



## Ejemplo:

---

# Construcción de ficheros de ayuda para Java con JavaHelp



## JavaHelp

---

- Qué es JavaHelp
- Características Básicas
  - Herramientas
  - Librerías
- Pasos en la construcción de la ayuda
- Incorporación de la ayuda en la aplicación



# Qué es JavaHelp

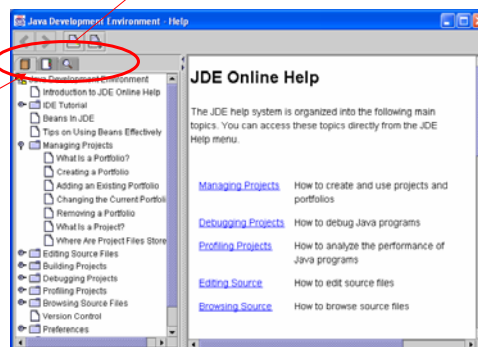
- Sistema que **facilita el desarrollo de ayuda *on-line***
- Es **independiente de la plataforma** (implementado en Java )
- Evita que cada usuario tenga que crear su propio sistema de ayuda propietario



# Características Básicas

Distintas vistas de la información

- **Visor de Ayuda**
  - Panel de contenidos
  - Panel de navegación
- **Tabla de contenidos**
- **Índice**
- **Búsqueda**





# Herramientas

---

- **hsvviewer**
  - Permite visualizar ficheros HelpSet. Puede invocarse desde línea de comandos o desde Windows
- **jhindexer**
  - Programa de línea de comandos que crea la base de datos de búsqueda
- **jhsearch**
  - Programa de línea de comandos que busca en la base de datos creada con jhindexer.



# Librerías

---

- **jh.jar**
  - Librería estándar que incluye todo lo necesario para el visor de ayuda y los navegadores
- **jhbasic.jar**
  - Subconjunto de la anterior que no soporta la búsqueda
- **jhtools.jar**
  - Incluye las herramientas para construir y consultar la base de datos
- **jhall.jar**
  - Contiene todas las clases de JavaHelp.





## Pasos en la Construcción de la Ayuda



## Pasos - Resumen

- 1. Creación de los temas de ayuda para el usuario (topics HTML)
- 2. Creación de los ficheros de metadatos que JavaHelp emplea para presentar la información
  - HelpSet
  - Map
  - Tabla de contenidos
  - Índice
  - ...



## Pasos Detallados

- 1. Crear topics HTML
- 2. Crear fichero map
- 3. Crear fichero HelpSet
- 4. Crear fichero tabla de contenidos
- 5. Crear fichero índice
- 6. Crear base de datos de búsqueda
- 7. Comprimir y encapsular ficheros de ayuda en ficheros JAR (Este paso no será necesario).

Imprescindibles



## Organización

```

..\help
  mi_helpset.hs
  mi_map.jhm
  mi_toc.xml
  mi_index.xml
  \ fuentes\
  \JavaHelpSearch

```

Topic1.html  
Topic2.html  
Topic3.html  
.....





# 1. Topics HTML

- Para crear *topics* se puede emplear cualquier herramienta (o bien directamente) que genere HTML
- Conviene agrupar *topics* relacionados y disponerlos jerárquicamente.
  - Si no, al menos, es altamente recomendable crear un subdirectorio donde se incluyan todos los topics html.



# 2. Fichero Map

- Asocia **identificadores** a los ficheros HTML que contienen los *topics*.
- Tiene extensión .jhm
- Sintaxis basada en XML



## Fichero Map (II)

```
<...
<map version="1.0">

  <mapID target="introduccion" url="fuentes/introduccion.html" />
  <mapID target="nuevo" url="fuentes/nuevo.html" />
  <mapID target="abrir" url="fuentes/abrir.html" />
  <mapID target="guardar" url="fuentes/guardar.html" />
  <mapID target="guardarcomo" url="fuentes/guardarcomo.html" />
  <mapID target="colorfondo" url="fuentes/colorfondo.html" />
  <mapID target="colortexto" url="fuentes/colortexto.html" />

</map>
```

Identificador

Fichero html asociado



## 3. Fichero HelpSet

- Es el **único fichero** del sistema de ayuda que es **referenciado explícitamente por la aplicación**.
- Es el primero que se carga cuando se activa el sistema de ayuda
- Contiene **toda la información necesaria para ejecutar el sistema de ayuda**
- Tiene la extensión **.hs**
- Sintaxis basada en XML



## Fichero HelpSet (II)

- Contiene, al menos, la siguiente información:
  - **Fichero map**, empleado para asociar identificadores a los nombre de las páginas html que constituyen la ayuda.
  - **Vistas**. Información que describe los **navegadores** que van a emplearse en el HelpSet (tabla de contenidos, índice, búsqueda)
  - **Título** del HelpSet
  - **<Home ID>** Identificador de la página a ser mostrada cuando se invoca el visor de ayuda sin especificar ningún identificador de página concreto



## Fichero HelpSet (III)

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<helpset>
  <title> Ejemplo de Ayuda </title>
  <maps>
    <homeID>introduccion</homeID>
    <mapref location="Map.jhm"/>
  </maps>
  <!-- views -->
  <view>
    <name>Tabla de contenidos</name>
    <label>Tabla de Contenidos</label>
    <type>javax.help.TOCView</type>
    <data>TOC.xml</data>
  </view>
</helpset>

```

Diagram illustrating the mapping of XML elements to their corresponding HelpSet components:

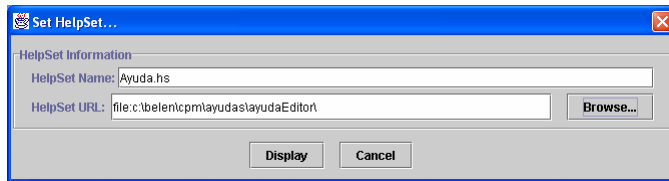
- `<title> Ejemplo de Ayuda </title>` → Título
- `<homeID>introduccion</homeID>` → Página a visualizar por defecto
- `<mapref location="Map.jhm"/>` → Fichero map
- `<data>TOC.xml</data>` → Fichero que contiene la tabla de contenidos





## Visualización del HelpSet

- En línea de comandos
  - `hsvviewer -helpset c:\ejercicios\ayudas\ayuda.hs`
  - `hsvviewer -helpset c:\ejercicios\ayudas\ejemplo.jar`
- En Windows



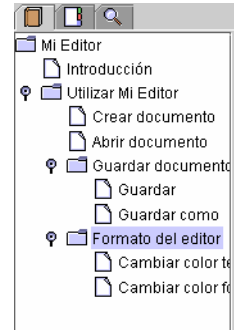
## 4. Fichero TOC

- Este fichero describe el **contenido y la distribución de la tabla de contenidos**
- Sintaxis basada en XML



## Fichero TOC (II)

```
<toc version="1.0">
  <tocitem text="Mi Editor ">
    <tocitem text="Introducción" target="introduccion"/>
    <tocitem text="Utilizar Mi Editor">
      <tocitem text="Crear documento" target="nuevo"/>
      <tocitem text="Abrir documento" target="abrir"/>
      <tocitem text="Guardar documento">
        <tocitem text="Guardar" target="guardar"/>
        <tocitem text="Guardar como" target="guardarcomo"/>
      </tocitem>
      <tocitem text="Formato del editor">
        <tocitem text="Cambiar color texto" target="colortexto"/>
        <tocitem text="Cambiar color fondo" target="colorfondo"/>
      </tocitem>
    </tocitem>
  </toc>
```



Especifica el texto para la entrada en la tabla de contenidos

Especifica el identificador del topic a mostrar cuando la entrada es seleccionada por el usuario. El identificador ha de estar definido en el fichero map



## 5. Fichero Índice

- Este fichero describe la **distribución y contenido del índice**
- Sintaxis basada en XML



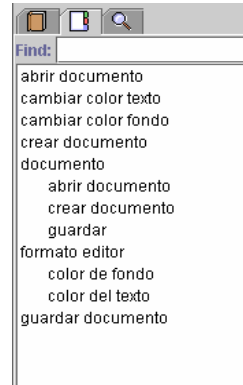
## Fichero Índice (II)

```
<index version="1.0">
  <indexitem text="abrir documento" target="abrir"/>
  <indexitem text="cambiar color texto" target="colortexto" />
  <indexitem text="cambiar color fondo" target="colorfondo" />
  <indexitem text="crear documento" target="nuevo" />

  <indexitem text="documento">
    <indexitem text="abrir documento" target="abrir" />
    <indexitem text="crear documento" target="nuevo" />
    <indexitem text="guardar" target="guardarcomo" />
  </indexitem>

  <indexitem text="formato editor">
    <indexitem text="color de fondo" target="colorfondo"/>
    <indexitem text="color del texto" target="colortexto"/>
  </indexitem>

  <indexitem text="guardar documento" target="guardar" />
</index>
```



Especifica el texto para la entrada en la tabla de contenidos

Especifica el identificador del topic a mostrar cuando la entrada es seleccionada por el usuario. El identificador ha de estar definido en el fichero map



## 6. Búsqueda

- Es necesario emplear el *jhindexer* para generar la base de datos de búsqueda
- La base de datos está formada por seis ficheros localizados en la carpeta *JavaHelpSearch*



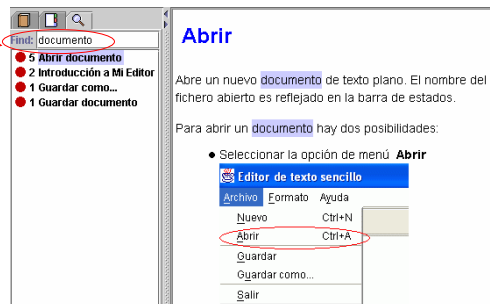
## Búsqueda (II)

- *Para crear la base de datos de búsqueda*
  - *jhindexer fuentes*
    - Se supone que fuentes es el directorio donde están almacenados todos los topics html
  - *jhindexer fuentes1 fuentes2*
    - Si tenemos más de un directorio donde estén almacenados los topics html. En este caso serían dos: fuentes1 y fuentes2
- **Funcionamiento**
  - jhindexer desciende recursivamente por la jerarquía de directorios indexando todos los ficheros que encuentra
  - Crea la carpeta JavaHelpSearch dónde almacena la base de datos de búsqueda



## Búsqueda (III)

- *Para verificar la validez de la búsqueda*
  - En línea de comandos
    - *jhsearch JavaHelpSearch*
  - Desde el visor de ayuda



## 7. Construcción ficheros JAR

- Una vez creada la ayuda se pueden encapsular los ficheros en un único fichero para entregar a los usuarios
- El formato empleado por JavaHelp para encapsular y comprimir es **JAR (Java ARchive)**
- Para ello se emplea el **comando jar**.



## Construcción ficheros JAR

- Creación de ficheros jar
  - **C:\ejercicios\ayudas> jar -cvf miayuda.jar \***
- Para ver los ficheros incluidos en un jar
  - **jar -tvf miayuda.jar**
- Para extraer los ficheros del jar
  - **jar -xvf miayuda.jar**





## Incorporación de la Ayuda en la Aplicación



## Clases Básicas Implicadas

- Clase URL
  - Paquete a importar: **import java.net.\*;**
  - Esta clase se utiliza para representar una URL (*Uniform Resource Locator*), es decir, para identificar un recurso.
    - Un recurso puede ser algo tan simple como un fichero o un directorio, o bien puede ser un objeto más complejo.





## Clases Básicas Implicadas

### ■ Clase HelpSet

- Paquete a importar: **import javax.help.\*;**
- Clase que permite tratar desde una aplicación Java **el conjunto de datos que constituyen el sistema de ayuda** (fichero helpset, fichero map, tabla de contenidos, índice y topics)
- Principales métodos
  - **findHelpSet** // método estático que localiza el fichero helpset y devuelve su URL
  - **createHelpBroker** // crea un objeto HelpBroker para este HelpSet



## Clases Básicas Implicadas

### ■ Clase HelpBroker

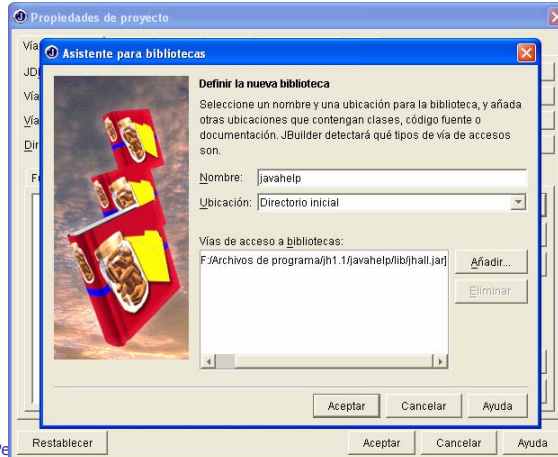
- Paquete a importar: **import javax.help.\*;**
- Esta clase es la encargada de gestionar **la visualización del contenido de la ayuda en la aplicación**
  - Proporciona los métodos apropiados para incorporar ayuda a la aplicación de una forma sencilla
- Principales métodos
  - **enableHelpKey** // Habilita la tecla de ayuda sobre un componente
  - **enableHelpOnButton** // Hace que se despliegue la ayuda cuando se presiona un determinado botón o elemento de menú.
  - **enableHelp** // Indica el topic a mostrar cuando se pulse la tecla de ayuda sobre un determinado componente



# Pasos

## para incorporar la ayuda en la aplicación

- 1. Incorporar la librería **jhall.jar** a la aplicación
  - Proyecto->Propiedades->Vías de acceso->Bibliotecas necesarias



# Pasos

## para incorporar la ayuda en la aplicación

- 2. Importar los paquetes que son necesarios en la aplicación
 

```
import javax.help.*;
import java.net.*; // para la clase URL
```





# Pasos

## para incorporar la ayuda en la aplicación

- 3. Encontrar el **fichero HelpSet** y crear un objeto HelpSet

```
Try{
    hsURL = HelpSet.findHelpSet(null,"ayuda/ayuda.hs"); // localiza el fichero helpset
    hs= new HelpSet(null,hsURL); // crea un objeto HelpSet

} catch (Exception ee){

    JOptionPane.showMessageDialog(this,"Fichero HelpSet no encontrado");
    return;
}
```



# Pasos

## para incorporar la ayuda en la aplicación

- 4. Crear un **objeto HelpBroker** que facilita la visualización de la ayuda en la aplicación

```
hb = hs.createHelpBroker ();
```



# Pasos

## para incorporar la ayuda en la aplicación

- 5. Asociar la ayuda a un botón o elemento de menú

```
hb.enableHelpOnButton(jButton1,"introduccion",hs)
```



jButton1



# Pasos

## para incorporar la ayuda en la aplicación

- 6. Asociar ayuda sensible al contexto a un componente
  - Se basa en asociar *topics* de ayuda con objetos en la interfaz gráfica de usuario: menús, botones, textfield y ventanas.
  - Se asignan identificadores (ID) a cada uno de estos elementos
  - Estos identificadores (ID) deben estar especificados en el fichero map.



# Pasos

## para incorporar la ayuda en la aplicación

- 6. Asociar ayuda sensible al contexto a un componente.  
Para ello:
  - Hacer que un componente responda a la pulsación de la tecla de ayuda (F1)  
`hb.enableHelpkey(getRootPane(),"introduccion",hs);`  
// responden a la tecla F1 todos los componentes situados en la ventana
  - Asociar a cada componente una página que será la que se muestre cuando se pulse la tecla de ayuda (F1) sobre él  
`hb.enableHelp(jButton1,"guardar",hs);`  
`hb.enableHelp(jButton2,"abrir",hs);`

