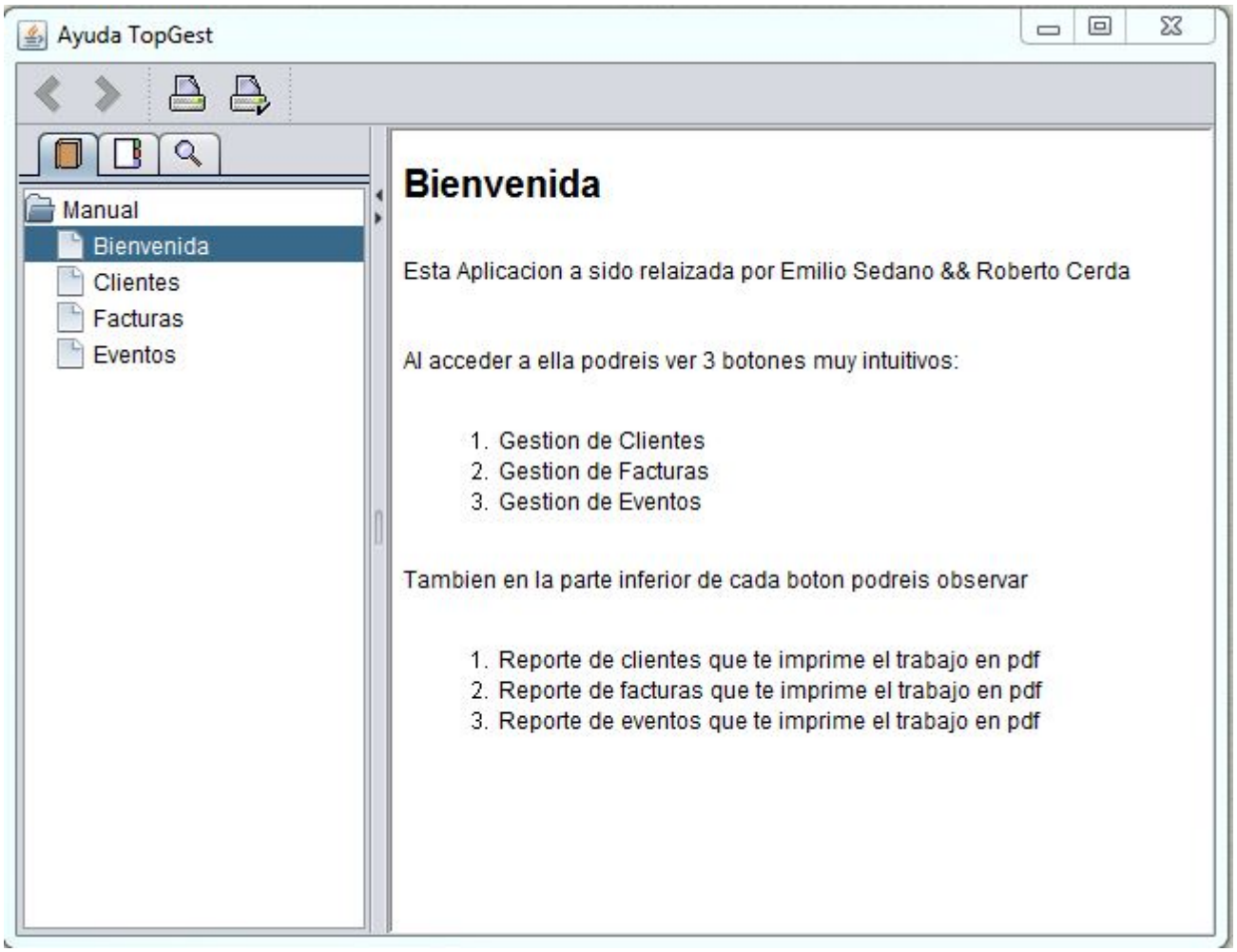
CREACIÓN DE AYUDA CON JAVAHELP



La generación de la ayuda con JavaHelp la podemos dividir en dos fases:

Fase 1. Creación de los archivos de la ayuda

En esta fase se procede a crear los archivos de la ayuda. Por un lado estarán los archivos que contienen la ayuda propiamente dicha y por otro los archivos de configuración.

Fase 2. Integración de la ayuda con la aplicación

En esta fase se procede a integrar la ayuda con creada en la fase 1 con nuestra aplicación.

1. CREACIÓN DE OS ARCHIVOS DE AYUDA

Vamos a distinguir entre dos tipos de archivos, los que contienen la ayuda propiamente dicha y los archivos de configuración.

1.1. ARCHIVOS DE AYUDA

Lo más significativo de la ayuda con JavaHelp es que esta información se almacena en documentos HTML que, por comodidad llamaremos páginas. Para ello podemos utilizar cualquier herramienta gráfica o escribir las etiquetas directamente.

Es conveniente que haya una página de inicio que contenga un mensaje de bienvenida, una introducción sobre la aplicación y sobre "el fabricante".

1.2. ARCHIVOS DE CONFIGURACIÓN

Una vez que tenemos los documentos de la ayuda debemos generar una serie de archivos para crear la ayuda. Esto archivos tienen una estructura XML. Los archivos son los siguientes:

- Mapa de archivos HTML (map.jhm)
- Tabla de contenidos (toc.xml)
- Índice (index.xml)
- HelpSet (helpset.hs)

MAPA DE ARCHIVOS HTML

Este archivo indica todos los archivos .html que componen nuestra ayuda. En este archivo-mapa damos a cada archivo .html una clave o nombre (target), que nos servirá a partir de ahora para identificar dicho archivo de ayuda.

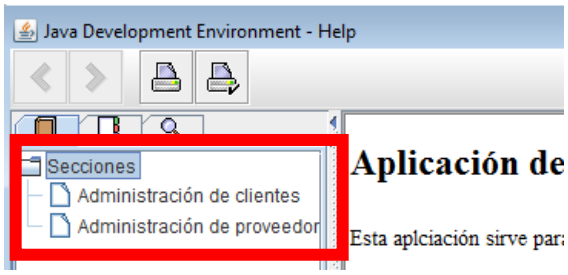
Este archivo mapa tiene formato xml, pero se le suele poner extensión .jhm (java help map). Lo normal es colocarlo en el directorio "ayuda" que creamos antes, junto o por encima de los archivos .html de nuestra ayuda.

El código de este archivo es como el que se muestra a continuación. Dentro de la etiqueta map se declaran las páginas de nuestra ayuda. La información de página está contenida dentro de una etiqueta mapID que contiene los siguientes atributos:

- **target**: el valor de este atributo será el utilizado en el resto de archivos para hacer referencia a las páginas de ayuda.
- **url**: contiene la ruta y nombre de la página de ayuda.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE map
PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp Map Version 2.0//EN"
"http://java.sun.com/products/javahelp/map_2_0.dtd">

<map version="1.0">
   <mapID target="inicio" url="principal.html" />
   <mapID target="clientes" url="html/clientes.html" />
   <mapID target="proveedores" url="html/proveedores.html" />
</map>
```

TABLA DE CONTENIDOS



Este archivo es una especie de tabla de contenidos. En ese archivo se van indicando los capítulos y subcapítulos de la ayuda. Lo que pongamos en este archivo saldrá en la parte izquierda de la ventana de ayuda con forma de árbol.

El nombre de este archivo suele ser toc.xml y no creo necesario cambiarlo.

El contenido de este archivo es el siguiente:

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

<toc version="1.0">
    <tocitem text="Secciones" target="inicio">
       <tocitem text="Administración de clientes" target="clientes" />
       <tocitem text="Administración de proveedores" target="proveedores" />
    </tocitem>
</toc>
```

La etiqueta tocitem contiene dos atributos:

- **text**: contendrá el texto que se va a mostrar.
- **target**: contiene la página que se abrirá cuando se pulse sobre el texto. La referencia a la página se hace a través del target definido en el Mapa de archivos HTML.

Hay que tener en cuenta que podemos crear niveles o estructura de árbol incluyendo etiquetas tocitem unas dentro de otras.

ÍNDICE

Contine el índice de la ayuda (index.xml).

```
<?xml version='1.0' encoding='UTF-8' ?>
<index version="1.0">
    <indexitem text="Administración de clientes" target="clientes" />
    <indexitem text="Administración de proveedores" target="proveedores" />
</index>
```

HELPSET

Es el archivo principal de configuración de JavaHelp. En este archivo indicamos cuales son los otros archivos que queremos usar entre los cuales están el mapa mapa.jhm y el de contenido toc.xml. (helpset.hs)

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE helpset
  PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp HelpSet Version 2.0//EN"
         "http://java.sun.com/products/javahelp/helpset_2_0.dtd">
<helpset version="2.0">
   <!-- title -->
   <title>Java Development Environment - Help</title>

   <!-- maps -->
```

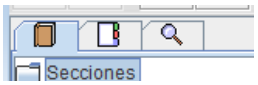
```
   <maps>
     <homeID>inicio</homeID>
     <mapref location="map.jhm" />
   </maps>

   <!-- views -->
   <view xml:lang="es" mergetype="javax.help.UniteAppendMerge">
      <name>TOC</name>
      <label>Table Of Contents</label>
      <type>javax.help.TOCView</type>
      <data>toc.xml</data>
   </view>

   <view xml:lang="es" mergetype="javax.help.SortMerge">
      <name>Index</name>
      <label>Index</label>
      <type>javax.help.IndexView</type>
      <data>indice.xml</data>
   </view>

   <view xml:lang="es">
      <name>Buscar</name>
      <label>Buscar</label>
      <type>javax.help.SearchView</type>
      <data engine="com.sun.java.help.search.DefaultSearchEngine">
         JavaHelpSearch
      </data>
   </view>

</helpset>
```

Dentro de este archivo encontramos diferentes bloques:

MAPS

Mapa de archivos HTML. Este bloque está declarado a través de la etiqueta maps y sirve para indicar el nombre del archivo que contiene el mapa de archivos HTML (mapref) y cuál será la página de inicio (homeID).

VIEW



Bajo esta etiqueta podemos declarar el contenido del panel izquierdo de la ayuda. E la imagen de la derecha se aprecia una serie de pestañas, pues bien, estas son las vistas que podemos definir.

Con la etiqueta type definimos el típo de elemendo de que se trata:

- javax.help.TOCView
- javax.help.IndexView
- javax.help.SearchView

El resto de etiquetas es bastante intuitivo, por lo que no se vana explicar.

1.3. Generación de búsqueda

Aunque esto forma parte del apartado anterior, se va a explicar a parte ya que es algo que se puede hacer de forma opcional y se haría de forma independiente.

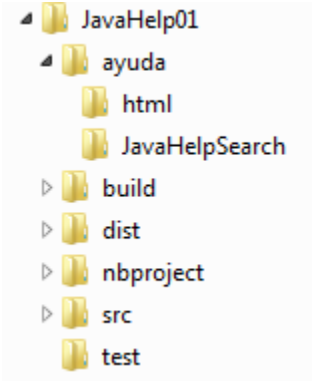
Nuestro sistema de ayuda puede permitir buscar información a través de un cuadro de texto en el que introduciremos los términos de la búsqueda. Esto se hace indexando o creando un índice del contenido de las páginas de ayuda.

Para crear el índice debemos utilizar jhindexer que se encuentra en D:\javahelp\bin. A este ejecutable sólo hay que pasarle como parámetro el directorio en el que se encuentran las páginas de nuestra ayuda. Con esto nos genera un directorio con nombre JavaHelpSearch dentro de nuestro esquema de ayuda.

```
:\>d:\javahelp\bin\jhindexer html
```

El comando jhindexer se ha ejecutado en el cmd dentro del directorio ayuda.

El directorio ayuda es el que contiene todos los archivos y directorios de nuestra ayuda que ya hemos mencionado.



2. Integración de la ayuda con la aplicación

Una vez que ya hemos creado todos los archivos, solo resta integrarla con nuestra aplicación de forma que pueda ser utilizada desde esta.

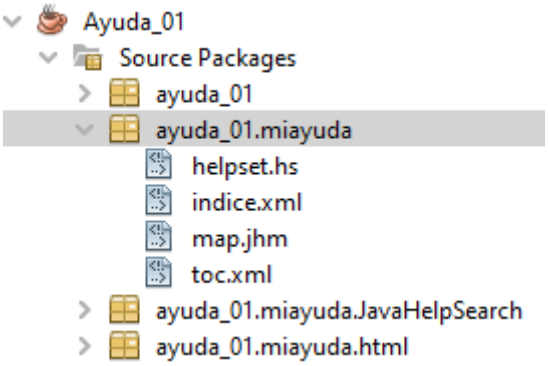
Proyecto Netbeans con ANT

Debemos incluir en nuestro proyecto la biblioteca JavaHelp, y esto se hace añadiendo javahelp\lib\jhall.jar.

Una característica de JavaHelp es que la ayuda no se va a mostrar al pulsar un botón de forma que se metería el código dentro de este, sino que este código va en el constructor de la clase y en el código se indica que la ayuda aparecerá cuando se pulse, en este caso, el elemento del menú jMenuItem (marcado en azul celeste).

El archivo que debemos pasar a nuestro código es el que hemos definido como hs (color rosa).

```
String AYUDA_HS = "ayuda_01/miayuda/helpset.hs";
try {
   ClassLoader cl = getClass().getClassLoader();
   HelpSet helpset = new HelpSet(cl, cl.getResource(AYUDA_HS));
   HelpBroker hb = helpset.createHelpBroker();
   JHelp jhelp = new JHelp(helpset);
   jhelp.setCurrentID("inicio");
   hb.enableHelpOnButton(jMenuItem1, "inicio", helpset);

} catch (HelpSetException ex) {
   System.err.println("Error al cargar la ayuda: " + ex);
}
```

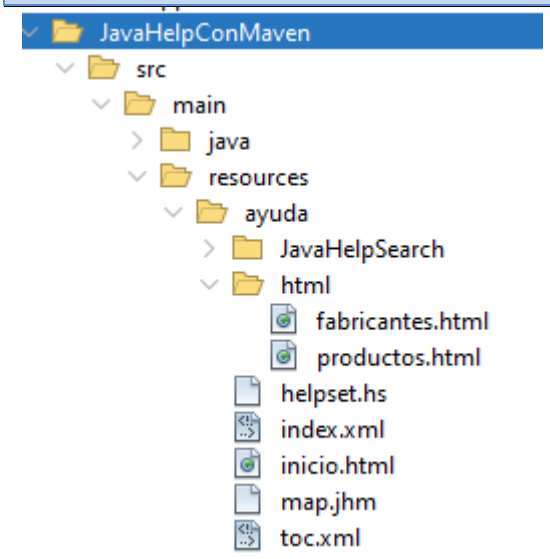



```
jMenuItem1.setText("Ayuda");
jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem1ActionPerformed(evt);
    }
});
```

PROYECTO NETBEANS CON MAVEN

En un proyecto con el gestor Maven podemos optar por frdvstdescargarnos las dependencias desde el repositorio de Maven o descargarnoa la biblioteca y añadirla a un repositorio loca. En este caso vamos a optar por descargar la biblioteca desde el respositorio de Maven. Para hacerlo debemos añadir a pom.xml la dependencia.

```
<dependency>
    <groupId>javax.help</groupId>
    <artifactId>javahelp</artifactId>
    <version>2.0.05</version>
</dependency>
```



Como ya se ha mencionado con anterioridad, en Maven los archivos que no son java se deben colocar a parte, en el directorio de recursos “resources”. La imagen de la derecha muestra el lugar de los archivos y directorios.

Con referencia al código, en Maven es igual, solamente hay que adaptar la ruta del archivo helpset.hs. Según la estrucdura de archivos y directorios de la imagen izquierda, la ruta del archivo es la siguiente:

```
String AYUDA_HS = "ayuda/helpset.hs";
```