegación. Crea estos ficheros y los distribuye con las aplicaciones. Incluye un administrador de proyectos, un compilador de ayuda y un editor de imágenes. HTML Help Workshop ofrece algunas ventajas sobre el estándar HTML, incluyendo la habilidad de implementar una tabla de elementos combinada y un índice, así como el uso de palabras clave para capacidades avanzadas de hiperenlazado.

El compilador permite comprimir HTML, gráficos y otros ficheros en un fichero compilado chm relativamente pequeño, que puede ser distribuido junto a la aplicación, o bien descargado desde Internet.

También se incluyen un control ActiveX y un applet de Java. El primero puede ser usado en cualquier navegador con soporte ActiveX, o en HTML Help Workshop, y el applet de Java puede ser utilizado en cualquier navegador que soporte Java. Con ambos controles puedes crear páginas web con soporte de ayuda HTML, incluyendo tablas de contenido, índices, y temas relacionados. Así mismo, crea ficheros compilados de ayuda para ser ejecutados directamente desde el web.

2.3. DOCBUILDER

Es una aplicación capaz de generar archivos de documentación y ayuda en diversos formatos para el software desarrollado por usted mismo. El programa reconoce varios lenguajes de programación (C/C++, Pascal, Delphi) y puede generar documentación en RTF, HTML y archivos Windows de ayuda.

Esta herramienta examina la estructura del código fuente del programa, distinguiendo los comandos de los comentarios sobre el software, analizando estos últimos y distribuyéndolos de acuerdo con el formato elegido. Es de las más antiguas.

2.4. WINHELP COMPILER

Ayuda a crear archivos de ayuda. Este compilador de Microsoft crea ficheros de ayuda Windows hlp que solo se pueden ver en Windows 95/98/NT4 y superiores. WinHelp Compiler. Este paquete incluye "Segmented Hypergraphics Editor" (SHED.EXE) y "Multi-Resolution Bitmap Compiler" (MRBC.EXE).

En la red podrá encontrar algunos más, pero, en general, estos son los más utilizados.

3. AYUDA GENÉRICA Y SENSIBLE AL CONTEXTO

Una ayuda genérica es una ayuda que no trata sobre ningún tema específico, sino sobre la aplicación en sí.

Frente a este concepto de ayuda esta la ayuda sensible al contexto. Los programas están volviéndose más inteligentes sobre cómo ofrecer ayuda que resulte apropiada para las tareas que actualmente usted está haciendo. Las cajas de diálogo tienen un pequeño botón arriba a la derecha con el símbolo de interrogación (?). Si usted pulsa este botón y después pulsa sobre algo en la ventana, conseguirá que surja una explicación. Esto resulta muy útil cuando no entiende las opciones que se presentan en la caja de diálogo.

4. TABLAS DE CONTENIDOS, ÍNDICES, SISTEMAS DE BÚSQUEDA, ENTRE OTROS

Para explicar esto utilizaremos el popular sistema de ayudas basado en archivos chm.

CHM utiliza un lenguaje de marcas HTML que es un lenguaje muy probado y nos ofrece más posibilidades y versatilidad que los derivados en RTF.

Una vez creados los diferentes capítulos de nuestra ayuda en diferentes archivos HTML, estos son aglutinados en un archivo de proyecto con extensión HHP. A este proyecto se le añaden:

Archivos H: añaden un ID para identificador de capítulo. Archivos HHK: índices. Archivos HHC: tablas de contenido.

Con todo esto, compilamos el proyecto y obtenemos un archivo de ayuda con extensión CHM. La estructura básica para crear dicho proyecto de ayuda CHM está basada un archivo de extensión HHP que aglutina todos los demás archivos que conforman la ayuda CHM.

Para crear este y los demás archivos podemos utilizar una aplicación como, por ejemplo, Help WorkShop, o un editor de textos sencillo como el bloc de notas.

Un archivo de proyecto HHP es un archivo de texto estructurado por secciones que se referencian con los caracteres "[y]". Las secciones más importantes son:

- **OPTIONS:** opciones de compilación. En esta sección damos a conocer al compilador de ayudas informaciones tales como:
 - Title: título de nuestra ayuda.
 - Language: ID del idioma utilizado para nuestra ayuda.
 - Indexfile: archivo de índice, (HHK).
 - Contents file: archivo de contenido, (HHC).
 - Compiled file: nombre del archivo de ayuda compilado, (CHM).

Ejemplo

[OPTIONS]

Compatibility=1.1 or later

Compiled file=PruebaHTML.chm

Contents file=PruebaHTML.hhc

Default topic=PruebaHTML1.htm

Display compile progress=No

Index file=PruebaHTML.hhk

Language=0xc0a Español (España)

Title=Prueba ayudas HTML

.....

- **ALIAS:** identificador de nombre del archivo HTM. Cada archivo HTM de nuestra ayuda puede identificarse por un alias (nombre o variable) que lo identifica y relaciona con un ID que es utilizado posteriormente. Por ejemplo:

Ejemplo

```
[ALIAS]
IDH_top1=PruebaHTML1.htm
IDH_Top2=PruebaHTML2.htm
```

- **MAP:** archivo con la definición de los nombres de alias y un ID para cada uno que luego es utilizado en la aplicación bajo Windows. Esta relación de archivos H en el que se definen los ALLAS descritos en la sección anterior se relacionan con un ID que se utilizará posteriormente. Por ejemplo:

Ejemplo

```
[MAP]
include PruebaHTML.h
```

4.1. ARCHIVOS HHC (CONTENIDO)

En un archivo HHC se define la estructura del libro de contenidos de nuestra ayuda. El formato de un archivo de contenido es HTML.

Un ejemplo de archivo HHC sería:

Ejemplo

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<HTML>
<HEAD>
<meta name="GENERATOR" content="Microsoft® HTML Help Workshop 4.1">

</HEAD>
<BODY>
<OBJECT type="text/site properties">
<param name="Window Styles" value="0x800025">
<param name="ImageType" value="Folder">
</OBJECT>
<UL>
<LI>
```

```

<OBJECT type="text/sitemap">
<param name="Name" value="Principal"> </OBJECT>
<UL>
<LI>
<OBJECT type="text/sitemap">
<param name="Name" value="Principal">
<param name="Local" value="PruebaHTML1.html">
<param name="Comment" value="Comentario">
<param name="ImageNumber" value="10">
</OBJECT>
<LI>
<OBJECT type="text/sitemap">
<param name="Name" value="Capitulo III">
<param name="Local" value="Prueba HTML2.html">
</OBJECT>
</UL>
</UL>
</BODY>
</HTML>

```

.....

4.2. ARCHIVOS HHK (ÍNDICE)

Un archivo de índices también está descrito en formato HTML. En él se hacen referencia todas las cadenas de búsqueda para cada capítulo.

Ejemplo

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<HTML>
<HEAD>
<meta name="GENERATOR" content="Microsoft® HTML Help Workshop 4.1">

</HEAD>
<BODY>
<UL>
<LI>
<OBJECT type="text/sitemap">

```

```

<param name="Name" value="Capítulo III">
<param name="Name" value="Capítulo III">
<param name="Local" value="PruebaHTML2.html">
<param name="URL" value="Prueba HTML2.htm">
</OBJECT>
</LI>
<LI>
<OBJECT type="text/sitemap">
<param name="Name" value="Principal">
<param name="Name" value=""Capítulo II">
«param name=»Local» value=»Prueba HTML2.html»>
<param name="Name" value="Principal">
<param name="Local" value="Prueba HTML1.html">
</OBJECT>.
</LI>
</UL>
</BODY></HTML>

```

4.3. ARCHIVOS H (DECLARACIÓN DE ID DE CONTEXTO)

En un archivo H definimos (con el comando `#define`) los alias que referenciaremos a cada capítulo (archivos HTM) de nuestra ayuda.

Un ejemplo de definición de estos alias sería:

Ejemplo

```

#define IDH Top1 1
#define IDH Top2 2

```

4.4. ARCHIVOS HTM (CUERPOS DE LA AYUDA)

Los archivos HTM son archivos en formato HTML en los que relatamos toda nuestra ayuda. Cada capítulo o tema de nuestra ayuda conforma un archivo HTM.

5. TIPOS DE MANUALES: MANUAL DE USUARIO, GUÍA DE REFERENCIA, GUÍAS RÁPIDAS, MANUALES DE INSTALACIÓN, CONFIGURACIÓN Y ADMINISTRACIÓN, DESTINATARIOS Y ESTRUCTURA

La documentación de los programas es un aspecto sumamente importante, tanto en el desarrollo de la aplicación como en el mantenimiento de la misma. Mucha gente no da la suficiente importancia a esta parte del desarrollo con lo que se pierde, muchas veces, la posibilidad de la reutilización de código en otras aplicaciones o la dificultad posterior de modificaciones del mismo.

La documentación de un programa empieza a la vez que la construcción del mismo y finaliza justo antes de la entrega del programa o aplicación al cliente. Así mismo, la documentación que se entrega al cliente tendrá que coincidir con la versión final de los programas que componen la aplicación.

Una vez concluido el programa, los documentos que se deben entregar son una guía técnica, una guía de uso y de instalación.

5.1. TIPOS DE DOCUMENTACIÓN

La documentación que se entrega al cliente se divide claramente en dos categorías, interna y externa:

- **Interna:** es aquella que se crea en el mismo código, ya puede ser en forma de comentarios o de archivos de información dentro de la aplicación.
- **Externa:** es aquella que se escribe en cuadernos o libros, totalmente ajena a la aplicación en sí. Dentro de esta categoría también se encuentra la ayuda electrónica.

5.2. LA GUÍA TÉCNICA

En la guía técnica o manual técnico se reflejan el diseño del proyecto, la codificación de la aplicación y las pruebas realizadas para su correcto funcionamiento. Generalmente, este documento está diseñado para personas con conocimientos de informática, generalmente programadores. El principal objetivo es el de facilitar el desarrollo, corrección y futuro mantenimiento de la aplicación de una forma rápida y fácil.

Esta guía está compuesta por tres apartados claramente diferenciados:

5.2.1. Cuaderno de carga

Es donde queda reflejada la solución o diseño de la aplicación. Esta parte de la guía es únicamente destinada a los programadores. Debe estar realizado de tal forma que permita la división del trabajo.

5.2.2. Programa fuente

Es donde se incluye la codificación realizada por los programadores. Este documento puede tener, a su vez, otra documentación para su mejor comprensión y puede ser de gran ayuda para el mantenimiento o desarrollo mejorado de la aplicación. Este documento debe tener una gran claridad en su escritura para su fácil comprensión.

5.2.3. Pruebas

Es el documento donde se especifican el tipo de pruebas realizadas a lo largo de todo el proyecto y los resultados obtenidos.

5.3. LA GUÍA DE USUARIO

Es lo que comúnmente llamamos el manual del usuario. Contiene la información necesaria para que los usuarios utilicen correctamente la aplicación.

Este documento se hace desde la guía técnica pero se suprimen los tecnicismos y se presenta de forma que sea comprensible para el usuario que no sea experto en informática.

Un punto a tener en cuenta en su creación es que no debe hacer referencia a ningún apartado de la guía técnica y, en el caso de que se haga uso de algún tecnicismo, debe ir acompañado de un glosario al final de la misma para su fácil comprensión.

Las secciones de un manual de usuario a menudo incluyen:

- **Una página de portada.**
- **Una página de título.** Una página de derechos de autor.
- **Un prefacio,** que contiene detalles de los documentos relacionados y la información sobre cómo navegar por la guía del usuario.
- **Una página de contenido.**
- **Una guía sobre cómo utilizar al menos las principales funciones del sistema,** es decir, sus funciones básicas.
- **Una sección de solución de problemas** que detalla los posibles errores o problemas que pueden surgir, junto con la forma de solucionarlos.
- **Una sección de preguntas frecuentes (FAQ).** Dónde encontrar más ayuda y datos de contacto.
- **Un glosario** y, para documentos más grandes, un índice.

5.4. LA GUÍA DE INSTALACIÓN

Es la guía que contiene la información necesaria para implementar dicha aplicación. Dentro de este documento se encuentran las instrucciones para la puesta en marcha del sistema y las normas de utilización del mismo. Dentro de las normas de utilización se incluyen también las normas de seguridad, tanto las físicas como las referentes al acceso a la información.

6. CONFECCIÓN DE TUTORIALES MULTIMEDIA

Se da por hecho que preparar un tutorial multimedia es algo fácil o instantáneo, se capta el momento con un móvil, lo plasma en un programa presentación multimedia, y listo. Es justo todo lo contrario. Los grandes tutoriales multimedia no admiten improvisaciones, y si es de los que piensan que sí, debe saber que hay un riesgo importante de ser catalogado, como un productor de tutoriales aficionado.

Si hace grandes esfuerzos para consolidar su marca personal y construir poco a poco una autoridad en su nicho, tirar por la borda tantos esfuerzos por culpa de una mala estrategia de marketing tutoriales es asumir un riesgo.

Aquí se presentan una serie de consejos importantes para preparar material multimedia.

6.1. ¿ENTIENDE LAS NECESIDADES DE SU TARGET?

Su misión es sencilla: hacer un material útil.

La idea es que eduque a su audiencia sobre una temática específica, interesante y relevante para su actividad. Incentivar a su audiencia para que ponga en marcha lo aprendido en su propio proyecto como un “hazlo tú mismo”.

¿Y por qué haría esto? Después de todo, educar al mercado es una inversión (para muchos casi una molestia). Pues porque si la gente sigue sus consejos y ve que en muy poco tiempo dan resultados, el plus de confianza y credibilidad que te habrá ganado es decisivo en el momento de transformar un lector o un seguidor en un cliente para su actividad.

Así que antes de crear un tutorial multimedia, asegúrese de que hay una necesidad clara comunicada por su audiencia. Y, para ello, primero es importante tener un blog y analizar los comentarios que le llegan y las peticiones que recibe a través del formulario de contacto antes de crear tutoriales multimedia.

Si está empezando puede “alquilar una audiencia” en foros verticales y comunidades online.

Entregar al mercado lo que el mercado espera de usted, es un regla fundamental para su éxito.

6.2. SEA BREVE, LA ATENCIÓN DISMINUYE RÁPIDO

Los tutoriales multimedia más populares suelen ser muy cortos. Los vídeos largos pueden funcionar (puede conseguir un 25% de atención después de 30 minutos de vídeo), pero en este caso el contenido debe ser Premium.

Pero tenga en cuenta que es mucho más seguro apostar por un formato más corto y, además, le costará menos producirlo.

6.3. ESPIAR A LA COMPETENCIA

Ver y analizar lo que otras personas han hecho sobre la temática sobre la que usted quiere hacer su tutorial.

6.4. PREPÁRESE ANTES DE APRETAR EL BOTÓN DE GRABAR

Como ve, no hay sitio para la improvisación si quiere grabar grandes tutoriales. El secreto está en tener un plan y ejecutarlo al pie de la letra, estar enfocado en lo que va a decir y hacerlo de forma breve.

6.5. CONCLUSIÓN

Resuma lo que ha presentado en el tutorial de forma muy breve.

7. HIERRAMIENTAS DE CAPTURA DE PANTALLAS Y SECUENCIAS DE ACCIONES

Aunque el vídeo es una herramienta muy útil para explicar procedimientos, en muchas ocasiones tan solo necesitamos de unas pocas capturas de pantalla para explicar algo. En momentos como esos, especialmente si escribe en un blog de software e Internet, le será de gran ayuda tener buenas aplicaciones a mano.

Se presenta una selección de siete aplicaciones gratuitas para Windows para hacer capturas de pantalla. Aunque Windows 7 y 8 ya viene con su propia herramienta de captura (Recortes), esta es bastante limitada, sirviendo solo para salir de un apuro. Por eso, conviene tener a mano programas como los que vamos a ver a continuación.

7.1. ZSCREEN

Está enfocado principalmente a poner nuestras capturas online. Nos permite subirlas a varios servicios al mismo tiempo o a nuestro servidor por medio de FTP. Tiene las funciones usuales de captura y algunas otras más como marcas de agua automáticas, captura y subida de texto (e incluso traducción) y captura de colores. No dispone de funciones de edición, pero podemos elegir fácilmente usar el que tengamos instalado, y nos permite configurar muchas de sus características.

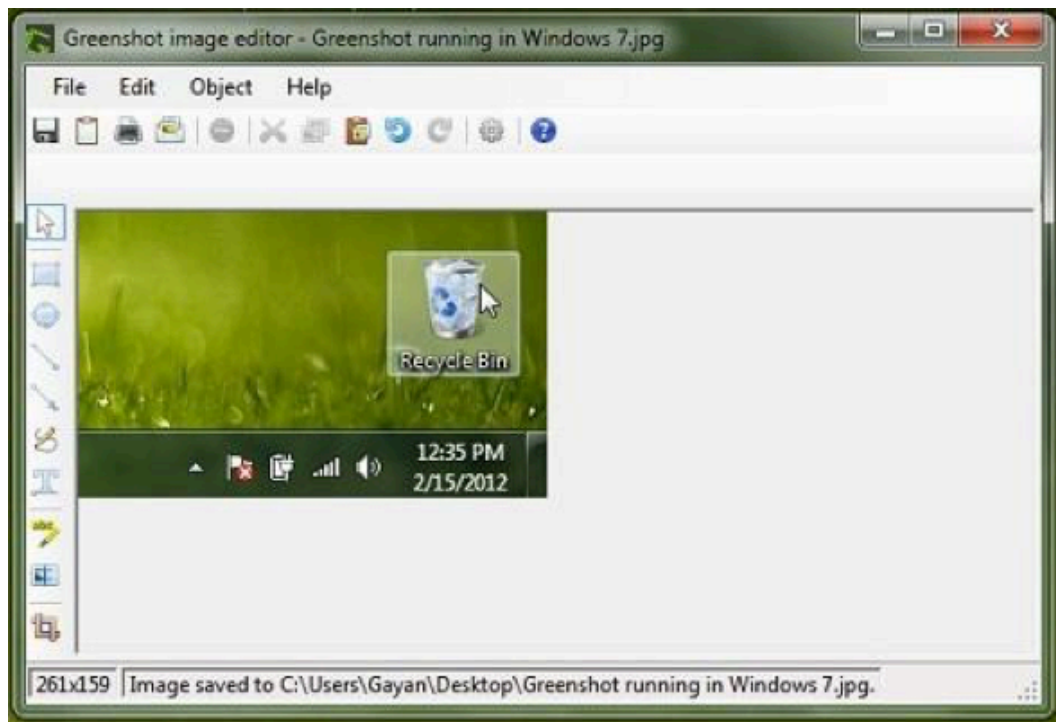
Puede descargar Zscreen de <https://sourceforge.net/projects/zscreen/>



7.2. GREENSHOT

Es un pequeño programa bastante minimalista pensado para que lo lleves a todas partes, pues no necesita de instalación. Dispone de un pequeño editor incorporado que nos permite modificar nuestra imagen y añadirle flechas, resaltar elementos, ofuscarlos, etc. Sin ser una maravilla, cumple con su función de manera correcta y, al ser portátil, nos permite llevarlo a cualquier parte en nuestra memoria USB.

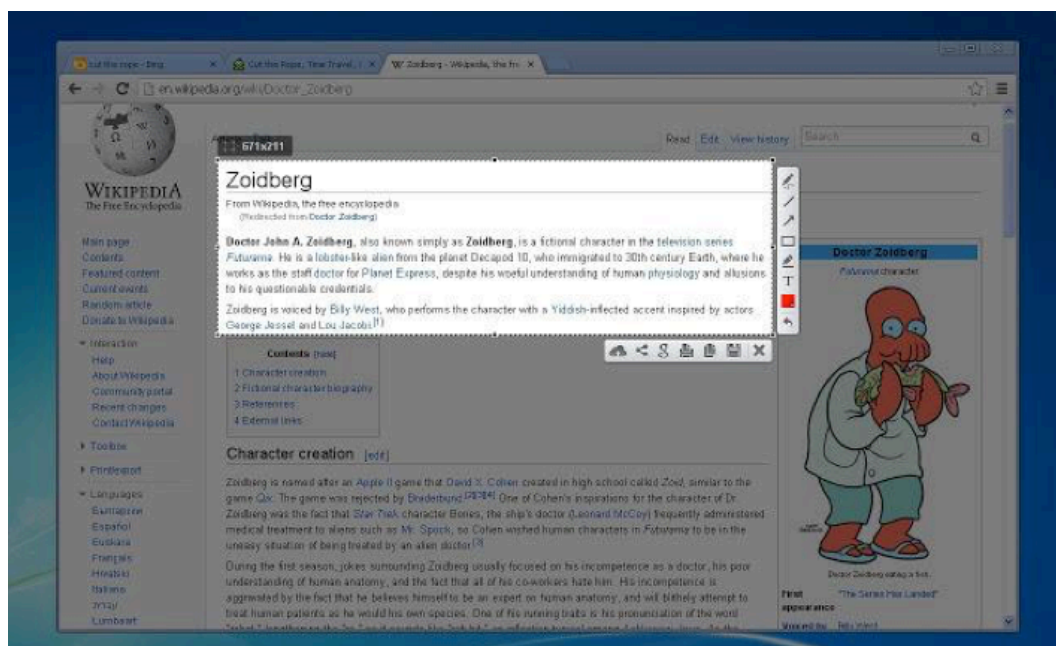
Puede descargar Greenshot de <https://sourceforge.net/projects/greenshot/>



7.3. LIGHTSHOT

Es un programa que es tanto una aplicación individual para PC como una extensión para Firefox, IE y Chrome. Nos permite guardar las imágenes en nuestro disco, ponerlas directamente on line o editarlas en la nube por medio de Pixlr. Desde luego, la opción más minimalista de todas, contando únicamente con las opciones indispensables y trasladando el resto a la nube.

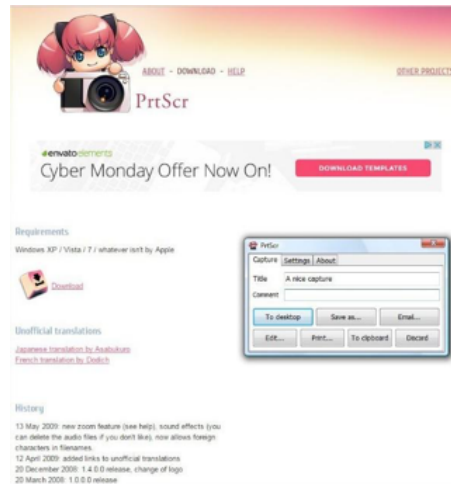
Puede descargar Lightshot <https://app.prntscr.com/es/index.html>



7.4. PRISRC

Otra aplicación de la que ya hablamos, con un espectacular efecto posterior a la captura en la que ésta se queda flotando ondulante sobre el escritorio. Más allá de eso, lo usual: captura de pantalla completa, ventanas y a mano alzada, que quizás es la que nos permite sacarle más partido al programa, haciendo las capturas como si se tratasen de recortes.

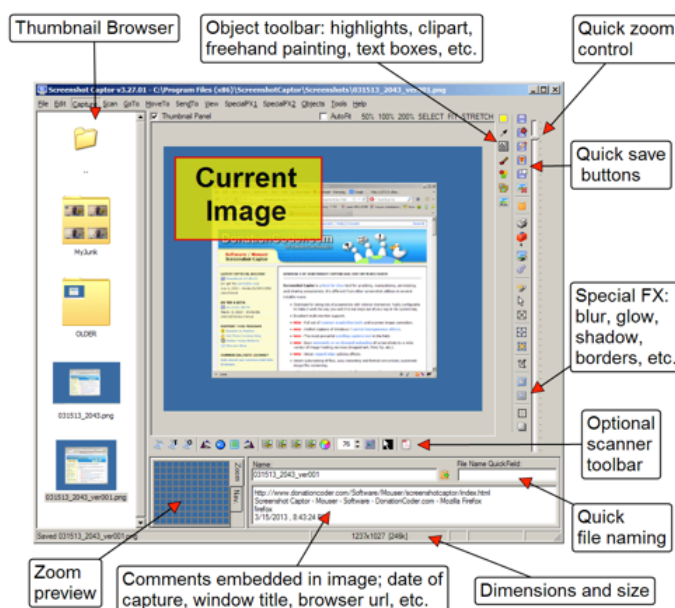
Puede descargar PrtScr de <http://www.fiastarta.com/PrtScr/Download.html>



7.5. SCREENSHOT CAPTOR

Uno de los más completos, con muchas opciones, tanto para capturar como para editar. Además, nos conserva un historial de todas las capturas que hemos tomado. El editor es muy completo y no necesita de un programa adicional, pues además cuenta con una gran variedad de imágenes incluidas en su galería. Eso sí, como forma parte de DonationCoder, necesitará un código gratuito para poder usarlo sin que aparezca la ventana del inicio.

Puede descargar Screenshot captor de <https://www.donationcoder.com/Software/Mouser/screenshotcaptor/>



7.6. JET SCREENSHOT

Otra opción pequeña pero potente de la que ya os contamos en Genbeta. Tiene un pequeño editor bastante funcional con el que podemos añadir indicaciones rápidamente. Nos permite subir las capturas directamente a nuestro FTP o a su web, donde permanecerá únicamente durante 24 horas. Es ligero e ideal si tan solo necesita unas pocas anotaciones en la imagen.

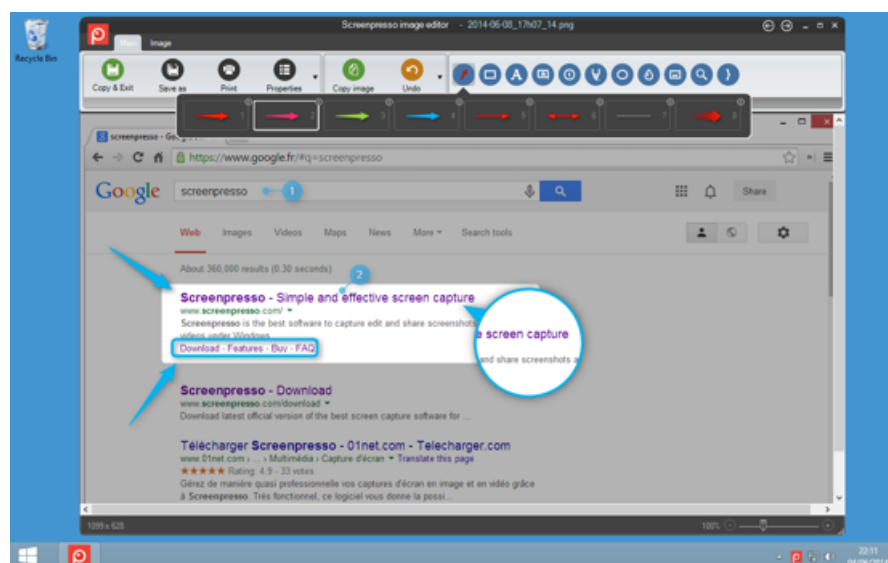
Puede descargar Jet Screenshot de <https://jetscreenshot.com/download.php>



7.7. SCREENPRESSO

Con un diseño atractivo y funcional, personalmente es este uno de los programas que más he usado. Tiene un historial de capturas junto a un pequeño editor que nos permite modificar la imagen recortándola o redimensionándola y añadirle elementos como flechas, recuadros, texto, etc. También nos da la opción de compartir nuestras capturas directamente en Twitter, Facebook, Flickr y otros medios.

Puede descargar Screenpresso de <https://www.screenpresso.com/download/>



8. HERRAMIENTAS PARA LA CONFECCIÓN DE TUTORIALES INTERACTIVOS, SIMULACIÓN

Una de las alternativas multimedia para elaborar contenidos multimedia, es mostrar lo que hacemos en nuestra pantalla, es decir, realizar una videocaptura o screencasting.

8.1. VERSIÓN ESCRITORIO

Las cuatro alternativas que se presentan ahora implican que debemos descargar e instalar algún tipo de software.

8.1.1. Aviscreen

Aviscreen nos permite grabar nuestra pantalla en formato AVI o bien tomar fotos de nuestra pantalla en formato JPG.

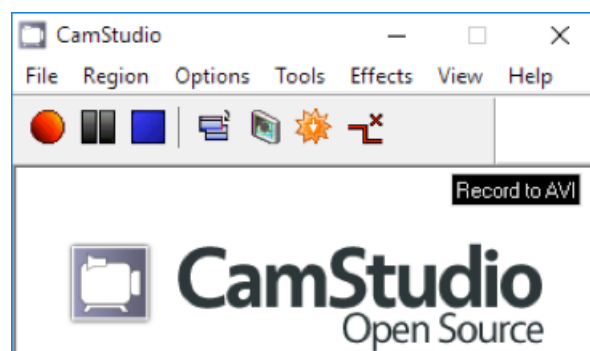
Puede descargar Screenpresso de <https://aviscreen-classic.jaleco.com/>



8.1.2. CamStudio

Es un software opensource o de código abierto que nos permite grabar nuestra pantalla en formato AVI. Funciona únicamente con Windows. Totalmente gratuito.

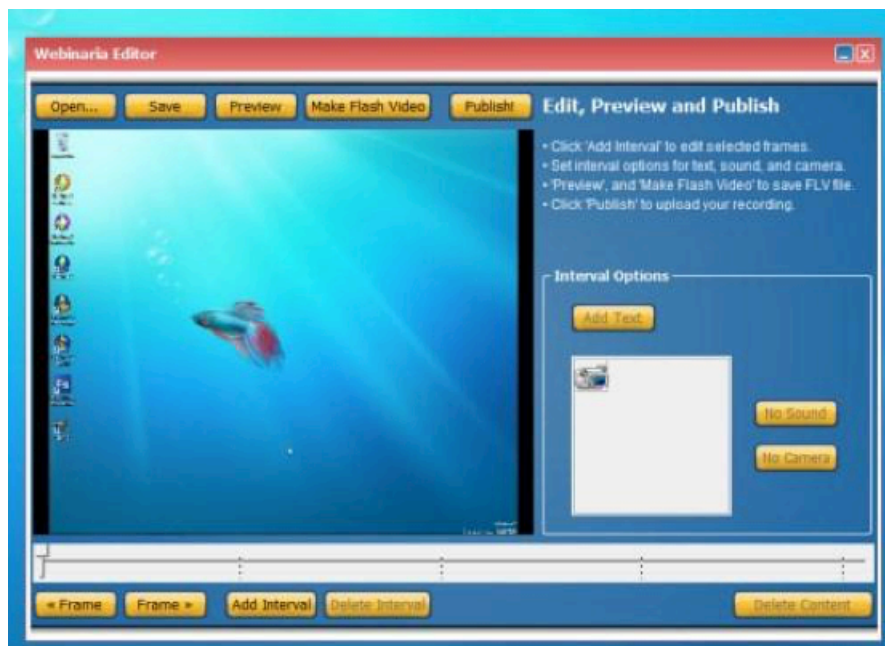
Puede descargar CamStudio de <https://sourceforge.net/projects/camstudio/>



8.1.3. Webinaria

Webinaria es otro software open source, en este caso podemos grabar en formato AVI y FLV. La principal ventaja de Webinaria es la posibilidad de grabar audio junto al vídeo de nuestra pantalla.

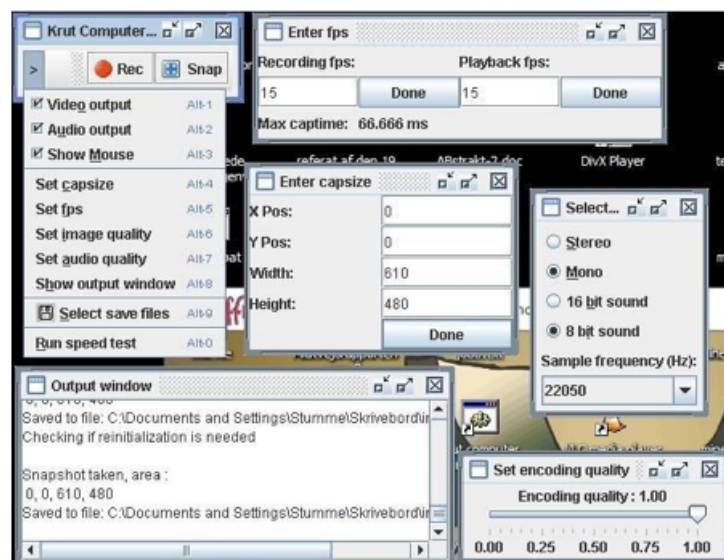
Puede descargar Webinaria webide <https://sourceforge.net/projects/webinaria/>



8.1.4. Krut

Es otra alternativa open source. En este caso los vídeos son grabados en formato quicktime (mov). Krut, además permite grabar audio y un sector específico de la pantalla.

Puede descargar Krut webide <https://sourceforge.net/projects/krut/>



8.1.5. Jing

Además de grabar nuestra pantalla Jing nos permite tomar fotos de la misma, subir nuestro vídeo a una cuenta gratuita o bien descargar el mismo el formato SWf.

Puede descargar Jing webide <https://www.techsmith.com/jing.html>

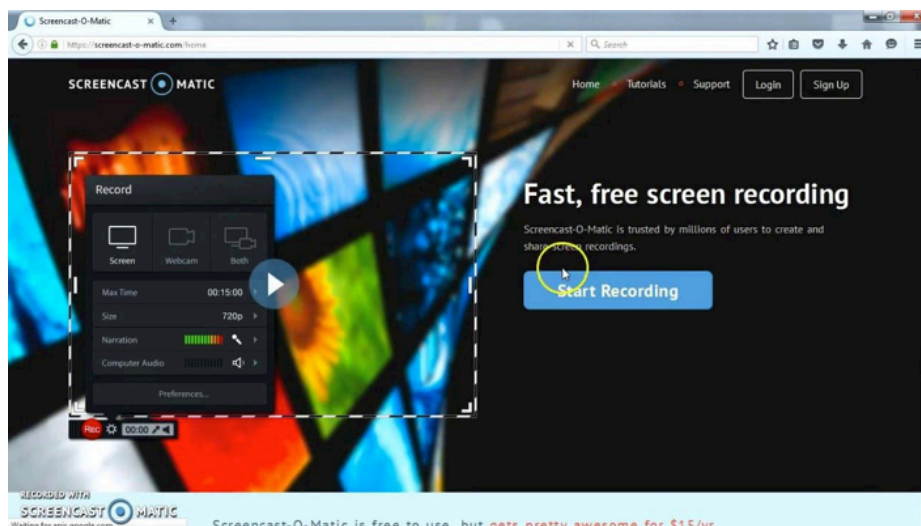


8.2. VERSIONES WEB

A continuación presentamos 2 alternativas web, es decir, con registrarse puede grabar su pantalla sin necesidad de descargar o instalar nada.

8.2.1. Screencast-O-Matic

Una web basada en Java que permite grabar nuestra pantalla sin necesidad de descargar software alguno. webide <https://screencast-o-matic.com/>



8.2.3. ScreenToaster

Screen Toaster permite grabar nuestros vídeos sin necesidad de descargar nada, enviar el vídeo por email o bien anexarlo en una web con un solo clic

webide <http://www.icyte.com/system/snapshots/fs1/6/1/3/6/6136fdcd5ad0da964634b9747f65ac5f2df32d5c/index.html>



9. COMPONENTES DE UNA APLICACIÓN. EMPAQUETADO

El empaquetado de aplicaciones consiste en proporcionar a los futuros usuarios las aplicaciones en forma de paquetes, a los que se suele llamar en inglés: software bundle o application bundle. Estos paquetes están formados por:

- Los programas ejecutables de la aplicación.
- Las bibliotecas necesarias de las que depende. Otro tipo de ficheros (como imágenes, bases de datos, ficheros de audio, traducciones y localizaciones, etc.). Todos en ellos dentro del mismo fichero forman un todo o un conjunto.
- Las bibliotecas de las que depende el programa pueden haber sido enlazadas tanto de forma dinámica como también estática.

Independientemente de ello, el usuario percibe que el paquete como un conjunto que representa al programa en sí, cuando en realidad incluye varios ficheros.

Una de las mayores ventajas de un correcto empaquetado de aplicaciones es que permite evitar los problemas de las dependencias tanto a la hora de instalar la aplicación como a la hora de usarla, ya que cada paquete lleva consigo sus dependencias, y la instalación o desinstalación de otro software no va a afectar a las dependencias de dicho paquete. De ahí radica su principal objetivo, el evitar la problemática de las dependencias, y que la aplicación se puede trasladar de un computador a otro sin necesidad de reinstalarla, ya que el paquete de la aplicación contiene todos los ficheros necesarios para ejecutarla.

Sin embargo, como desventaja se presenta que estos paquetes ocupan mucho más espacio en el disco, especialmente si el paquete incluye bibliotecas.

Por lo general, cada distribución tiene su propia forma de empaquetar sus aplicaciones:

9.1. LINUX

Tenemos dos tipos de paquetes que sobresalen del resto:

- **.rpm**: (Redhat Package Manager) de la familia de RedHat (RHEL, Fedora, CentOS), Mandriva, Suse.
- **.deb**: de la familia de Debian (Debian, Ubuntu, y derivados).

9.2. WINDOWS

Tomando como base su IDE Visual Studio el formato de empaquetado es msi. Se definen como instaladores de Microsoft, a saber, aquellos paquetes de software que contienen la información necesaria para automatizar su instalación, minimizando la intervención manual del usuario, ya que toda la información iría contenida en el propio fichero msi. La información de instalación y, a menudo, los archivos mismos, son empaquetados en paquetes de instalación, bases de datos estructuradas como OLE Structured Storage (almacenamiento estructurado de ficheros) y comúnmente conocido como "MSIfiles" por su extensión de archivo. El sistema de archivos msi es muy útil cuando queremos distribuir aplicaciones en equipos pertenecientes a un dominio Windows.

9.3. JAVA

El principales jar. Es un tipo de archivo que permite ejecutar aplicaciones escritas en el lenguaje Java. Las siglas están deliberadamente escogidas para que coincidan con la palabra inglesa “jar” (tarro). Los archivos JAR están comprimidos con el formato zip y cambiada su extensión a jar. Es el más popular entre los dispositivos móviles y en los SO más importantes.

10. INSTALADORES. PAQUETES DE INSTALACIÓN. HERRAMIENTAS PARA CREAR PAQUETES DE INSTALACIÓN. PARÁMETROS DE LA INSTALACIÓN.

Un paquete de instalación es un conjunto de archivos necesarios para instalar una aplicación. Un paquete contiene una distribución de la aplicación, archivos de configuración (del proceso de instalación y de la configuración inicial de la aplicación), y el resto de archivos de la aplicación.

10.1. WINDOWS

En equipos Windows existen los ficheros que comúnmente son llamados ejecutables. Los primeros fueron los llamados bat (extensión de un fichero formado por un lote de ordenes MS-dos, que ya no se usan), los exe y los msi. Estos dos últimos son utilizados con mayor frecuencia en el entorno gráfico de Windows.

La herramienta encargada de llevar a cabo la instalación es Windows Installer que es un motor para la instalación, mantenimiento y eliminación de programas en plataformas Windows.

Los paquetes MSI (MicroSoft Installer) se definen como instaladores de Microsoft o paquetes de software que contienen la información necesaria para automatizar su instalación, minimizando la intervención manual del usuario, ya que toda la información iría contenida en el propio fichero "msi".

Una de las ventajas de los paquetes msi es su facilidad para la distribución de software desde servidores Windows Server. Estos paquetes se pueden crear incluso con software de terceros para luego instalarlos en equipos en red pertenecientes a un dominio.

Un paquete describe la instalación completa de un producto, ya que Windows Installer no maneja dependencias entre productos y está identificado por un GUID. El paquete está compuesto de componentes agrupados dentro de sus características.

Un componente es la mínima parte de un producto. Cada componente es tratado por Windows Installer como una unidad. Los componentes pueden contener archivos, grupos de archivos, directorios, componentes COM, claves del registro de Windows, accesos directos y otro tipo de datos. El usuario final no interviene directamente con los componentes.

También los componentes al estar identificados globalmente por GUID, ello permite que un mismo componente sea compartido entre varios del mismo paquete o de múltiples paquetes, idealmente a través del uso de la unión de módulos (aunque para trabajar correctamente, diferentes componentes no deberían compartir ningún subcomponente).

Una ruta maestra es un fichero específico, clave de registro, o fuente de datos ODBC que el autor del paquete especifica como crítico para un componente dado. Como las rutas maestras más utilizadas son en forma de fichero, se suele utilizar el término fichero maestro (key file). Un componente puede contener, a lo sumo, una ruta maestra; si un componente no tiene establecida de manera explícita una ruta maestra, el directorio destino del componente es tomado como la ruta maestra. Cuando se ejecuta una aplicación basada en MSI, Windows Installer comprueba la existencia de estos ficheros críticos o claves de registro (es decir, las rutas maestras). Si existe un desajuste entre el estado actual del sistema y el valor especificado en el paquete MSI (ejemplo: un fichero maestro desaparecido), entonces la característica asociada es reinstalada. Este proceso es también conocido como auto reparación. Dos componentes no pueden utilizar la misma ruta maestra u otra.

InstallShield es una herramienta de software para crear instaladores. InstallShield se utiliza sobre todo para instalar software del escritorio y las plataformas de servidor de Windows, pero también se puede usar para administrar aplicaciones y paquetes de software en una amplia gama de móviles y portátiles. Fue desplazado por Windows Installer.

Por otro lado, para la creación del instalador msi de Windows, Visual Studio tiene su propia herramientas VSI (Visual Studio Installer). Con el Visual Studio Installer, se puede desarrollar un proyecto de instalación creando un archivo.wip como una parte integral de su solución. El archivo de paquete de instalación de Windows (msi), se construye a partir del proyecto de Visual Studio Installer (wip) que contiene todos los datos e instrucciones necesarias para instalar la aplicación.

En la red hay diferentes manuales que explican su uso, cosa no complicada, ya que se trata de un asistente.

10.2. LINUX

10.2.1. Compilar el paquete

Esta es la forma clásica, y antigua, de instalar paquetes. Consiste en bajar el código fuente, comprimido en un archivo targzo.tar.bz2.

Una vez bajado, entramos en la consola (shell) y nos movemos hasta el directorio donde tengamos el paquete. Si el paquete está en formato targz escribimos:

```
# tar -xzf archivo.tar.gz //para descomprimir el paquete
```

Si está en tar.bz2 escribimos:

```
bzip2 -dic archivo.tar.bz2 tar -xv
```

Una vez hecho esto, hay que entrar en el directorio creado y compilar el código para obtener la aplicación funcional y que consiste en escribir en la siguiente línea de comandos lo siguiente:

```
it. . / configure it make it make install
```

Lo que estamos haciendo es compilando el programa a partir de código fuente.

Uno de los principales problemas de este método es si el paquete tiene dependencias, es decir, si depende de algún otro paquete para que funcione correctamente. En ese caso, habrá que instalarlos manualmente.

10.3. PAQUETES .DEB Y .RPM

Los paquetes .deb, y rpm son un método de instalación muy efectivos para sus respectivas distribuciones.

Los paquetes .deb, son paquetes que se pueden instalar en la distribución Debian y derivados (Ubuntu, Kubuntu...).

Los rpm (RedHat Package Manager) son los de la distribución Red Hat y derivados (OpenSuse, Mandriva, Fedora...).

Un paquete rpm no lo podremos instalar en la distro Debiano derivados, y un deb tampoco en RedHat y derivados. No obstante, existen programas que permite convertir un paquete rpm a deb y viceversa.

Para instalar un paquete debentramos la siguiente línea de comandos en la consola (situándonos en el directorio donde está el paquete).

```
dipkg -i nombre del paquete. deb
```

Para crear un paquete deb ya hay asistentes que facilitan bastante su creación. Uno de ellos es checkinstal. Los pasos son los siguientes:

Para empezar, tenemos que instalar checkinstall. Así que hacemos (como root):

```
apt-get install checkinstall
```

Lo siguiente es ir a la carpeta en la que tenemos el código de la aplicación y abrir una terminal. Ejecutamos los siguientes comandos, uno a uno:

```
it. ./configure
```

```
it make
```

Con “./configure” se configuran los paquetes para nuestra distribución y se crea un “Makefile” (un archivo que contiene instrucciones de compilación) y con “make” se compila el código y deja los binarios, librerías, etc., en la carpeta “src”. Ahora, antes de continuar, es recomendable no tener instalada la aplicación de la que se hará el paquete. Si lo está:

```
# make un install
```

Y, entonces, es momento de comenzar con la parte importante, el uso de checkinstal. En esa misma terminal, escribimos:

```
# Checkinstall
```

Y se abrirá el “asistente” de checkinstall. En él podemos modificar la información que tendrá el paquete que crearemos. Las opciones que podemos modificar son:

- **Maintainer:** el desarrollador principal del paquete.
- **Summary:** una descripción del paquete.
- **Name:** nombre que quieres darle al paquete.
- **Versión:** versión del paquete.
- **Release:** viene siendo la versión principal del paquete, podemos dejarlo como venga.
- **License:** licencia de la aplicación, es preferible no tocarlo.
- **Group:** grupo por el cuál fue creado, podemos dejarlo como está.
- **Architecture:** arquitectura de procesador del paquete.
- **Source location:** nombre de la carpeta (solo la carpeta, no la ruta entera) en la que está el código del paquete.
- **Alternate source location:** no es necesario modificarlo.
- **Requires:** dependencias que deben ser instaladas para su correcto funcionamiento.

- **Provides:** nombre del paquete que provee, no es necesario modificarlo.
- **Conflicts:** paquetes con los que entra en conflicto.
- **Replaces:** paquetes a los que reemplaza.

Cada una tiene un número a su izquierda, así que para editarla solo escribimos su número. No es necesario modificar todas, pero sí las más importantes como Maintainer.

10.3.1. Summary, Name y Version

Una vez hayamos modificado lo que queremos, presionamos Enter (sin ningún número previo) y comenzará a compilar e instalar el paquete. Cuando haya terminado, en el directorio donde compilamos habrá aparecido un paquete deb de la aplicación, listo para instalar. Hay que asegurarse en "Version" de que no haya letras. Eso suele evitar que se cree el paquete.

Es posible que "Requires" dé un fallo con dejar el espacio en blanco. En RedHat para instalar un paquete rpm introducimos:

```
rpm -install nombre del paquete. rpm
```

Para crear un paquete rpm.

```
yum groupinstall Development Tools" i yum install rpmdevtools i yum install rpmlint
```

```
rpmdev-setuptree
```

Ésto nos creará un entorno de trabajo para crear los RPM o "Árbol del Proyecto".

Necesitaremos dos cosas: un Archivo "spec" y el código fuente del programa para instalar. El archivo "spec" lo puedes conseguir de cualquier "src, rpm" creado para tu versión de Fedora (con una versión anterior del código fuente por ejemplo) o con un "src. rpm" para una versión antigua de fedora, una fuente para dicho tipo de paquetes está en: <http://www.rpmfind.net/>

Debemos abrir el paquete "src, rpm" que hayas descargado con un gestor de archivadores, ahí dentro encontrarás el "spec". Debido a que lo que haremos será construir un Nuevo RPM desde código fuente más reciente, generalmente solo debemos cambiar el campo "Version" y agregar un "Changelog" y fijarse en el campo "Release". Por ejemplo, usar el paquete de Kimes 2.0.5-2 de Fedora 14 para crear el 2.0.6.1-0 usable en el mismo Fedora, en ese caso, solo modificaríamos el apartado Version por el nuevo, quitando el 2 de Release y poniendo un 0 ("Release" es el número que aparecerá después del guión) y agregando un changelog copiando uno de los ejemplos anteriores y modificándolo según aplicó con el changelog oficial.

Otra cosa que es chequear el archivo "spec" en su apartado de #Build Requires ya que todo lo que esté en este apartado son dependencias sin las cuales el paquete no podrá ser creado. Instalamos todos los paquetes requeridos con: yum -y install paquete1 paquete2 paquete3 Vamos a la carpeta "rpmbuild" y luego en la carpeta de SOURCES debemos colocar el código fuente de nuestra aplicación, el "tarball" para explicarme mejor. Por otro lado, debemos copiar nuestro "spec" modificado recientemente creado en SPECS y continuamos con el siguiente paso.

Usamos una pequeña herramienta llamada rpmlint para nuestro cometido. Ésta herramienta nos avisará de errores en la estructura del archivo que acabamos de modificar/escribir y su uso es bastante sencillo:

```
rpmlint program. Spec
```

Se nos mostrará un Output donde nos dará información acerca de nuestro archivo. Si hay errores o advertencias, simplemente tenemos que hacer caso y modificar según se necesite.

Ya tenemos todo listo para la construcción:

```
it rpmbuild -ba program. spec
```

Tras un tiempo de compilado y trabajo, tendremos un paquete RPM listo para usar, su "src.rpm" y también los paquetes adicionales que se hayan creado.

Terminada la creación del paquete, si todo salió bien, podremos irnos al directorio rpmbuild y en el sub-directorio RPMS encontraremos los binarios de nuestra aplicación y los extras que la acompañen, mientras que en el directorio SRPMS encontraremos los paquetes "src.rpm" para la misma aplicación. Tener a mano estos dos tipos de paquetes es lo que debería importarte. Ahora simplemente se chequea el RPM final generado (el que usaremos para la instalación de la aplicación) con rpmlint:

```
rpmlint *. rpm
```

El último paso tanto en deb como rpm es subirlos a una web de repositorios para que puedan ser descargados con aptitude o apt-get en el primer caso o con yum en el segundo caso.

11. ASISTENTES DE INSTALACIÓN Y DESINSTALACIÓN. PAQUETES AUTOINSTALABLES

11.1. WINDOWS

Usamos una pequeña herramienta llamada rpmlint para nuestro cometido. Ésta herramienta nos avisará de errores en la estructura del archivo que acabamos de modificar/escribir y su uso es bastante sencillo:

```
it rpmlint program. Spec
```

Se nos mostrará un Output donde nos dará información acerca de nuestro archivo. Si hay errores o advertencias, simplemente tenemos que hacer caso y modificar según se necesite.

Ya tenemos todo listo para la construcción:

```
it rpmbuild -ba program. spec
```

Tras un tiempo de compilado y trabajo, tendremos un paquete RPM listo para usar, su "src.rpm" y también los paquetes adicionales que se hayan creado.

Terminada la creación del paquete, si todo salió bien, podremos irnos al directorio rpmbuild y en el sub-directorio RPMS encontraremos los binarios de nuestra aplicación y los extras que la acompañen, mientras que en el directorio SRPMS encontraremos los paquetes "src.rpm" para la misma aplicación. Tener a mano estos dos tipos de paquetes es lo que debería importarte. Ahora simplemente se chequea el RPM final generado (el que usaremos para la instalación de la aplicación) con rpmlint:

```
rpmlint *. rpm
```

El último paso tanto en deb como rpm es subirlos a una web de repositorios para que puedan ser descargados con aptitude o apt-get en el primer caso o con yum en el segundo caso.

Los archivos pueden ser modificarse con cualquier editor de texto plano/ASCII, como el Bloc de notas. Todos los archivos autoextraíbles creados por el uso de Iexpress se comprimen con el Make Cab (MAKE-CAB.EXE) herramienta, y se extraen mediante el Wextract (wextract.exe) de Microsoft.

Iexpress.exe se encuentra en la carpeta System32 de Windows. La interfaz frontal (IExpress Wizard) se puede iniciar manualmente escribiendo IExpress en la ventana Ejecutar del menú Inicio. También se puede utilizar desde la línea de comandos (o archivo por lotes en la consola) para crear paquetes de instalación personalizada o desatendida (funcionamiento automático):

```
c:> IEXPRESS / N letraDeUnidad \, directory name X file name. SED
```

A través de la creación de interfaz del Asistente de IExpress le permite al usuario especificar un título para los paquetes, añadir la solicitud de la licencia de usuario final que debe ser aceptado como una orden para permitir la extracción, seleccione los archivos que se archivan, la visualización avanzada ventana de opciones y, finalmente, especificar un mensaje que se mostrará al finalizar.

La desinstalación de software es el proceso de revertir los cambios producidos en un sistema por la instalación del mismo. Por ello no solo deben ser borrados los archivos, sino también cambios en otros aspectos del software como, por ejemplo, eliminar usuarios que hayan sido creados, retirar derechos concedi-

dos, borrar directorios creados hasta llevar la contabilidad en un sistema de gestión del sistema, en el caso de Windows el Registro.

Debido a la creciente complejidad de sistemas operativos y sus interfaces (API), la desinstalación de software puede ser no solo contraproduktiva, sino también poner en peligro la estabilidad del sistema. Por esta razón la calidad de un software no solo depende de sus efectos productivos o creativos, sino también de su capacidad de integración en el sistema operativo y compatibilidad con otros programas. El desarrollador del software debe ofrecer una función para desinstalar su software sin dañar o desestabilizar el sistema.

Cada vez es más difícil la desinstalación, dado que muchas bibliotecas se comparten entre aplicaciones de diferentes productores de software que utilizan enlaces duros o simbólicos a través del directorio.

En sistemas de alta complejidad, el esfuerzo para desinstalar un programa puede ser mayor que el de la instalación.

En Windows la desinstalación mejora si utilizamos software que además de eliminar los archivos del software eliminen las entradas en los registros. Esto es debido a que el desinstalador que lleva Windows (Agregar y Quitar Programas) no es muy fiable en ese aspecto. Existen varias herramientas como por ejemplo es Ccleaner, RegCleaner, Revo Uninstaller o IObit Uninstaller entre otras, que además de desinstalar hacen búsquedas en la base de datos del Registro para eliminar aquellas entradas que no están asociadas a ningún software instalado.

11.2. LINUX

En cuanto a Linux los comandos para desinstalar aplicaciones son los siguientes:

- **Desinstalar paquete:** `sudo apt-get remove "nombre-paquete."`
- **Borrar los archivos de configuración:** `sudo apt-get purge nombre-paquete"`
- **Borrar los archivos descargados con la aplicación:** `sudo apt-get clean nombre-paquete"`

12. FICHEROS FIRMADOS DIGITALMENTE

El formato de firma es la forma como se genera el documento de firma y como se guarda o estructura la información de firma en el documento generado.

La existencia de múltiples formatos de firma se debe a razones históricas, a cómo se ha ido introduciendo la firma en formatos de documentos ya existentes y a cómo se han ido añadiendo funcionalidades a lo largo del tiempo.

Un fichero de firma tiene un formato que viene determinado por estos aspectos:

12.1. ESTRUCTURA DE LA FIRMA

Una firma electrónica es un fichero que contiene información sobre el documento original, el firmante, la fecha de la firma, algoritmos utilizados y posible caducidad de la firma.

Cómo se estructura esta información (el orden de esa información dentro del fichero, las etiquetas que indican cuando empieza un campo y cuándo termina, la opcionalidad de esos campos, etc.) viene determinado por distintos formatos:

12.1.1. CAdES (CMSAvanzado)

Es la evolución del primer formato de firma estandarizado. Es apropiado para firmar ficheros grandes, especialmente si la firma contiene el documento original, porque optimiza el espacio de la información. Tras firmar, no podrás ver la información firmada, porque la información se guarda de forma binaria.

12.1.2. XAdES (XML, Avanzado)

El resultado es un fichero de texto XML, un formato de texto muy similar al HTML que utiliza etiquetas. Los documentos obtenidos suelen ser más grandes que en el caso de CAdES, por eso no es adecuado cuando el fichero original es muy grande. Aplicaciones como ecoFirma del Ministerio de Industria y Comercio, solo firman en XAdES.

12.1.3. PAdES (PDF Avanzado)

Este es el formato más adecuado cuando el documento original es un pdf. El destinatario de la firma puede comprobar fácilmente la firma y el documento firmado. Con los formatos anteriores esto no es posible si no se utilizan herramientas externas.

12.1.4. OOXML y ODF

Son los formatos de firma que utilizan Microsoft Office y OpenOffice, respectivamente.

12.2. ¿DÓNDE SE GUARDA EL DOCUMENTO ORIGINAL?

Según cómo se referencia o dónde se guarde el documento original en el fichero de firma, podemos tener dos casos:

12.2.1. El documento original se incluye en el fichero de firma

Ventaja: no es necesario guardar siempre el documento original y el documento de firma porque aquel ya está incluido en éste. Es, por tanto, un formato cómodo de almacenar.

Desventaja: si el tamaño del fichero es elevado, se consume más espacio de almacenamiento, porque al final se acaba teniendo, por un lado, el documento original, que siempre habrá que guardarlo y, por otro, la firma.

En el caso de CAdES estas firmas se llaman firmas implícitas.

En el caso de firmas XAdES XML, lo habitual es que el documento esté incluido en el fichero de firma. Hablamos de firmas despegadas (detached), envoltentes (en veloping) y envueltas (en veloped) según en qué sitio del propio fichero de firma se guarde el documento original.

En la práctica, se suele utilizar el caso 1, que es la forma de funcionar por defecto de las aplicaciones de firma. Se obtienen ficheros de firma más grandes pero, como contrapartida, no requiere almacenar el fichero original como otro documento aparte junto al de firma.

12.2.2. El documento no se incluye en la firma

En este caso, el documento no se incluye en el resultado de firma o solamente se incluye una referencia al lugar en el que se encuentra para que el documento pueda ser localizado. Por tanto, se obtienen ficheros de tamaño más reducido pero, por el contrario, el documento original siempre hay que guardarlo junto a la firma.

En el caso de CAdES estas firmas se llaman firmas explícitas. En el caso de firmas XAdES XML, solo para las firmas despegadas (detached), el documento puede estar fuera.

12.3. FIRMAS CON MÚLTIPLES USUARIOS

En el mundo del papel y de la firma manuscrita, un documento puede contener la firma de varias personas:

- En un caso, las firmas pueden tener el mismo peso o valor legal, por lo que da igual el orden en el que se estampen las firmas en el documento.
- Otro caso, es el que unas firmas sirven para refrendar o certificar otras firmas anteriores, por lo que el orden en el que se estampan las firmas es importante.

El equivalente a esas firmas en el mundo electrónico son las firmas múltiples. Atendiendo al criterio del número de firmantes podemos tener:

- Firmas simples. Son las firmas básicas que contienen la firma de un solo firmante.
- Cofirma o firma en línea. Es la firma múltiple en la que todos los firmantes están al mismo nivel y en la que no importa el orden en el que se firma. La cofirma se utiliza en la firma de documentos que son resultados de reuniones, conferencias o comités.
- Contrafirma o firma en cascada. Firma múltiple en la que el orden en el que se firma es importante, ya que cada firma debe refrendar o certificar la firma del firmante anterior. La contrafirma se utiliza especialmente en aplicaciones como los Porta Firmas, en los que un documento debe seguir una línea específica a través de varios firmantes hasta que todo el proceso es aprobado.

12.4. FIRMAS LONGEVAS Y SELLO DE TIEMPO

Para verificar una firma es necesario: Comprobar la integridad de los datos firmados asegurando que éstos no hayan sufrido ninguna modificación.

Comprobar que el estado del certificado con el que se firmó era el correcto, es decir, era vigente en el momento de la operación.

En el caso de la firma electrónica básica, si el certificado está caducado automáticamente se da la firma como no válida.

Entonces, ¿cómo sabemos que el certificado estaba vigente o no en la fecha en la que se firmó? Y, ¿qué debe hacerse para que cuando se quiera validar o verificar una firma en el futuro la validación sea posible aunque esté caducado el certificado?

Para dar respuesta a estas preguntas, los formatos. AdES (forma genérica de llamar a los formatos CAdES, XAdES y PAdES) contemplan la posibilidad de incorporar a las firmas electrónicas información adicional que garantiza la validez de una firma a largo plazo, una vez vencido el período de validez del certificado.

Estos formatos añaden a la firma evidencias de terceros (de autoridades de certificación) y certificaciones de tiempo, que realmente certifican cuál era el estado del certificado en el momento de la firma.

Concretamente, existen distintos formatos de firma que van incrementando la calidad de la misma hasta conseguir una firma que pueda ser verificada a largo plazo (de forma indefinida) con plenas garantías jurídicas:

Firma Básica (AdES - BES), es el formato básico para satisfacer los requisitos de la firma electrónica avanzada.

- **AdEST**, se añade un sellado de tiempo (T de TimeStamp) con el fin de situar en el tiempo el instante en que se firma un documento.
- **AdESC**, añade un conjunto de referencias a los certificados de la cadena de certificación y su estado, como base para una verificación longeva (C de Cadena).
- **AdES X**, añade sellos de tiempo a las referencias creadas en el paso anterior (X de eXtendida).
- **AdES XL**, añade los certificados y la información de revocación de los mismos, para su validación a largo plazo (XL de eXtendido Largo plazo).
- **AdESA**, permite la adición de sellos de tiempo periódicos para garantizar la integridad de la firma archivada o guardada para futuras verificaciones (A de Archivo).

12.6. SELLO DE TIEMPO

Como se ha visto en el apartado anterior, el sellado de tiempo es un método para probar que un conjunto de datos existió antes de un momento dado y que ninguno de estos datos ha sido modificado desde entonces.

El Sello de Tiempo es una firma de una Autoridad de Sellado de Tiempo (TSA), que actúa como tercera parte de confianza testificando la existencia de dichos datos electrónicos en una fecha y hora concretos.

El sellado de tiempo proporciona un valor añadido a la utilización de firma digital, ya que la firma por sí sola no proporciona ninguna información acerca del momento de creación de la firma, y en el caso de que el firmante la incluyese, ésta habría sido proporcionada por una de las partes, cuando lo recomendable es que la marca de tiempo sea proporcionada por una tercera parte de confianza.

12.7. RESELLADO

Puesto que el Sello de Tiempo es una firma realizada con el certificado electrónico de la Autoridad de Sellado, cuando ese certificado caduca, el sello y, por tanto, la firma dejan de ser válidas.

Por eso, antes de que el certificado de la TSA caduque es necesario resellar o aplicar de nuevo el Sello Temporal para mantener la validez temporal de la firma.

12.8. HERRAMINETAS GRATUITAS PARA LA FIRMA ELECTRÓNICA BASADAS EN CERTIFICADO DE DOCUMENTOS

Las aplicaciones de Firma son los programas que permiten firmar un documento electrónico.

Existen algunos programas de uso cotidiano, por ejemplo, Adobe Acrobat o Microsoft Word, que permiten firmar el mismo documento que se genera. Sin embargo, este tipo de firma tiene dos inconvenientes:

No todos los programas que generan documentos son capaces también de firmarlos. En general, el destinatario del documento firmado deberá tener la misma aplicación para ser capaz de verificar la firma.

Las herramientas o aplicaciones específicas de firma electrónica son capaces de firmar cualquier tipo de documento electrónico y ayudan a superar los inconvenientes anteriores. Además se pueden descargar gratuitamente.

Presentamos 5 aplicaciones de firma que le ayudarán a firmar documentos PDF (y también otros formatos).

12.8.1. Firma fácil

Firma Fácil es una aplicación de firma realizada por el Ministerio de Hacienda y Administraciones Públicas. Su principal objetivo es ofrecer al usuario un sistema de firma en el que este pueda firmar cualquier tipo de documento de manera sencilla. El usuario indica qué fichero quiere firmar y la aplicación escoge automáticamente el formato de firma que debe aplicar, liberando así, al usuario de cualquier duda técnica.

12.8.2. Firma

La aplicación Firma, realizada por el Ministerio de Hacienda y Administraciones Públicas, es una aplicación java instalable en cualquier sistema operativo.

El Cliente Firma es una aplicación avanzada de firma que soporta la firma en múltiples formatos y permite la firma múltiple mediante la confirma y la contrafirma. Para mayor flexibilidad, Firma permite al usuario elegir el formato en el que desea firmar.

12.8.3. EcoFirma

Ecofirma, realizada por el Ministerio de Industria, es una aplicación de firma que permite generar y validar firmas electrónicas en formato XMLAdES.

12.8.4. Ksi ESecure

ESecure soporta tanto firma electrónica como el cifrado con contraseña o certificado para cualquier formato de fichero y decenas de algoritmos que van desde las opciones muy básicas como RC2 hasta opciones muy avanzadas como AES256.

ESecure dispone de varios modelos de licenciamiento: una versión gratuita para ciudadanos (licencia USER), una licencia EDU condicionada a la realización de un curso online gratuito y las versiones de pago PRO y PYME que se salen del alcance de esta serie por este motivo.

La versión USER soporta los formatos de firma PKCS#7/CMS/CAAdES (X-L), la versión EDU soporta además PDF/PAdES (basic y enhanced) y las versiones de pago permiten firmar en formato XMLDsig/XAdES (X-L) junto con funcionalidades adicionales como una funcionalidad de portafirmas electrónico o correo electrónico seguro S/MIME.

Las versiones gratuitas USER y EDU cuentan con limitaciones, algunas importantes. En el caso de la licencia USER:

- Soporta la firma de PDF solo para tamaños <1Mbyte, aunque permite emplear el formato PAdES.
- No soporta el trabajo con servidores OCSP para la consulta del estado revocación de los certificados, ni con servidores TSP para el sellado de tiempo.
- No soporta el trabajo con múltiples documentos (portafirmas)

En el caso de la licencia EDU elimina las limitaciones de los ficheros PDF, pero mantiene las demás limitaciones.

La interfaz se asemeja al explorador de ficheros Windows de XP (mantiene esta apariencia también en Windows 7). El conjunto completo de funcionalidades solo está disponible en las versiones de pago.

ESecure es compatible solamente con plataformas Windows. La aplicación se integra con el almacén de certificados de Windows y permite además importar certificados, soporta tanto ficheros PKCS#12, como librerías PKCS#11 para el uso de tarjetas criptográficas como el DNLe, soporta y dispone de una larga serie de posibilidades de configuración (con los límites según la licencia en cuestión).

Entre las principales ventajas de ESecure se pueden destacar la gran riqueza de su funcionalidad y número de estándares de firma y cifrado soportados que cubre incluso formatos aún no ampliamente soportados, como lo es PAdES.

La única desventaja es una política comercial bastante restrictiva que llega a limitar a las versiones de pago incluso en cosas básicas como la validación de los certificados vía OCSP y que la falta de soporte a las plataformas Mac OSy Linux.

12.8.5. XolidoSagra

XolidoSign es una herramienta completamente gratuita, sin limitaciones en su funcionalidad. Es decir, es 100% legal utilizarla tanto en el sector público como privado. El modelo de negocio de Xolido parece apostar en ese sentido por generar sus ingresos mediante su cartera de servicios en la nube y sus servicios de consultoría y auditoría.

13. DESCARGA, INSTALACIÓN Y EJECUCIÓN DE APLICACIONES UBICADAS EN SERVIDORES WEB.

Un sistema de gestión de paquetes, también conocido como “gestor de paquetes”, es una colección de herramientas que sirven para automatizar el proceso de descarga, instalación, actualización, configuración y eliminación de paquetes de software. El término se usa comúnmente para referirse a los gestores de paquetes en sistemas UNIX, especialmente GNU/Linux, ya que se apoyan considerablemente en estos sistemas de gestión de paquetes.

En estos sistemas, el software se distribuye en forma de paquetes, frecuentemente encapsulado en un solo fichero. Estos paquetes incluyen otra información importante, además del software mismo, como pueden ser el nombre completo, una descripción de su funcionalidad, el número de versión, el distribuidor del software, la suma de verificación y una lista de otros paquetes requeridos para el correcto funcionamiento del software. Esta meta información se introduce normalmente en una base de datos de paquetes local.

Sistemas de gestión de paquetes	Instalador
Forma parte del sistema operativo	Cada producto viene unido a sus propio instalador
Usa una única base de datos de instalación	Rastrea su propia instalación
Puede verificar y administrar todos los paquetes sobre el sistema	Solo trabaja con su propio producto
Un único vendedor de sistema de administración de paquetes.	Múltiples vendedores de instalador
Un único formato de paquetes	Múltiples formatos de instalación

Los sistemas de gestión de paquetes tienen la tarea de organizar todos los paquetes instalados en el sistema y se encargan de mantener su usabilidad. Esto se consigue combinando las siguientes técnicas:

Comprobación de la suma de verificación para evitar que haya diferencias entre la versión local de un paquete y la versión oficial.

Comprobación de la firma digital. Instalación, actualización y eliminación simple de paquetes.

Resolución de dependencias para garantizar que el software funcione correctamente.

Búsqueda de actualizaciones para proveer la última versión de un paquete, ya que normalmente solucionan bugs y proporcionan actualizaciones de seguridad.

Agrupamiento de paquetes según su función para evitar la confusión al instalarlos o mantenerlos.

Muchos de los sistemas de gestión de paquetes ampliamente utilizados utilizan backends simples para instalar los paquetes. Por ejemplo, YUM utiliza RPM como backend y APT utiliza dpkg.

En los sistemas donde las aplicaciones comparten módulos, como en la mayor parte de las distribuciones de GNU/Linux, la resolución de dependencias al instalar y desinstalar software se convierte en una necesidad. Algunos de los sistemas de gestión de paquetes más avanzados tienen la capacidad de desinstalar los paquetes recursivamente o en cascada, de forma que se eliminan todos los paquetes que dependen

del paquete a desinstalar y todos los paquetes de los que el paquete a desinstalar depende, respectivamente.

Es común que un administrador instale software que no está disponible en los repositorios provistos. Algunos ejemplos pueden ser una nueva versión de una aplicación que todavía no está disponible en la distribución o una alternativa distinta de la elegida por la distribución. Si este software adicional solo se distribuye en forma de código fuente, la instalación requerirá la compilación del código. Sin embargo, la instalación de este software adicional en el sistema ocasionará que el estado del sistema y la base de datos del gestor de paquetes no estén sincronizados, por lo que el administrador deberá tomar medidas adicionales para asegurar que el sistema de gestión de paquetes se mantenga al día, puesto que éste no es capaz de hacerlo automáticamente.

Alien es un programa que convierte entre los diferentes formatos de paquetes de GNU/Linux. Soporta la conversión entre Linux Standard Base, RPM, deb, Stampede (.slip) y paquetes de Slackware (.tgz).

Otra problemática aparte de la actualización de software es la actualización de ficheros de configuración. Ya que los sistemas de gestión de paquetes solo son capaces de sobrescribir o retener los ficheros de configuración, en lugar de poder aplicarles reglas de modificación. Sin embargo, hay excepciones, que normalmente se aplica al proceso de configuración del núcleo, ya que si estos son incorrectos pueden ocasionar fallos al reiniciar el sistema, pudiendo incluso hacer que el sistema no arranque. Estos problemas pueden ocasionarse cuando el formato de los ficheros de configuración cambia. Por ejemplo, cuando el antiguo fichero de configuración no deshabilita nuevas opciones que deberían ser deshabilitadas. Algunos sistemas de gestión de paquetes, como el `dipkg` de Debian, permiten configurar el software durante la instalación. En cualquier otra situación es preferible instalarlos paquetes con la configuración, por defecto, y sobrescribirla posteriormente.

El software normalmente se pone a disposición de los usuarios en los repositorios, con el fin de proporcionar a los usuarios de un sencillo control sobre los diferentes tipos de software que van a instalar en su sistema y, en ocasiones, debido a razones legales o conveniencias por parte de los distribuidores.

Cuando un usuario interactúa con el gestor de paquetes para realizar una actualización, éste suele mostrar una lista de las tareas a realizar (normalmente la lista de paquetes a actualizar y, posiblemente, también los números de versión) y también es probable que permita realizar una actualización completa o bien seleccionar los paquetes que se desea actualizar. Algunos gestores de paquetes permiten indicar los paquetes que no se desea actualizar nunca o solamente cuando estos corrigen errores importantes en la versión anterior. A este proceso se lo suele denominar versión pinning. Por ejemplo, `yum` permite esto mediante la sintaxis `exclude=openoffice`, `pacman` con `IgnorePkg=openoffice` (en ambos casos para evitar la actualización de OpenOffice), mientras que las herramientas de Debian poseen una sintaxis más compleja y potente.

13.1. SISTEMAS BASADOS EN PAQUETES BINARIOS

13.1.1. Dpkg

Usado originalmente por Debian y ahora también por otros sistemas, usa el formato `.deb`, y fue el primero en poseer una herramienta de resolución de dependencias ampliamente conocida, APT.

Comando	Acción
apt-get install <paquete>	Instalar un paquete.
apt-get remove <paquete>	Desinstalar un paquete.
apt-get purge <paquete>	Eliminar un paquete incluidos sus ficheros de configuración.
apt-get autoremove	Eliminar de forma automática aquellos paquetes que no se estén utilizando.
apt-get update <paquete>	Actualizar un paquete a la última versión disponible en el repositorio.
apt-get upgrade	Actualizar el sistema. Actualizará todos los paquetes que dispongan de una versión superior dentro de la rama instalada de la distribución.
apt-get dist-upgrade	Actualizar la distribución completa. Actualizará nuestro sistema a la siguiente versión disponible de la distribución.
apt-get source <paquete>	Descargar únicamente las fuentes de un paquete para manipularlas de forma manual.
apt-get clean apt-get autoclean	Limpiar cachés, paquetes descargados, etc.
apt-get check	Verificar que no tenemos ninguna dependencia incumplida.
apt-cache search <paquete>	Buscar un paquete en los repositorios, se puede especificar un patrón, expresión regular, el nombre exacto del paquete, etc.
apt-cache showdpkg <paquete>	Mostrar información sobre un paquete específico (nombre del paquete, versión, dependencias...).
apt-cache show <paquete>	Mostrar información del paquete incluyendo la descripción, información del paquete como su sitio web, página de bugs.
apt-cache depends <paquete>	Mostrar dependencias de un paquete.
apt-cache pkgnames	Mostrar los nombres de todos los paquetes instalados en el sistema.

13.1.2. Fink

Para Mac OS X, deriva parcialmente de dpkg/apt y de ports. Esta herramienta pretende hacer más sencilla la instalación de programas libres en Mac OS X

13.1.3. RPM

El sistema RPM, creado por Red Hat y usado por un gran número de distribuciones de GNU/Linux, es el formato de paquetes del Linux Standard Base. Para trabajar con este sistema de paquetes existen muy diversas herramientas como aptArpm, up2date (de Red Hat), urpmi (de Mandriva), YaST (de SuSE) y YUM (usado por Fedora Yellow Dog. Linux):

Comando	Acción
yum -y install paquete	Instala la última versión del paquete indicado. Instala sin pedir confirmación.
yum -y install paquete1 paquete2	Instala la última versión de los paquetes indicados, no hay límite de cuántos paquetes se pueden indicar. Instala sin pedir confirmación.
yum -y install paquete.arch	Instala la última versión del paquete indicado con la arquitectura indicada, por ejemplo: yum install mysql.i386
yum -y update	Actualiza todos los paquetes en el sistema.
yum -y update --exclude=sendmail	Actualiza todos los paquetes del sistema, excepto sendmail.
yum -y update httpd	Actualiza solo el paquete indicado, en este caso el servidor Web Apache.
yum -y update opera Firefox	Actualiza los paquetes indicados.
yum -y update --enablerepo=centosplus	Además de los repositorios que se tengan se habilita otro, en este caso "centos plus", esta opción también aplica para install".
yum -y upgrade	Actualiza los paquetes indicados, pero tomando en cuenta paquetes obsoletos en el cálculo de la actualización.
yum check-update	Muestra una lista de paquetes que necesitan ser actualizados sin instalarlos.
yum info paquete	Descripción completa del paquete indicado. Ejemplo: "yum info samba yum info recent". Muestra información resumida de los últimos paquetes instalados. O actualizados.
yum info available	Muestra información resumida de los paquetes disponibles a actualizarse.
yum list	Lista de todos los paquetes disponibles para instalación, actualización o ya instalados.
yum list grep mysql	Muestra solo los paquetes disponibles o ya instalados de mysql.
yum list installed	Lista de todos los paquetes instalados en el sistema.
yum list available	Lista de todos los paquetes disponibles para ser instalados.
yum list updates	Lista de todos los paquetes disponibles para ser actualizados.
yum remove telnet	Remueve el paquete indicado.
yum -y remove telnet vncserver	Remueve los paquetes indicados sin pedir confirmación.
yum search paquete	Busca el paquete " en la base de datos de paquetes instalados o para instalar. paquete" puede ser una palabra parcial del paquete a buscar.
yum clean headers	Elimina todos los archivos de encabezados que yum utiliza para resolver dependencias.
yum clean packages	Cuando utiliza la opción update" o "install" el paquete que se descarga e instala o actualiza no se elimina del sistema, ocupando espacio; con esta opción elimina esos paquetes.
yum clean all	Limpia tanto archivos de encabezados como paquetes, como utilizar las dos opciones previas, pero al mismo tiempo.
yum repolist	Lista los repositorios que se tengan de yum.

13.1.4. TGZ

El sistema tgz, usado por Slackware, empaqueta el software usando tar y gzip,. Pero, además, hay algunas herramientas de más alto nivel para tratar con este formato: slapt-get, slackpkg and suaret.

13.1.5. Pacman

Para Arch Linux usa binarios pre compilados distribuidos en un fichero pkg.taraz.

13.2. SISTEMAS DE METAPAQUETES

Los siguientes sistemas unifican la gestión de paquetes para muchas o todas las distribuciones de GNU/Linux y otras variantes de Unix basándose también en el concepto de ficheros-receta:

13.2.1. Klik

Proporciona una forma sencilla de instalar paquetes de software para la mayor parte de distribuciones sin los problemas de dependencias tan comunes en otros formatos de paquetes.

13.2.2. Autopackage

Usa fichero package.

13.2.3. Epm

Desarrollado por Easy Software Products (creadores de CUPS), es un meta empaquetador que permite crear paquetes nativos para todas las distribuciones de GNU/Linux y otros sistemas operativos basados en Unix (deb, rpm, tgz para GNU/Linux; pkg para Solaris y *BSD, dmg para Mac OS X.) a partir de un único fichero list.

Uno de los sistemas para la distribución de aplicaciones en Linux, aparte de los repositorios SVN de las aplicaciones es la web sourceforge.net. Es una central de desarrollos de software que controla y gestiona varios proyectos de software libre y actúa como un repositorio de códigos fuente. Fue creado en 1999. En el año 2013 giró hacia posiciones más mercantilistas, mediante inserciones aduare, aunque sigue manteniendo su esencia.

Su funcionamiento se basa en la sincronización de un proyecto con sourceForge además de gestionar las versiones de su código, pudiendo crear tiques, tareas pendientes, dar de alta varios desarrolladores para ese proyecto, etc. El proceso es el siguiente:

Debe crearse una cuenta en la web de sourceforge. Se registra un nuevo proyecto (nombre, la ruta de la web del proyecto en source...). Subir el código de la versión que tenga del proyecto.

Según el tipo de lenguaje e IDE que utilice la sincronización de su versión del proyecto con el que ha distribuido o es manual, o como en Java hay addons en Eclipse y en Netbeans que automatizan la sincronización de las sucesivas versiones que va obteniendo.

13.3. SISTEMAS PRIORITARIOS

En la actualidad, una gran variedad de sistemas de gestión de paquetes es usada por algunos sistemas operativos propietarios para tratar con la instalación tanto de paquetes propietarios como libres. De los que nos interesan tenemos el framework NET de Microsoft, un ensamblado es una biblioteca de código parcialmente compilado destinado al uso en deployment, versioning y seguridad.

13.4. GESTIÓN DE PAQUETES INCRUSTADA EN APLICACIONES

Algunos sistemas de gestión de paquetes no forman parte nativa del sistema operativo, como pueden ser fink en MacOSX o el entorno Unix-like de Cyguin (para Windows).

Algunos lenguajes de programación interpretados tienen su propio sistema de gestión de paquetes para manejar módulos del lenguaje, como pasa con los lenguajes de programación Perl (CPAN), Python (pip), PHP (PEAR) o Ruby (Ruby Gems). Otros programas pueden venir con su propio sistema para gestionar módulos.

Actividad

1. Descargue e instale Help Maker (<http://sourceforge.net/projects/helpmaker/>). Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
2. Descargue e instale Microsoft HTML Help Workshop (<http://www.microsoft.com/en-us/download/details.aspx?id=21138>). Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
3. Descargue e instale DocBuilder (<http://www.interactivetools.com/docs/docbuilder/>). Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
4. Descargue e instale Zscreen. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
5. Descargue e instale Greenshot. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
6. Descargue e instale el Lightshot. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
7. Descargue e instale PrtSrc. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.

Actividad

8. Descargue e instale Screenshot captor. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
9. Descargue e instale Jet Screenshot. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
10. Descargue e instale Screenpresso. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
11. Descargue e instale Aviscreen. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
12. Descargue e instale Camstudio. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
13. Descargue e instale Webinaria. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
14. Descargue e instale Krut. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
15. Descargue e instale Jing. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
16. Visite y navegue por la web de Screencast-o-matic. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
17. Visite y navegue por la web de ScreenToaster. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
18. Visite y navegue por la web de ScreenToaster. Haga capturas de todo el proceso, tanto de la descarga como de la instalación.
19. Visita la web de afirma (http://firmaelectronica.gob.es/Home/Empresas/Aplicaciones-Firma.html#arroba_firma), descargue el fichero (java) para poder realizar la firma con Ofirma y firme un documento de su propiedad. Haz capturas de todo el proceso seguido.
20. Visita la web de ecofirma (<http://firmaelectronica.gob.es/Home/Empresas/Aplicaciones-Firma.html#ecofirma>), descarga el fichero (java) para poder realizar la firma con ecofirma y firma un documento de su propiedad. Haz capturas de todo el proceso seguido.
21. Visita la web de KSI Esecure (<http://www.ksitdigital.com/web/productos.html>), descarga la aplicación para poder realizar la firma con KSI Esecure y firma un documento de su propiedad. Haz capturas de todo el proceso seguido.
22. Visita la web de XolidoSign (<http://www.xolido.com/lang/productosxolidosign/>), descarga la aplicación para poder realizar la firma con XolidoSign y firma un documento de su propiedad. Haz capturas de todo el proceso seguido.

Índice

Introducción	6
1. FICHEROS DE AYUDA. FORMATOS.....	7
1.1. PDF (O CUALQUIER FORMATO OFIMÁTICO).....	7
1.2. WINHELP (.HLP).....	7
1.3. CHM.....	7
1.4. MICROSOFT ASISTENCIA MARKUP LANGUAGE (MAML)	8
1.5. FICHEROS DE AYUDA EN LINUX.....	9
2. HERRAMIENTAS DE GENERACIÓN DE AYUDAS	10
2.1. HELPMAKER	10
2.2. MICROSOFT HTML HELP WORKSHOP	10
2.3. DOCBUILDER	10
2.4. WINHELP COMPILER	10
3. AYUDA GENÉRICA Y SENSIBLE AL CONTEXTO	11
4. TABLAS DE CONTENIDOS, ÍNDICES, SISTEMAS DE BÚSQUEDA, ENTRE OTROS.....	12
4.1. ARCHIVOS HHC (CONTENIDO).....	13
4.2. ARCHIVOS HHK (ÍNDICE)	14
4.3. ARCHIVOS H (DECLARACIÓN DE ID DE CONTEXTO)	15
4.4. ARCHIVOS HTM (CUERPOS DE LA AYUDA)	15
5. TIPOS DE MANUALES: MANUAL DE USUARIO, GUÍA DE REFERENCIA, GUÍAS RÁPIDAS, MANUALES DE INSTALACIÓN, CONFIGURACIÓN Y ADMINISTRACIÓN, DESTINATARIOS Y ESTRUCTURA	16
5.1. TIPOS DE DOCUMENTACIÓN	16
5.2. LA GUÍA TÉCNICA	16
5.2.1. Cuaderno de carga	16
5.2.2. Programa fuente.....	16
5.2.3. Pruebas.....	17
5.3. LA GUÍA DE USUARIO.....	17

5.4. LA GUÍA DE INSTALACIÓN	17
6. CONFECCIÓN DE TUTORIALES MULTIMEDIA.....	18
6.1. ¿ENTIENDE LAS NECESIDADES DE SU TARGET?	18
6.2. SEA BREVE, LA ATENCIÓN DISMINUYE RÁPIDO.....	18
6.3. ESPIAR A LA COMPETENCIA	18
6.4. PREPÁRESE ANTES DE APRETAR EL BOTÓN DE GRABAR.....	19
6.5. CONCLUSIÓN	19
7. HERRAMIENTAS DE CAPTURA DE PANTALLAS Y SECUENCIAS DE ACCIONES.....	20
7.1. ZSCREEN.....	20
7.2. GREENSHOT	20
7.3. LIGHTSHOT.....	21
7.4. PRSRC	22
7.5. SCREENSHOT CAPTOR.....	22
7.6. JET SCREENSHOT	23
7.7. SCREENPRESSO.....	23
8. HERRAMIENTAS PARA LA CONFECCIÓN DE TUTORIALES INTERACTIVOS, SIMULACIÓN	24
8.1. VERSIÓN ESCRITORIO	24
8.1.1. Aviscreen	24
8.1.2. CamStudio.....	24
8.1.3. Webinaria	25
8.1.4. Krut.....	25
8.1.5. Jing.....	26
8.2. VERSIONES WEB	26
8.2.1. Screencast-O-Matic.....	26
8.2.3. ScreenToaster.....	27
9. COMPONENTES DE UNA APLICACIÓN. EMPAQUETADO	28
9.1. LINUX	28

9.2. WINDOWS	28
9.3. JAVA	29
10. INSTALADORES. PAQUETES DE INSTALACIÓN. HERRAMIENTAS PARA CREAR PAQUETES DE INSTALACIÓN. PARÁMETROS DE LA INSTALACIÓN.....	30
10.1. WINDOWS	30
10.2. LINUX	31
10.2.1. Compilar el paquete	31
10.3. PAQUETES .DEB Y .RPM	31
10.3.1. Summary, Name y Version.....	33
11. ASISTENTES DE INSTALACIÓN Y DESINSTALACIÓN. PAQUETES AUTOINSTALABLES	35
11.1. WINDOWS	35
11.2. LINUX	36
12. FICHEROS FIRMADOS DIGITALMENTE	37
12.1. ESTRUCTURA DE LA FIRMA.....	37
12.1.1. CAdES (CMSAvanzado).....	37
12.1.2. XAdES (XML, Avanzado)	37
12.1.3. PAdES (PDF Avanzado)	37
12.1.4. OOXML y ODF	37
12.2. ¿DÓNDE SE GUARDA EL DOCUMENTO ORIGINAL?	37
12.2.1. El documento original se incluye en el fichero de firma.....	38
12.2.2. El documento no se incluye en la firma	38
12.3. FIRMAS CON MÚLTIPLES USUARIOS.....	38
12.4. FIRMAS LONGEVAS Y SELLO DE TIEMPO	39
12.6. SELLO DE TIEMPO	39
12.7. RESELLADO.....	40
12.8. HERRAMINETAS GRATUITAS PARA LA FIRMA ELECTRÓNICA BASADAS EN CERTIFICADO DE DOCUMENTOS	40

12.8.1. Firma fácil	40
12.8.2. Firma	40
12.8.3. EcoFirma.....	40
12.8.4. Ksi ESecure	41
12.8.5. XolidoSigra	41
13. DESCARGA, INSTALACIÓN Y EJECUCIÓN DE APLICACIONES UBICADAS EN SERVIDORES WEB.	42
13.1. SISTEMAS BASADOS EN PAQUETES BINARIOS.....	43
13.1.1. Dpkg	43
13.1.2. Fink	44
13.1.3. RPM.....	44
13.1.4. TGZ	46
13.1.5. Pacman.....	46
13.2. SISTEMAS DE METAPAQUETES	46
13.2.1. Klik.....	46
13.2.2. Autopackage	46
13.2.3. Epm.....	46
13.3. SISTEMAS PRIORITARIOS	47
13.4. GESTIÓN DE PAQUETES INCRUSTADA EN APLICACIONES	47
Índice	49