

DESARROLLO DE INTERFACES
TÉCNICO EN DESARROLLO DE APLICACIONES
MULTIPLATAFORMA

Documentación de aplicaciones: Ayudas

ÍNDICE

/ 1. Introducción y contextualización práctica	3
/ 2. Ficheros de ayuda y formatos	4
/ 3. Herramientas de generación de ayuda	5
/ 4. Tablas de contenido	6
/ 5. Tipos de documentación	6
/ 6. Tipos de manuales	7
/ 7. Generación de un sistema de ayuda con JavaHelp	8
7.1. Ficheros de aplicación JavaHelp	9
7.2. Implementación de ficheros JavaHelp	10
7.3. Incorporación de la ayuda en Eclipse	11
/ 8. Caso práctico 1: “Diseño de un sistema de ayuda con JavaHelp (I)”	13
/ 9. Caso práctico 2: “Diseño de una sistema de ayuda con JavaHelp (II)”	14
/ 10. Resumen y resolución del caso práctico de la unidad	15
/ 11. Bibliografía	15

OBJETIVOS

Identificar sistemas de generación de ayudas.

Generar ayudas en los formatos habituales.

Generar ayudas sensibles al contexto.

Documentar la estructura de la información persistente.

Confeccionar manuales de usuario y guía de referencia.

Confeccionar manuales de instalación, configuración y administración.

Confeccionar tutoriales.

/ 1. Introducción y contextualización práctica

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 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 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.

Escucha el siguiente audio donde planteamos la contextualización práctica de este tema, encontrarás su resolución en el apartado Resumen y resolución del caso práctico.



Fig. 1. Tipos de manuales y guías.



Audio intro. "Ayuda genérica y ayuda sensible al contexto"

<https://bit.ly/2F2c7uN>



/ 2. Ficheros de ayuda y formatos

Un **fichero** es un elemento que contiene cualquier tipo de información y que puede estar recogida en múltiples tipos de formatos, tanto en soporte físico como en digital.

Un **fichero de ayuda** es, por tanto, 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. En este tema, nos vamos a centrar en los **ficheros de ayuda en formato digital**.

Es importante conocer la estructura general de cualquier fichero de ayuda. Estos suelen estar compuestos por dos partes claramente diferenciadas: **mapa de fichero y la vista de información**.



Fig. 2. Diagrama de sistemas de ayuda.

El **mapa de fichero**, como su propio nombre indica, 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.

Por otro lado, la **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.

Es posible encontrar múltiples tipos de 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



Fig. 3. Icono HLP.



- 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, mientras que utilizan ficheros de tipo .shg para la representación de gráficos.

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

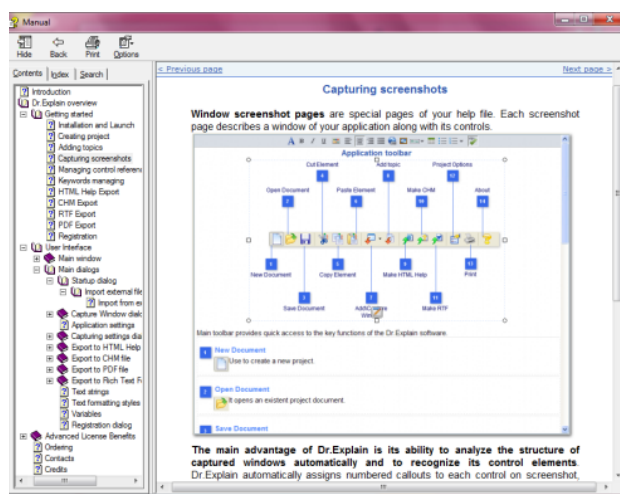


Fig. 4. Interfaz CHM.

• 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.

/ 3. Herramientas de generación de ayuda

a. *Help Workshop*

Esta herramienta permite la creación de ficheros de ayuda para Windows. Está formada por un editor de imágenes, el administrador de proyectos y el compilador, este último es 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*.

b. *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.

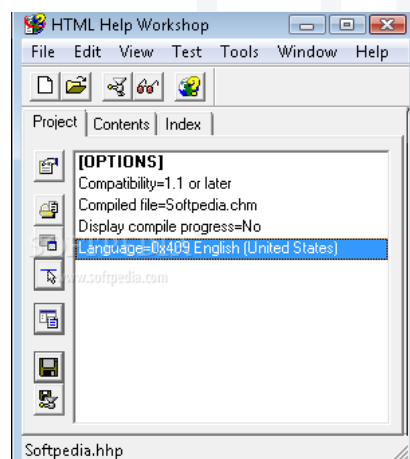


Fig. 5. Interfaz *Help Workshop*.

c. 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.

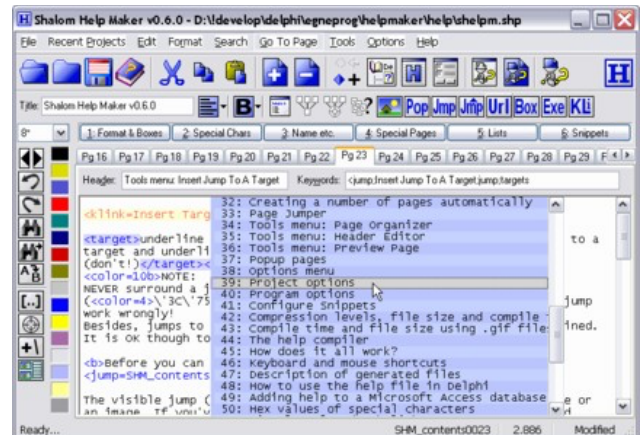


Fig. 6. Interfaz Shalom Help Maker.

/ 4. Tablas de contenido

Una tabla de contenido permite reflejar la estructura y contenido de un documento. Se encuentra esquematizada 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 se exponen en el siguiente diagrama:

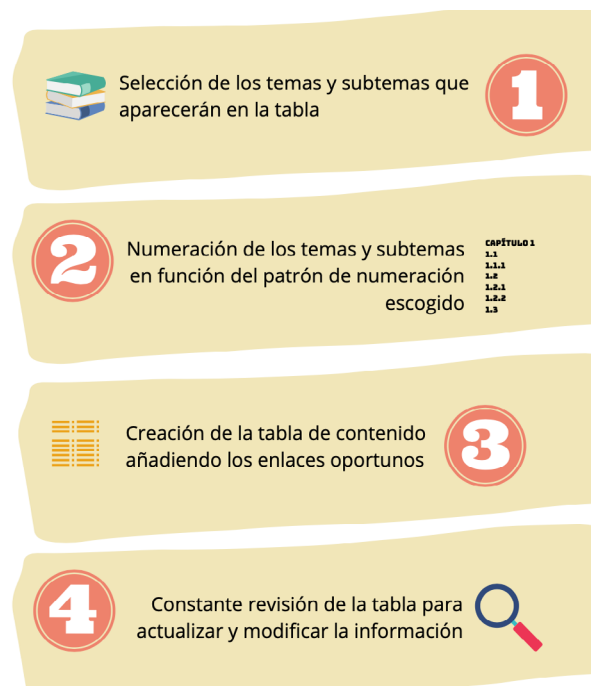


Fig. 7. Diagrama de pautas de diseño de una tabla de contenidos.

/ 5. 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.

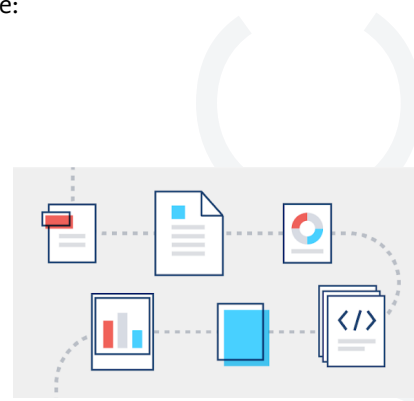


Fig. 9. Diagrama tipos de documentación y formatos.

/ 6. 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.

a. 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.



Vídeo1. "Tutoriales visuales"

<https://bit.ly/3hZARIM>



b. Guía rápida y guía de referencia

Además de los manuales indicados anteriormente, la documentación de una aplicación puede complementarse con estas guías, según corresponda en cada caso.

- **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.

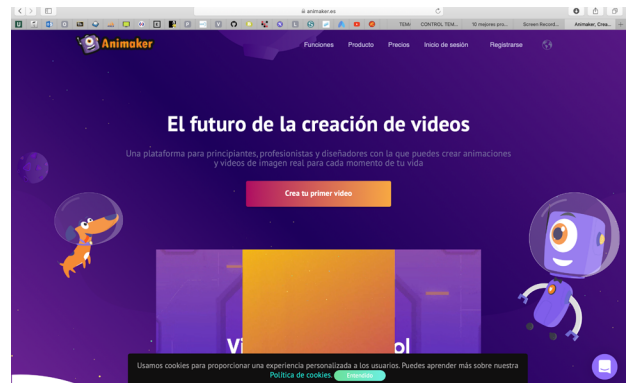


Fig. 10. Sitio web Animaker.



Audio 1. "Manual y guía de explotación"

<https://bit.ly/3jMSYvX>



/ 7. Generación de un sistema de ayuda con JavaHelp

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...

Para crear un sistema de ayuda con *JavaHelp*, se deben seguir un conjunto de pasos secuenciados.

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

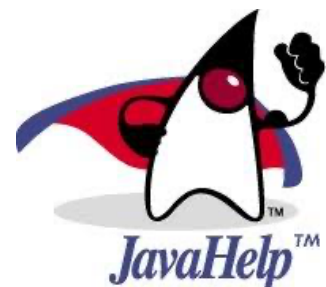


Fig. 11. Icono JavaHelp.



5. Se añade la ayuda a la aplicación.

1. Dar forma a la ayuda
<ul style="list-style-type: none"> • Especificar organización • Diseñar temas
2. Descargar e instalar JavaHelp
<ul style="list-style-type: none"> • Diversos repositorios • Desde la página de Sun
3. Crear ficheros JavaHelp
<ul style="list-style-type: none"> • Según organización determinada
4. Construir fichero JAR
<ul style="list-style-type: none"> • Único fichero con todos los contruidos anteriormente
5. Añadir ayuda a la aplicación
<ul style="list-style-type: none"> • Según organización determinada

Tabla 1. Diagrama de diseño de un sistema de ayuda.

7.1. 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.

Fichero	Tipo	Descripción
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.
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.

Tabla 2. Ficheros de aplicación JavaHelp.

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



Fig. 12. Estructura de directorios y ficheros en JavaHelp.

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



Vídeo 2. "Sistema de ficheros JavaHelp, vinculación a interfaz en Java"
<https://bit.ly/3lPSoiM>



7.2. 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>
```

Código 1. map_file.jhm.

2. Fichero **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>
```

Código 2. toc.xml.



3. Fichero **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>
```

Código 3. *indice.xml*.

4. Fichero **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>Búsqueda</name>
    <label>Buscar</label>
    <type>javax.help.SearchView</type>
    <data engine="com.sun.java.help.search.DefaultSearchEngine">
      JavaHelpSearch
    </data>
  </view>

  <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>tablaDeContenidos.xml</data>
  </view>
</helpset>
```

Código 4. *help_set.hs*.

7.3. 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:
 - ***enableHelpKey*:** Permite habilitar la tecla de ayuda.
 - ***enableHelpOnButton*:** Produce que se despliegue la ayuda al pulsar un elemento.
 - ***enableHelp*:** Hace referencia al tema o topic que se debe visualizar al pulsar una tecla de ayuda sobre un componente determinado.

```
/**
 * 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();
    }
}
```

Código 5. Método para vincular a interfaz la ayuda en Eclipse.



/ 8. Caso práctico 1: “Diseño de un sistema de ayuda con JavaHelp (I)”

Planteamiento: Se va a implementar un sistema de ayuda utilizando JavaHelp. Los elementos contenidos en esta ayuda se muestran a continuación. Diseña todos los ficheros `map_file.jhm`, `index.xml` y `toc.xml` para desplegar este sistema de ayuda.

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

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

Desenlace:

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>
```

Código 6. Código `map_file.jhm` caso práctico 1.

Fichero `toc.xml`:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<toc version='1.0'>
  <tocitem text="Página Principal" target="Página Principal"/>
  <tocitem text="Capítulo 1" target="Capítulo 1"/>
  <tocitem text="Capítulo 2" target="Capítulo 2"/>
  <tocitem text="Capítulo 3" target="Capítulo 3"/>
  <tocitem text="Capítulo 4" target="Capítulo 4"/>
</tocitem>
</toc>
```

Código 7. Código `toc.xml` caso práctico 1.

Fichero index.xml:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<index version='1.0'>
  <indexitem text="Página Principal" target="Página Principal"/>
  <indexitem text="Capítulo 1" target="Capítulo 1"/>
  <indexitem text="Capítulo 2" target="Capítulo 2"/>
  <indexitem text="Capítulo 3" target="Capítulo 3"/>
  <indexitem text="Capítulo 4" target="Capítulo 4"/>
</index>
```

Código 8. Código index.xml caso práctico 1.

/ 9. Caso práctico 2: “Diseño de una sistema de ayuda con JavaHelp (II)”

Planteamiento: 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 en el caso práctico 1. De esta forma, quedan vinculados tanto el índice como la tabla de contenidos.

Nudo: Los ficheros implementados en el caso práctico 1 son los referenciados desde el archivo **hep_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 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.

También se utiliza para incluir el fichero *Índice.xml*, bajo el cual se recoge la estructura del índice.

Desenlace: 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>
```

Código 9. Código help_set.hs caso práctico 2.
JavaHelp.



/ 10. Resumen y resolución del caso práctico de la unidad

Como se ha visto en este tema, el desarrollo de interfaces gráficas requiere de la inclusión de **sistemas de ayuda y manuales** que permiten documentar todo el funcionamiento de una aplicación. Estos sistemas de ayuda resultan clave tanto para usuarios como para expertos.

Existen **diferentes tipos de documentación** que se ajustan a la aplicación en cada caso. Podemos hablar, a grandes rasgos, de la **documentación de pruebas y la documentación técnica**. La primera queda dividida en documentación de entrada y de salida, mientras que la de tipo técnico se subdivide en documentación interna y externa.

Una de las herramientas utilizadas para la generación de sistemas de ayuda en aplicaciones con Java es **JavaHelp**, la cual permite la creación de documentos de ayuda muy completos.

Para desarrollar este tipo de sistema, se han de crear un conjunto de ficheros (map_file.jhm, index.xml, toc.xml, ...) que permiten el **mapeo de fichero y la inclusión de sistemas de organización** como el índice o la tabla de contenido.

Las tablas de contenido permiten reflejar la estructura y contenido de un documento, estas se encuentran estructuradas en varios niveles de elementos entre los que se distinguen títulos y subtítulos.

Resolución del caso práctico inicial

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.

Ahora bien, 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).

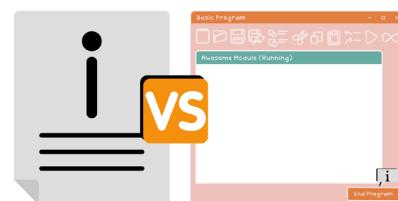


Fig. 13. Diferencia entre ayuda genérica y ayuda sensible al contexto.

/ 11. Bibliografía

Fernández, A., García-Miguel, B. y García-Miguel, D. (2020). *Desarrollo de Interfaces* (1.a ed.). Madrid, España: Síntesis.