

DESARROLLO DE INTERFACES

TÉCNICO EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA

**Documentación de aplicaciones: Ayudas**

**JavaHelp**

Cuando se habla de documentación de aplicaciones software, nos referimos a todos aquellos elementos que detallan las características de una aplicación, y que pueden ser necesarios para su **uso, mantenimiento o modificación**.

En concreto, se trata de ficheros de ayuda, manuales y demás guías de uso y/o mantenimiento, ya sean dirigidos 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 a la vez sea sencilla de consultar. Los motivos para ello son varios, por ejemplo, conseguir la mejora de la comprensión de uso de una aplicación por parte de un usuario, independientemente de su perfil.

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 que puede realizar el propio código del programa.

## Ficheros de ayuda y formatos

Un **fichero de ayuda** es un documento que contiene toda la información que puede servir de ayuda o de manual para los usuarios de la herramienta sobre la que se ha desarrollado.

Estructura general de cualquier fichero de ayuda:

Suelen estar compuestos por **dos partes claramente diferenciadas**:

- **Mapa de fichero:**  
Constituye una especie de **mapa de navegación** del sistema de ayuda, **asociando identificadores** para cada uno de los temas contenidos a las **URL** donde se encuentra el contenido relativo al tema seleccionado.
- **Vista de información:**  
Es la parte que **se muestra al usuario**, habitualmente en forma de **índice, glosario, tabla de contenidos** e incluso **buscador** de temas.

**Formatos digitales para la creación** y posterior consulta de **ficheros de ayuda**:

- **HLP**

Actualmente, este tipo de formato de fichero de ayuda **se encuentra en desuso**, fue **sustituido por CHM**. Se trata de los ficheros utilizados habitualmente para la generación de ayuda de **Windows**.

- 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**)

- **CHM** (Ayuda **HTML Compilado**)

Este nuevo formato incluía **ciertas mejoras sobre HLP**, por eso quedó incluido a partir de **Windows Vista**. Si bien es cierto que desde **Windows 7 solo** aparece como ayuda para **algunas** aplicaciones.

- 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**.

- **HPJ**

Los ficheros HPJ, al igual que los anteriores, utilizan:

- ficheros de tipo **.cnt** para la **representación de la tabla** de contenido,
- ficheros de tipo **.shg** para la representación de **gráficos**.

Este tipo de ficheros son creados utilizando herramientas del tipo **Help Workshop**.

- **IPF** (Information Presentation Facility)

Se trata de ficheros que **utilizan IPF**, un lenguaje muy similar a HTML. Son utilizados sobre todo para **ayuda en línea** e **hipertexto**.

- **JavaHelp**

Finalmente, encontramos este último tipo de formato de fichero de ayuda que será tratado en el desarrollo de este capítulo. Reciben este nombre tanto el **formato de ficheros de ayuda** en línea como el **sistema** que se encarga de su generación.

Este tipo de ficheros están implementados en **Java** y se utilizan para la generación de ayuda de aplicaciones desarrolladas en Java.

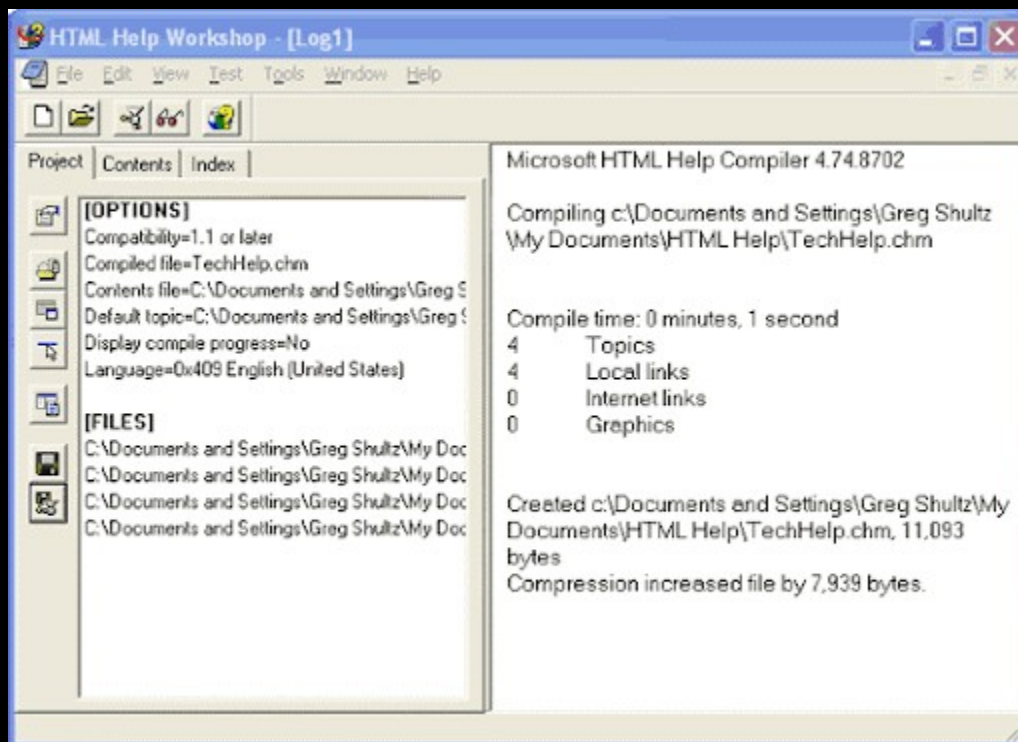
## Herramientas de generación de ayuda

### Help Workshop

Esta herramienta permite la creación de **ficheros de ayuda para Windows**. Está formada por:

- un editor de imágenes,
- el administrador de proyectos,
- el compilador, clave para el desarrollo de manuales y ayuda, puesto que permite reducir el tamaño final para su posterior distribución.

Para descargar, esta herramienta está disponible en este [enlace](#). Tras la descarga, se ejecuta el **instalador de software htmlhelp.exe**.



## Help Maker

Help Maker es una herramienta gratuita que permite la creación de ficheros de ayuda a través de un **editor de archivos**. Incluye diferentes opciones de personalización que permiten dar **formato al texto**, lo cual posibilita conseguir documentos más organizados y fáciles de leer.

Otra de las características de esta herramienta es que toda la información queda contenida en **un único fichero**, por lo que es posible **exportar** toda la ayuda en **un solo documento PDF**.

## Shalom Help Maker

Herramienta gratuita que permite la **creación de ficheros de ayuda para aplicaciones**. Al igual que en el caso anterior, este tipo de herramientas permite la creación de documentos de ayuda a través de **editores de texto** que resultan sencillos de utilizar.

Además de las características propias de un editor de texto, **permite** la creación de **índices**, contenidos **enlazados** a otras páginas, **links externos**, así como la **creación de imágenes** que enriquecen el contenido del documento final.

## Tablas de contenido

Una tabla de contenido permite reflejar la **estructura y contenido** de un documento. Se encuentra **esquemática** en varios niveles de elementos entre los que se distinguen **títulos y subtítulos**. Aunque el diseño dependerá del desarrollador y de otros muchos factores, algunas de las características típicas son:

- Pueden **mostrar el número de página o no**, en función del diseño.
- **Casi siempre** incluye un **enlace directo** en cada uno de los **títulos o subtítulos**, por lo que aparece compuesta, ya que apunta al contenido de cada uno de esos títulos.
- Se sitúan **al inicio** de cualquier documento, como si de un índice de libro se tratara.
- Su elaboración no es trivial, requiere de un **proceso de análisis completo** de la documentación que se va a exponer, así como de los títulos y subtítulos que a continuación recogen.
- **No debe duplicarse** la información.
- Los **títulos** deben ser lo **suficientemente claros** para que un usuario no necesite de otro documento para entender cómo funciona el manual de ayuda.

Las pautas de diseño de una buena tabla de contenidos son:

### Diagrama de pautas de diseño de una tabla de contenidos

1. Selección de los **temas y subtemas** que aparecerán en la tabla.
2. **Numeración** de los temas y subtemas en función del patrón de numeración escogido.
3. Creación de la tabla de contenido añadiendo los **enlaces** oportunos.
4. Constante revisión de la tabla para **actualizar y modificar** la información.

## Tipos de documentación

En función del tipo de aplicación, del tipo de usuario o de la fase concreta en la que se encuentra la aplicación, es posible realizar diferentes **clasificaciones** en cuanto los tipos de documentación se refiere.

A grandes rasgos, podemos dividir la documentación de aplicaciones software en **dos grandes grupos**:

- **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:

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

- **Documentación técnica:**

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

- **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, de por sí, debe ser claro. En este ámbito, debe **tenerse en cuenta** lo siguiente:

- **Comienzos de módulos.**

- **Variables, constantes, procedimientos y funciones.**

- **Comentarios introductorios** sobre **bloques** de código, funciones o módulos.

- **No comentar lo obvio**, ya que el exceso de comentarios puede ser contraproducente.

- **Documentación externa:** En este caso, suele tratarse de un **manual técnico** orientado a programadores para **facilitar el mantenimiento y desarrollo** de la aplicación en un futuro.



## Tipos de manuales

Existen tipos de manuales y guías diferentes entre sí con relación al contenido de cada uno y a su manera de utilización.

### Manual o guía de usuario

El manual de usuario contiene la información necesaria para **facilitar al usuario la comprensión y utilización** del programa desarrollado en **diferentes ámbitos**:

- **formación** del usuario,
- guía de consulta ante **dudas**,
- ayuda para detectar y corregir **errores**,
- etc.

Aunque no hay ninguna norma sobre su elaboración, debe tratarse, por tanto, de un documento que sea **amigable y claro** para el usuario, ya que, al fin y al cabo, es quien va a utilizarlo. Para ello, es aconsejable el uso de **imágenes y gráficos, textos claros y concisos, ejemplos** ilustrativos, etc.

### Tutoriales visuales

Debemos de tener en cuenta al realizarlos:

1. Analizar las necesidades del público al que va dirigido.
2. Hay que ser claros y concisos. Mejor cortos y avisar del contenido al comienzo.
3. Analizar otros contenidos similares
4. Guionizar el vídeo antes de ser grabado. No basta con dar al botón de grabar.

Herramientas para captar la pantalla:

Screecast, LightShot, Animaker (nos permite crear película con imágenes) ...

Nota: A veces puede grabarse el pase de diapositivas con la propia aplicación de diapositivas, por ejemplo en Microsoft Power Point se puede lograr esto.

## Guía rápida

Puede orientarse a usuarios y/o encargados de mantenimiento, es decir, para una misma aplicación, pueden desarrollarse **varias guías rápidas** en función de la **complejidad**.

Se caracteriza por proporcionar **información muy concreta** sobre diversos procedimientos que puede realizar una aplicación. En aplicaciones muy sencillas, puede constituir **por sí misma un manual**.

## Guía de referencia

Estas suelen desarrollarse **para usuarios** con cierto conocimiento y **experiencia** en el uso de la aplicación.

Por ello, suelen contener información asociada a **aspectos más técnicos**: relación de **mensajes de error** y su posible origen, tipos de **datos de entrada** permitidos en la aplicación, **comandos** aceptados, etc.

## Manual y guía de explotación

Contienen la información necesaria para **poner en uso** la aplicación. Es decir, están muy orientados 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, etc).

En función de la complejidad de la aplicación, puede:

- **dividirse** en Manual de **instalación** y Manual de **configuración**,
- o bien constituir un **único documento**.

## JavaHelp

### Generador de sistema de ayuda

JavaHelp es una aplicación que permite la creación de sistemas de ayuda para ser integradas posteriormente en una aplicación. Esta herramienta está orientada al uso de **aplicaciones desarrolladas con** lenguaje de programación **Java**.

Esta herramienta permite la creación de **documentos de ayuda** muy completos que incluye

- **tablas** de contenidos,
- motor de **búsqueda**,
- **glosario**,
- sección de **favoritos**...

#### Pasos secuenciados para crear un sistema de ayuda con JavaHelp.

1. En primer lugar, se ha de **dar forma** a la ayuda, lo que implica
  - **especificar la organización**,
  - **diseñar los temas** que van a tomar la ayuda.
2. Descarga e **instalación de JavaHelp**, o del sistema escogido, desde:
  - diversos repositorios,
  - la página de Sun.
3. Creación de los **ficheros de JavaHelp** necesarios en función de, o según, la organización de la ayuda determinada (según organización determinada).
4. Se construye un **fichero JAR** que incluya **todos los ficheros** y permita **mejorar su distribución**.
5. Se **añade** la ayuda a la aplicación.

## Ficheros de aplicación JavaHelp

La herramienta requiere del uso de **diferentes ficheros** de **varios tipos** para el diseño de cada una de las partes que forman la ayuda final.

- **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**.
- **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. Se muestra uno de ellos por pantalla. Deben tener organización **jerárquica**.
- **Base de datos de búsqueda:** Se debe utilizar la herramienta **jhindexer** para generarla.
- **Tabla de contenidos:** XML, Incluye el **contenido** de la ayuda y su **distribución**.

Tras la creación de todos los ficheros, el directorio contenedor queda de la siguiente forma:

### Estructura de directorios y ficheros en JavaHelp

```
./help
  indice.xml
  Map.jhm
  Helpset.hs
  TablaDeContenidos.xml
  help/html
    Tema1.html
    Tema2.html
./JavaHelpSearch
```

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

## Sistema de ficheros JavaHelp, vinculación a interfaz en Java

Una vez creada la estructura del sistema de ayuda, la invocaremos desde el fichero correspondiente de java. Hay que tener claros ciertos pasos:

La estructura de las carpetas y la forma de ser invocada la estructura desde java.

La estructura de carpetas es la siguiente:

- Carpeta **bin**:
  - librerías que necesitaremos en formato .jar para implementar javahelp.
- Carpeta **src**:
  - fichero .java: con este fichero desplegamos el sistema de ayuda. Se trata de un programa habitual donde modelamos una interfaz, y el método `ponLaAyuda()`, en primer lugar carga el fichero de ayuda para que cargue el fichero `help_set.hs.`, en el que vinculábamos todo. Además crea `HelpSet helpset` y `HelpBroker hb`, y a continuación `hb.enableHelpOnButton` y `hb.enableHelpKey` hacen que aparezca la ayuda dependiendo de dónde se pulse.
- Carpeta **help** (la más importante): Aquí desplegamos el sistema de ayuda que hemos programado. Dentro de help:
  - carpeta **html**: con ficheros que implementan la ayuda, en html (p.e. `main.html`, `principal.html`, `secundaria.html`). Cada fichero puede llamar a otro dentro de la misma carpeta.
  - Carpeta **JavaHelpSearch**: se crean automáticamente sus elementos al compilar: `DOCS`, `DOCS.TAB`, `OFFSETS`, `POSITIONS`, `SCHEMA` `TMAP`.

El **resto de ficheros** (de **configuración**) dentro de help son los que nosotros vamos a implementar:

- `map_file.jhm`: mapeo entre todas las páginas e identificador de cada una de ellas.
- `toc.xml`: mapeo para la tabla de contenidos.
- `indice.xml`: modelado del índice de nuestro sistema de ayuda.
- `help_set.hs`: se vinculan todos los ficheros anteriores.

De esta manera tendríamos creado el sistema de ayuda y vinculado a nuestra aplicación en java.

## Implementación de ficheros JavaHelp

Para desarrollar cada una de las páginas o ficheros de ayuda contenidos en esta aplicación, se deben escribir tantos ficheros en HTML como se necesiten para la documentación de la aplicación. Además, son **necesarios cuatro ficheros de configuración** que se detallan a continuación:

1. **map\_file.jhm**: Este archivo se encarga de **mapear** cada uno de los HTML de ayuda creados indicando su **ruta** y un **identificador** para cada uno de estos ficheros, a través del cual serán **referenciados desde el resto de la aplicación**.

```
<?xml version='1.0' encoding='ISO-8859-1' ?>

<map version="1.0">

    <mapID target="Principal" url="html/principal.html" />

    <mapID target="Tema 1" url="html/tema1.html" />

    <mapID target="Tema 2" url="html/tema2.html" />

</map>
```

2. **toc.xml**: Es un fichero que recoge la **tabla de contenidos y su índice**. La estructura se basa en 'tocitems' para indicar cada uno de los **elementos** en los que se organiza la tabla.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<toc version="1.0">

    <tocitem text="Principal" target="Principal"/>

    <tocitem text="Tema 1" target="Tema 1"/>

    <tocitem text="Tema 2" target="Tema 2"/>

</tocitem>

</toc>
```

3. **indice.xml**: es un fichero que recoge la **tabla del índice**. La estructura se basa en 'indexitems' para indicar cada uno de los elementos en los que se organiza el índice.

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<index version="1.0">
    <indexitem text="Principal" target="Principal"/>
    <indexitem text="Tema 1" target="Tema 1"/>
    <indexitem text="Tema 2" target="Tema 2"/>
-</index>
```

4. **help\_set.hs**. Este último fichero de configuración contiene a los anteriores. En él se **describen** las secciones relativas a las **vistas**.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<helpset version="1.0">
    <title>Ayuda JavaHelp</title>
    <maps>
        <homeID>Principal</homeID>
        <mapref location="map.jhm"/>
    </maps>
    <view>
        <name>Busqueda</name>
        <label>Buscar</label>
        <type>javax.help.SearchView</type>
        <data
            engine="com.sun.java.help.search.DefaultSearchEngine">
JavaHelpSearch
        </data>
    </view>
```

```
<view>
    <name>Indice</name>
    <label>indice</label>
    <type>javax.help. IndexView</type>
    <data>Indice.xml</data>
</view>
<view>
    <name>Tabla de Contenidos</name>
    <label>Tabla de Contenidos</label>
    <type>javax.help.TOCView</type>
    <data>tablaDeContenidos.xml</data>
</view>
</helpset>
```



## Incorporación de la ayuda en Eclipse

Finalmente, se incluye la ayuda desarrollada **en la aplicación de Java** en la que va a estar contenida. En primer lugar, se deben instalar los **paquetes de JavaHelp** en el entorno de desarrollo.

- `Import java.net.*`: Permite **representar URL**, es decir, una ruta al fichero.
- `Import javax.help.*`: Permite **utilizar los ficheros** que hemos creado **desde** una aplicación **Java**. Incluye la **clase HelpSet** y **HelpBroker**, que se utilizarán posteriormente.

La inclusión de estos paquetes permitirá trabajar con las siguientes **clases y métodos**:

- Clase `HelpSet`: Posibilitará **usar los ficheros** del sistema de ayuda. **Métodos**:
  - `findHelpSet`: Devuelve la **URL del fichero HelpSet**.
  - `createHelpBroker`: Crea un objeto **HelpBroker asociado al HelpSet**.
- Clase `HelpBroker`: Permite **visualizar el contenido** de la ayuda desde la aplicación.

Principales **métodos**:

- `enableHelp`: Hace **referencia al tema o topic** que se debe visualizar al pulsar una tecla de ayuda sobre un componente determinado.
- `enableHelpKey`: Permite habilitar la **tecla de ayuda**.
- `enableHelpOnButton`: Produce que se **despliegue la ayuda** al pulsar un elemento.

## Método para vincular a interfaz la ayuda en Eclipse

```
/**
 * Hace que el item del menu y la pulsacion de F1 visualicen la ayuda.
 */
private void ponLaAyuda() {
    try {
        // Carga el fichero de ayuda
        File fichero = new File("../help/help_set.hs");
        URL hsURL = fichero.toURI().toURL();
        // Crea el HelpSet y el HelpBroker
        HelpSet helpset = new HelpSet(getClass().getClassLoader(), hsURL);
        HelpBroker hb = helpset.createHelpBroker();
        // Pone ayuda a item de menu al pulsarlo y a F1 en ventana
        // principal y secundaria.
        hb.enableHelpOnButton(itemAyuda, "aplicacion", helpset);
        hb.enableHelpKey(principal.getContentPane(), "ventana_principal", helpset);
        hb.enableHelpKey(secundaria.getContentPane(), "ventana_secundaria", helpset);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

## Ejemplo sistema de ayuda con JavaHelp

### Página principal y Capítulos

#### Ficheros map\_file.jhm, toc.xml e index.xml

Los elementos contenidos en la ayuda se son:

- Página principal
- Capítulo 1, 2, 3 y 4. (Todos los ficheros de ayuda en formato .html se encuentran dentro de la ruta html/capitulos).

Se diseñan los ficheros indicando la ruta en la que se encuentra los ficheros en los que aparece la ayuda.

Fichero map\_file.jhm

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<map version='1.0'>
    <mapID target="Página Principal" url="html/principal.html" />
    <mapID target="Capítulo 1" url="html/capitulos/cap1.html" />
    <mapID target="Capítulo 2" url="html/capitulos/cap2.html" />
    <mapID target="Capítulo 3" url="html/capitulos/cap3.html" />
    <mapID target="Capítulo 4" url="html/capitulos/cap4.html" />
</map>
```

Fichero toc.xml

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<toc version="1.0">
    <tocitem text=""Página" Principal="" target=""Página">
        <tocitem text=""Capítulo" target=""Capítulo"/>
        <tocitem text=""Capítulo" target=""Capítulo"/>
        <tocitem text=""Capítulo" target=""Capítulo"/>
        <tocitem text=""Capítulo" target=""Capítulo"/>
    </tocitem>
</toc>
```

```
</tocitem>
```

```
</toc>
```

Fichero index.xml

```
<?xml version='1.0' encoding='ISO-8859-1'?>
```

```
<index version="1.0">
```

```
<indexitem text=""Página" Principal="" target=""Página">
```

```
<indexitem text=""Capítulo" target=""Capítulo"/>
```

```
<indexitem text=""Capítulo" target=""Capítulo"/>
```

```
<indexitem text=""Capítulo" target=""Capítulo"/>
```

```
<indexitem text=""Capítulo" target=""Capítulo"/>
```

```
</indexitem>
```

```
</index>
```

## Fichero help\_set.hs

Para la inclusión de un sistema de ayuda generado con JavaHelp, se elabora un último fichero, **help\_set.hs**, en el que se **incluyen los archivos creados** anteriormente. De esta forma, quedan **vinculados tanto el índice como la tabla de contenidos**.

Los ficheros implementados: map\_file.jhm, toc.xml y index.xml, vistos anteriormente, son los referenciados desde el archivo help\_set.hs.

Este documento relaciona todo el sistema de ayuda, incluye el índice y la tabla de contenidos.

En este documento, se diferencia claramente **dos partes**:

- **<maps>**. Esta etiqueta referencia el sistema de mapeo donde se recogen cada uno de los **HTML de ayuda creados** indicando su **ruta (location)** y un **identificador** para cada uno de estos ficheros.
- **<view>**. En este caso, se desarrollan dos vistas: una de ellas, la correspondiente al **fichero toc.xml**. Este es el fichero que recoge la **tabla de contenidos y su índice**. También se utiliza para incluir el fichero **Índice.xml**, bajo el cual se recoge la **estructura del índice**.

### Fichero help\_set.hs

```
<?xml version='1.0' encoding='ISO-8859-1'?>

<helpset version=""1.0"">

    <title> Sistema de ayuda </title>

    <maps>

        <homeID> Página principal </homeID>

        <mapref location=""map.jhm""/>

    </maps>

    <view>

        <name> Índice </name>

        <label> Índice </label>

        <type>javax.help.IndexView </type>

        <data> Indice.xml </data>
```

</view>

<view>

<name> Tabla de contenidos </name>

<label> Tabla de contenidos </label>

<type>javax.help.TOCView </type>

<data> toc.xml </data>

</view>

</helpset>

## Ayuda genérica y ayuda sensible al contexto

Al instalar una aplicación solemos encontrarnos con un conjunto de manuales y guías de instalación incluidas en los ficheros de ayuda. Aunque la lectura de los mismos es más que recomendable, en la mayoría de las ocasiones sólo se recurre a éstos ante un problema en la instalación o la necesidad de realizar una configuración muy específica de la aplicación.

En cuanto al tipo de información contenida en un fichero de ayuda, cabe diferenciar entre dos bloques de información claramente distintos, hablamos de la **ayuda genérica** y la **ayuda sensible** al contexto.

- La **ayuda genérica** es el tipo de ayuda que recoge **toda la información** de la aplicación. Podríamos decir que se trata del manual completo de uso del sitio. Para localizar la información que sea de utilidad en cada momento, habrá que **navegar por el mapa del sitio**.
- La **ayuda sensible al contexto** es una ayuda **concisa sobre la funcionalidad u operación** que se está realizando en cada momento. Habitualmente, aparece **accesible** o cada vez es más habitual y aconsejable que así se haga, desde cada ventana, cuadro de texto o acción en la que se incluya. Suele aparecer enlazada en botones con el **símbolo de interrogación ?** y, en ocasiones, con la **letra i (información)**.