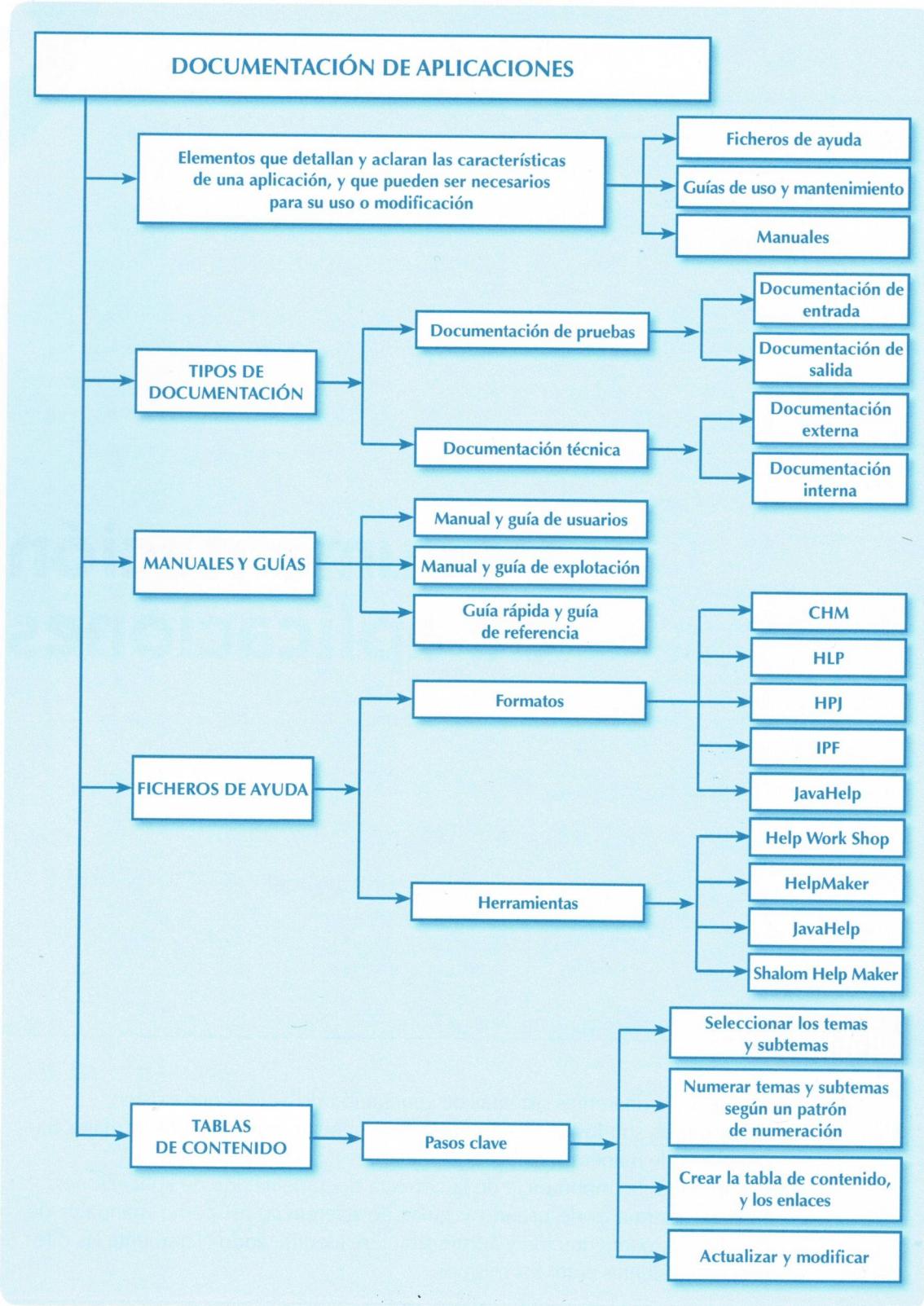


Mapa conceptual



Glosario

Documentación de pruebas. Documentación que incluye información relacionada con los requisitos y procedimientos de los escenarios de pruebas, así como de los informes resultantes de su realización.

Documentación. En el ámbito de las aplicaciones software, nos referimos a todos los elementos que detallan y aclaran las características de una aplicación y que pueden ser necesarios para su uso o modificación.

Documentación técnica. Contiene información de carácter técnico relacionada con una aplicación concreta, en la que se detallan las especificaciones y requisitos del sistema, guía de uso, etc. Está formada habitualmente por guías, hojas de especificaciones, manuales, etc.

Fichero de ayuda. Documento que contiene información de ayuda sobre un determinado campo. Un ejemplo de fichero de ayuda podría ser un manual de uso de una herramienta concreta, dirigido a los usuarios de la misma.

Guía de referencia. Documentación que está enfocada a usuarios que tienen un mayor nivel de conocimiento sobre el uso de la aplicación, así como una mayor experiencia.

Guía rápida. Las guías rápidas aportan información muy concreta, relacionada con diferentes procedimientos o campos de una aplicación.

JavaHelp. Es a la vez una herramienta para generar ficheros de ayuda y un fichero de ayuda en sí mismo. Está orientado a la documentación de aplicaciones desarrolladas en Java.

Manual de usuario. Contiene toda la información necesaria con el fin de hacer más fácil al usuario el uso y comprensión del programa. Los ámbitos de aplicación son varios y diversos: formación del usuario, guía de consulta ante dudas, ayuda para detectar y corregir errores.

8.1. ¿Por qué es importante documentar?

Cuando se habla de documentación de aplicaciones software nos referimos a todos los elementos que detallan y aclaran las características de una aplicación, y que pueden ser necesarios para su uso o modificación. En concreto, se trata de ficheros de ayuda, manuales y demás guías de uso y/o mantenimiento, ya sean dirigidas para los usuarios de la aplicación, para equipos de soporte o incluso para otros diseñadores.

Para un sistema software es especialmente importante contar con una documentación completa y que sea sencilla de consultar. Los motivos para ello son varios:

- Facilitar su comprensión y posibilitar su uso por parte del usuario, independientemente de cuál sea su perfil.
- Facilitar el mantenimiento de la aplicación.

- Facilitar el desarrollo de mejoras por otros equipos de trabajo.
- Facilitar que el sistema pueda ser comprendido por otros diseñadores.

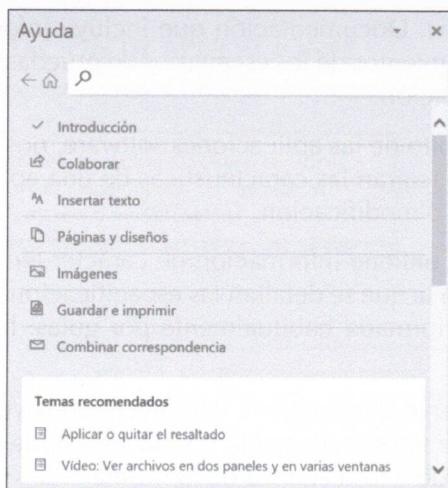


Figura 8.1
Ejemplo de menú de ayuda
de Microsoft Word.

A pesar de tratarse de un trabajo que puede resultar tedioso, se le debe otorgar toda la importancia que tiene. No es suficiente con desarrollar aplicaciones funcionales y atractivas, sino que, además, estas deben contar con una documentación adecuada. Se trata de una tarea tan importante como la de realizar el propio código del programa.

Actividad propuesta 8.1



En tu opinión, ¿crees que la documentación de aplicaciones informáticas puede ser una tarea importante para el éxito de las mismas?

8.2. Tipos de documentación

A grandes rasgos, podemos dividir la documentación de aplicaciones software en dos grupos: documentación de pruebas y documentación técnica.

8.2.1. Documentación de pruebas

Documentar las pruebas realizadas sobre un programa determinado es fundamental para detectar y corregir posibles errores. Podemos distinguir a su vez dos tipos:

- a) *Documentación de entrada:* en la que se especifican los escenarios de prueba y se detallan los procedimientos de las mismas.
- b) *Documentación de salida:* se trata de los informes resultantes de aplicar las pruebas sobre el programa definidas en el punto anterior.

Estos informes recogerán el histórico de pruebas realizadas, documentándose así cada problema que haya podido surgir y su posterior corrección.

8.2.2. Documentación técnica

Pertenece a este grupo el resto de documentación: guías, hojas de especificaciones, manuales. Al igual que en el caso anterior, puede dividirse en dos grupos:

- a) *Documentación interna:* se trata de los comentarios incluidos por el desarrollador en el código, que deben describir distintos aspectos sobre el mismo.

Se debe incidir en la importancia de realizar un programa ordenado y claro, en el que los comentarios ayuden a entender el código, pero este ya de por sí debe ser claro. En este ámbito, debe tenerse en cuenta lo siguiente:

- Incluir comentarios al comienzo de módulos.
- Comentar variables, contestantes, procedimientos y funciones.
- Incluir comentarios introductorios sobre bloques de código, funciones o módulos.
- No comentar lo obvio, ya que el exceso de comentarios puede ser contraproducente.

- b) *Documentación externa:* en este caso, suele tratarse de un manual técnico (orientado a programadores para facilitar el mantenimiento y desarrollo de la aplicación a futuro) y de un manual y/o guía de usuario (para facilitar su uso).



Actividad propuesta 8.2

¿Podrías indicar algún ejemplo que conozcas de cada uno de los tipos de documentación indicados anteriormente?

8.3. Manuales y guías

Existen diferentes tipos de manuales y guías, diferentes entre sí en función del contenido de



TOMA NOTA

En el momento de realizar un manual o guía es muy importante pensar como lo haría el usuario que utilice la aplicación para entender qué problemas puede encontrarse y de qué forma podemos ayudarle a resolverlos. Este es el fin principal de los manuales y las guías de usuario.

8.3.1. Manual y guía de usuario

El *manual de usuario* contiene toda la información necesaria con el fin de hacer más fácil el uso y la comprensión del programa. Los ámbitos de aplicación son varios y diversos: formación del usuario, guía de consulta antes dudas, ayuda para detectar y corregir errores, etc.

Es cierto que no existe ninguna norma acerca de cómo se debe elaborar, aunque hay ciertos patrones y usos que se han de seguir. En primer lugar, el documento en sí mismo debe ser claro y atractivo desde el punto de vista del usuario, que a la postre es quien va a utilizarlo. Para ello, puede resultar aconsejable utilizar imágenes y gráficos para ayudar a entender el texto, utilizar un vocabulario claro y conciso, ejemplos prácticos, etc.

Una posible estructura del manual de usuario puede ser la siguiente:

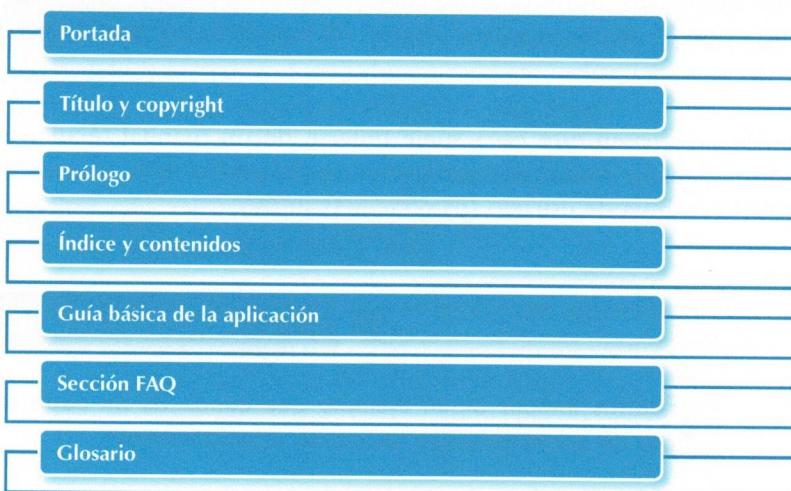


Figura 8.2

Possible estructura de manual de usuario de una aplicación.

RECUERDA

- ✓ La estructura de un manual de usuario dependerá en gran medida del objetivo que se pretenda conseguir, así como del tipo de usuario que vaya a utilizarlo.

8.3.2. Manual y guía de explotación

El *manual y guía de explotación* contiene la información necesaria para utilizar y explotar la aplicación. Es decir, va muy orientado a la instalación, configuración y puesta en marcha, por lo que, evidentemente, irá muy ligado al contexto en el que se vaya a realizar la instalación (tipo de organización, requerimientos, número de usuarios).

Aunque en este caso tampoco hay una norma para su elaboración, debe contener al menos los siguientes puntos:

- ✓ Requerimientos mínimos del sistema en cuanto a procesador, memoria, espacio libre.
- ✓ Proceso de instalación:
 - Necesidad o no de preparación previa, si es necesaria alguna tarea para ello (por ejemplo, habilitar la conexión a Internet, activar o desactivar permisos).
 - Opciones de instalación: a través de USB, red, etc.
 - Procedimiento de instalación paso a paso, detallando y explicando sus diferentes opciones.
- ✓ Actualizaciones del sistema y copias de seguridad.
- ✓ Glosario.
- ✓ Índice.

En función de la complejidad de la aplicación, puede dividirse en manual de instalación y manual de configuración.

8.3.3. Guía rápida y guía de referencia

De manera adicional a los manuales estudiados anteriormente, la documentación de una aplicación puede completarse con este tipo de guías, en función del caso en concreto. Sus características son las siguientes:

1. *Guía rápida*: se orienta a usuarios del sistema y/o encargados de mantenimiento. En función de la complejidad de la aplicación puede ser necesario realizar varias, cada una con distinta temática. Las guías rápidas aportan información muy concreta y muy diferente, relacionada con diversos procedimientos o campos de una aplicación.
2. *Guía de referencia*: al contrario que en el caso de las guías rápidas, estas suelen estar pensadas para usuarios que tienen un mayor nivel de conocimiento sobre el uso de la aplicación, así como una mayor experiencia. Por ello, es habitual que contengan información relacionada con aspectos más técnicos: tipos de mensajes de error y su causa, tipos de datos de entrada que le son permitidos a la aplicación, tipos de comandos, etc.



Actividad propuesta 8.3

Obtén información y ejemplos sobre diferentes guías rápidas y de referencia de algunas aplicaciones que conozcas.

8.4. Ficheros de ayuda. Formatos. Herramientas para generarlos

Un *fichero* es un elemento que puede contener varios tipos de información en diferentes formatos, ya sea en soporte físico o en soporte digital, que suele ser el procedimiento habitual en el campo de las aplicaciones informáticas.

De esta forma, un *fichero de ayuda* es un documento que contiene información de ayuda sobre un determinado campo. Un ejemplo de fichero de ayuda podría ser un manual de uso de una herramienta concreta, dirigido a los usuarios de la misma. Los ficheros de ayuda están formados por dos partes diferenciadas:

- *Mapa del fichero*: se trata de un apartado que facilita la navegación por el fichero, que identifica claramente el contenido de este a través de identificadores.
- *Vista de información*: muestra al usuario contenidos en forma de glosario o tabla de contenidos.

8.4.1. Formatos

En el siguiente cuadro se muestran algunos de los principales formatos de ficheros de ayuda, junto con las principales características de estos:

CUADRO 8.1

Comparativa de algunos de los principales formatos de ficheros de ayuda existentes

Formato	Características	Plataforma
CHM	<ul style="list-style-type: none"> – Generado a partir de HLP. – Utiliza HTML para generar la ayuda. – Enlaces mediante hipervínculos a la tabla de contenido. – Permite fusionar varios ficheros de ayuda. – Puede ser creado a partir de HTML Help Workshop. 	Microsoft Windows
HLP	<ul style="list-style-type: none"> – Puede incluir tabla de contenido en fichero .cnt. – Incluye información extra en fichero .gid. – Utiliza ficheros RTF para generar la ayuda. – Necesita compilación (por ejemplo mediante HTML Help Workshop). 	Windows
HPJ	<ul style="list-style-type: none"> – Contiene tabla de contenido (fichero .cnt). – Se crea mediante herramienta tipo Help Workshop. 	Windows
IPF	<ul style="list-style-type: none"> – Similar a HTML. – Se utiliza para ayuda en línea o web. – Es necesario compilar para generarlo. 	Web (ayuda en línea)
JavaHelp	<ul style="list-style-type: none"> – Implementado en Java, por lo que es utilizado habitualmente para implementar la ayuda de aplicaciones desarrolladas en este lenguaje. 	Varias

RECUERDA

- ✓ La elección del formato depende de cómo se implemente la ayuda de la aplicación en sí misma. Es importante tener claro este aspecto previamente a la realización del contenido para intentar optimizarlo.

8.4.2. Herramientas para generar ficheros de ayuda

A continuación, se detallan las principales características de algunas herramientas que se pueden utilizar para generar ficheros de ayuda:

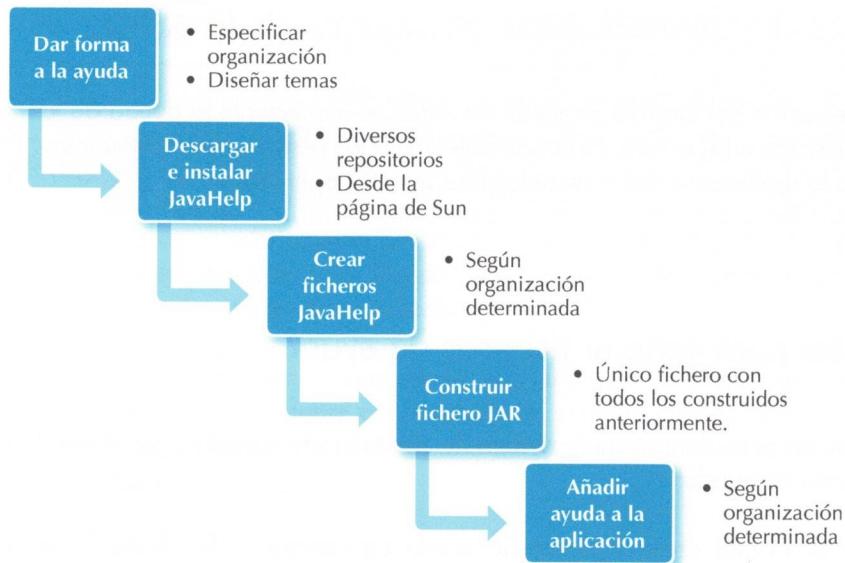
- ✓ *Help Work Shop*: posibilita crear ficheros de ayuda en entornos Windows. Es una aplicación que consta de un editor de imágenes, un administrador de proyectos y un compilador.
- ✓ *HELPMAKER*: es una herramienta de carácter gratuito que cuenta con múltiples opciones para personalizar los ficheros. Utiliza un editor de texto para facilitar la generación de los ficheros de ayuda.
- ✓ *JavaHelp*: JavaHelp por sí mismo es un sistema de creación de ficheros de ayuda y, además, un formato de ficheros de ayuda propiamente dicho. Se caracteriza por:
 - Utilizar Java para su implementación.
 - Utilizarse en aplicaciones Java.
 - Ser independiente de la plataforma.
 - Facilitar la realización de ayuda online.
- ✓ *SHALOM HELP MAKER*: se trata de otra herramienta gratuita, que utiliza un editor de texto para la creación de ficheros de ayuda, de manera similar a HELPMAKER. Permite incluir enlaces externos, insertar o crear imágenes, crear índices, etc.

**Actividad propuesta 8.4**

Obtén información acerca de las herramientas más utilizadas y extendidas para generar ficheros de ayuda. Puedes emplear para ello buscadores vía web.

8.4.3. Generación de un sistema de ayuda con JavaHelp

Para crear un sistema de ayuda con JavaHelp se deben seguir unos pasos concretos, indicados en el siguiente diagrama:

**Figura 8.3**

Pasos necesarios para generar un sistema de ayuda con JavaHelp.

En el siguiente cuadro vemos los ficheros JavaHelp y para qué se utilizan:

CUADRO 8.2**Ficheros JavaHelp y su uso**

Fichero	Extensión	Utilización
Índice	XML	Incluye la distribución del sistema de ayuda.
Map	JHM	Asocia elementos (imágenes, ficheros HTML, etc.) del fichero HTML con un identificador.
Helpset	HS	Contiene la información necesaria para que el sistema de ayuda se ejecute.
Tabla de contenidos	XML	Incluye el contenido de la ayuda y su distribución.
Temas o topics	HTML	Para crearlos se puede utilizar cualquier herramienta para generar HTML. Contienen la información de ayuda como tal, debiéndose realizar uno por cada tema o asunto de ayuda. Se muestra uno de ellos por pantalla. Deben tener organización jerárquica.
Base de datos de búsqueda	N/A	Se debe utilizar la herramienta jhindexer para generarla.

El directorio con los ficheros creados debe seguir una estructura similar a la siguiente:

```

/help
  - Índice.xml
  - Map.jhm
  - Helpset.hs
  - TablaDeContenidos.xml
/html
  - Tema1.html
  - Tema2.html
/JavaHelpSearch

```

Figura 8.4
Directorio de ficheros JavaHelp.

Es fundamental que los ficheros HTML estén en el directorio ./help, ya que, en caso contrario, la aplicación no funcionará.

Para generar fichero JavaHelp de prueba vamos a seguir paso a paso lo indicado anteriormente. Para visualizar un ejemplo, nos crearemos una ayuda ficticia basada en tres páginas HTML:

- Página principal a modo de introducción.
- Página con la ayuda necesaria para el “tema 1”.
- Página con la ayuda necesaria para el “tema 2”.

El resultado final será el siguiente:



Figura 8.5
Ejemplo de ayuda creada con JavaHelp.

Veamos el procedimiento paso a paso.

1. Dar forma a la ayuda

En este primer paso, construiremos la ayuda de nuestra aplicación, creando tantos ficheros HTML como necesitemos. Es importante construir adecuadamente la estructura de la ayuda,

ya que es fundamental que sea clara y concisa para facilitar la localización de la información por parte del usuario.

En nuestro ejemplo, al tratarse de un caso sencillo, únicamente hemos creado dos temas de ayuda, junto a una página principal.

El código HTML del fichero principal es el siguiente:

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8 />
<title>Esta es nuestra ayuda JavaHelp de prueba </title>
</head>
<body>

<h1>Esta es una ayuda creada con JavaHelp</h1>

<h2>Esta ayuda la hemos creado a modo de prueba</h2>

<p>Para ello podemos utilizar un generador de código HTML cualquiera<h2></h2>
<p>Podemos también añadir imágenes o incluso referenciar
entre sí las diferentes páginas de la ayuda<h2></h2>



<p></p>

Este es el enlace al

<a href="tema1.html">Tema 1</a> de ayuda

<p><a>Este es el enlace al </a>
    <a href="tema2.html">Tema 2</a> de ayuda

</body>
</html>
```

2. Descargar e instalar JavaHelp

Los ficheros necesarios para poder trabajar con JavaHelp los podemos encontrar en diversos repositorios. Es recomendable descargarlos desde un sitio seguro y oficial, como puede ser la página de Sun Microsystems.

Una vez descargados, seguiremos el proceso de instalación hasta completarlo.

3. Crear ficheros JavaHelp

- *Índice.* Será un fichero con extensión .xml en el que se detallarán todas las páginas que tiene nuestra ayuda (tres en nuestro ejemplo). El fichero tendrá la siguiente apariencia: se compone de una etiqueta principal llamada <index>, que a su vez contiene varias subetiquetas llamadas <indexitem>, que hacen referencia a cada elemento de ayuda que compondrá el Índice. De esta forma, se mostrará el texto que se incluya en “text”, mientras que el campo “target” hace referencia a la clave o ID. Esta clave debe coincidir con la indicada en el fichero map dentro del mismo campo “target”.
- *Fichero map.* En este caso, hemos creado un “mapID” por cada fichero HTML, con el fin de poder referenciarlo. Evidentemente, se debe indicar la ruta en la que está dicho fichero. El fichero se puede crear con cualquier editor de texto, salvándolo en formato .jhm.
- *Helpset.* Se podría decir que es el fichero principal, que debe construirse en formato .xml, aunque se guardará con extensión .hs.

Un ejemplo de fichero HelpSet para nuestro caso puede ser el siguiente, aunque habría diversas posibilidades: el contenido del fichero se encuentra dentro de la etiqueta <helpset> que define el principio y final del mismo. A partir del comienzo, podemos definir tres tipos de etiquetas:

<title>: será el título que aparezca en la ventana de ayuda.
 <maps>: indica la ubicación del fichero map.
 <homeID>: fichero que aparece por defecto al ejecutar la ayuda.
 <mapref>: nombre del fichero map.
 <view>: uno por cada elemento de la ayuda. En este caso, Búsqueda, Índice y Tabla de Contenidos.

El orden de los mismos en el fichero determina cómo aparecen en la ayuda:

<name>: nombre del view.
 <label>: etiqueta que se mostrará en relación con este elemento.
 <type>: muestra la clase de JavaHelp relacionada con el contenido.
 <data>: indica el fichero que está relacionado con esa parte, de los que se crearon anteriormente.

En este caso, en el campo de Búsqueda, se incluye el subdirectorio que se crea a partir de la herramienta jhindexer.jar, como se verá a continuación.

- *Tabla de contenidos.* Detallará la organización de los ficheros de nuestra ayuda, que en nuestro ejemplo es muy sencilla. Deberá guardarse en formato .xml, con cualquier editor de texto. La etiqueta <toc> describe el elemento en su conjunto, que a su vez se divide en varios <tocitem>. Estos elementos pueden anidarse entre sí. Dentro de los mismos, de igual forma que en el fichero Índice, se mostrará el texto que se incluya en “text”. Así mismo, el “target” hace referencia a la clave o ID, que debe coincidir con la indicada en el fichero map.
- *Base de datos de búsqueda.* Para crearla es necesaria la ejecución del comando jhindexer.jar, que invoca a una de las herramientas de JavaHelp.

Para ello, basta con ejecutar la orden “jhindexer.jar html”, cambiando previamente el PATH al directorio en el que tenemos los ficheros de ayuda, es decir, el directorio

help. El parámetro html hace referencia al directorio en el que están almacenados los ficheros de ayuda.

La creación de esta base de datos permitirá contar con una pestaña de búsqueda, que hará posible navegar por la ayuda de una manera más ágil y eficaz.

4. Construir fichero JAR

Para construir los ficheros JAR será necesario ejecutar: Jar -cvf help.jar. Mediante esta orden se crean ficheros JAR independientes. Se debe ejecutar desde el directorio help en el que están los ficheros de ayuda.

5. Añadir ayuda a la aplicación

Para añadir la ayuda que hemos creado a cualquier aplicación Java, se deben incluir los siguientes apartados en el mismo:

Paquete javax.help.*

Clase HelpSet, que posibilita el uso de ficheros y datos de la ayuda.

Clase HelpBroker, que ayuda a visualizar el contenido de la ayuda.

Método createHelpBroker.

Método findHelpSet.

Método enableHelpOnButton, para poder visualizar la ayuda al pulsar un menú.

Método enableHelp, que al hacer clic sobre un elemento indicará el tema de ayuda.

Método enableHelpKey, que activa la tecla de ayuda sobre un elemento.

8.4.4. Ayuda genérica y sensible al contexto

Aunque a priori pudiera parecer una distinción evidente, debemos tener en cuenta las diversas formas en las que puede presentarse y/o elaborar una documentación de ayuda relacionada con una aplicación:

CUADRO 8.3

Características de los diferentes tipos de ayuda

Ayuda genérica	Ayuda sensible al contexto
<ul style="list-style-type: none"> - Contiene toda la información de ayuda sobre una determinada aplicación, dividida por temas o campos. - Generalmente presenta un índice o menú. - Permite navegar o buscar entre los elementos que la forman. 	<ul style="list-style-type: none"> - Se trata de ayuda particularizada para un tema o acción concreta. - Puede presentarse de diferentes maneras: ayuda en línea, breve detalle del uso de un menú en el programa.

8.5. Tablas de contenidos

Las tablas de contenidos nos ayudan a modelar tanto el contenido como la estructura de un documento determinado. La información estará organizada en forma de esquema, con diversos niveles entre los que se cabe destacar títulos y subtítulos. Lógicamente, el diseño de las mismas depende de quién haga el desarrollo, así como de otros muchos factores (objetivo, usuario al que se dirigen, etc.). Sin embargo, existe una serie de características comunes a todas ellas:

- ✓ Es posible o no que muestren el número de páginas que contienen, depende del diseño que se realice sobre las mismas. Se trata de un elemento completamente opcional.
- ✓ Las tablas de contenidos habitualmente contienen enlaces directos a cada título o subtítulo, que apunta al contenido de los mismos. Así se consigue una navegación y utilización más fluida y sencilla para el usuario.
- ✓ Suelen situarse al comienzo de cada documento, de manera similar al índice de los libros.
- ✓ Aunque puede resultar sencillo, su proceso de elaboración no es elemental, ya que es necesario realizar un análisis previo y pormenorizado de toda la documentación que contienen. Además, es necesario que los títulos sean lo más claros e intuitivos posible. Es también importante que no haya información duplicada en varias partes del documento.

Como resumen, en la siguiente figura se muestran algunos consejos que se han de tener en cuenta para elaborar una tabla de contenidos clara y de calidad, que sea útil para el usuario:

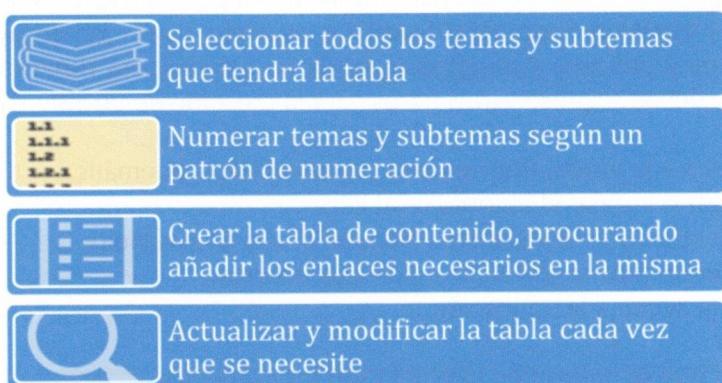


Figura 8.6

Consejos para elaborar una tabla de contenidos de calidad.



Actividad propuesta 8.5

¿Cuáles son las principales características que debe cumplir una tabla de contenidos? ¿Podrías indicar algún ejemplo real que conozcas?

Resumen

- Tal y como hemos estudiado en este capítulo, es muy importante documentar de forma adecuada cada aplicación o programa que se realice. Es decir, el desarrollo de cualquier aplicación (también el desarrollo de interfaces gráficas) implica la elaboración de manuales y sistemas de ayuda al usuario, de modo que se establezca una documentación completa sobre el funcionamiento de la misma. Estos sistemas de ayuda resultan clave tanto para un usuario básico como para usuarios con mayor experiencia, y pueden ayudar al éxito de la aplicación en sí.
- Para lograr esa tarea de elaborar una documentación completa y atractiva, se pueden utilizar diferentes tipos de documentaciones, que se pueden adaptar a cada caso concreto y a cada aplicación. A grandes rasgos, podemos diferenciar un primer tipo de documentación denominada "de pruebas" de otro tipo de documentación denominada "técnica". La documentación técnica quedará dividida a su vez en documentación de entrada y documentación de salida, mientras que la de tipo técnico se subdivide en documentación interna y documentación externa.
- La documentación de ayuda suele estar formada por *ficheros de ayuda*. Un ejemplo de fichero de ayuda podría ser un manual de uso de una herramienta concreta, dirigido a los usuarios de la misma. Los ficheros de ayuda están formados por el mapa del fichero y por la vista de información.
- En cuanto a los formatos de los ficheros de ayuda, algunos de los más comunes son los siguientes: CHM, HLP, HPJ, IPF y JavaHelp.
- Existen multitud de herramientas que nos ayudan a generar ficheros de ayuda, como pueden ser Help Work Shop, HELPMAKER o JavaHelp. En el caso de JavaHelp, es una aplicación muy utilizada para generar ayuda en aplicaciones desarrolladas en Java. Para su uso, deben seguirse varios pasos para la creación de un sistema de ayuda:
 1. Dar forma a la ayuda, indicando la organización y la temática.
 2. Descargar e instalar el entorno JavaHelp.
 3. Creación de los ficheros necesarios para el uso de JavaHelp.
 4. Construir un fichero JAR utilizando estos ficheros.
 5. Añadir la ayuda a la aplicación.
- Los ficheros que se utilizan en JavaHelp son los siguientes:
 - Índice.
 - Map.
 - HelpSet.
 - Tabla de contenidos.
 - Temas o topics.
 - Base de datos de búsqueda.
- Por último, las tablas de contenido permiten indicar la estructura y el contenido de un documento determinado. Habitualmente, están estructuradas en diferentes niveles de elementos, entre los que cabe destacar los títulos y los subtítulos.

Ejercicios propuestos



1. Como hemos visto en este capítulo, un sistema de ayuda sirve para detallar las características de una aplicación e indicar qué elementos pueden ser necesarios para su uso, mantenimiento o modificación. La clave se encuentra en realizar una documentación completa y que a la vez sea sencilla de consultar.

En este caso se utilizará la herramienta JavaHelp para la inclusión de un sistema de ayuda. Esta aplicación permite la creación de sistemas de ayuda para ser integrados posteriormente en un programa desarrollado en lenguaje Java. Será necesario elaborar cuatro ficheros para desplegar el sistema de ayuda: map_file.jhm, toc.xml, index.xml y help_set.hs.

- Primero se debe especificar la organización y diseñar los temas de la ayuda.
- Después será necesario instalar JavaHelp.
- Crear los ficheros de JavaHelp necesarios. En este caso será necesario implementar cuatro ficheros llamados:
 - Fichero 1: map_file.jhm
 - Fichero 2: toc.xml
 - Fichero 3: index.xml
 - Fichero 4: help_set.hs

Los tres primeros ficheros son los referenciados desde el archivo hep_set.hs. Este documento relaciona todo el sistema de ayuda, incluyendo el índice y la tabla de contenidos.

Vamos a construir los ficheros JavaHelp necesarios para documentar una aplicación llamada "Calculadora", que se encarga de realizar las cuatro operaciones básicas: suma, resta, multiplicación y división.

- Continuando con los pasos del ejercicio 1, construye un fichero JAR que incluya todos los ficheros y permita mejorar su distribución. Después añade la ayuda a la aplicación "Calculadora" desarrollada en temas anteriores.
- Si tuvieras que diseñar una guía / manual de usuario de esta aplicación "Calculadora", ¿cómo lo construirías? ¿Qué puntos contendría?

ACTIVIDADES DE AUTOEVALUACIÓN

- La parte del fichero de ayuda que se muestra al usuario, habitualmente en forma de índice, glosario, tabla de contenido e incluso buscador de temas, es:
 - a) La guía de inicio.
 - b) El mapa de navegación.
 - c) El mapa de fichero.
 - d) La vista de información.