

# Desarrollo de interfaces:

## 6. DOCUMENTACIÓN DE APLICACIONES

Jose Alberto Benítez Andrades

[jose@indipro.es](mailto:jose@indipro.es)

[www.indipro.es](http://www.indipro.es)

@indiproweb

@jabenitez88



# DOCUMENTACIÓN DE APLICACIONES – Contenido

- Ficheros de ayuda. Formatos.
- Herramientas de generación de ayudas.
- Ayuda genérica y sensible al contexto.
- Tablas de contenidos, índices, sistemas de búsqueda, entre otros.
- Incorporación de la ayuda a la aplicación.
- Tipos de manuales: anual de usuario, guía de referencia, guías rápidas, manuales de instalación, configuración y administración. Destinatarios y estructura.
- Confección de tutoriales multimedia. Herramientas de captura de pantallas y secuencias de acciones.
- Herramientas para la confección de tutoriales interactivos; simulación.

# FORMATOS DE FICHEROS DE AYUDA EXISTENTES EN FUNCIÓN DEL LENGUAJE DE PROGRAMACIÓN

# FORMATOS DE FICHEROS DE AYUDA

- **HLP**
- El estándar por excelencia para Windows.
- Desde sus primeras versiones adoptó este formato, (RTF y Word), y extensión de archivo, (HLP), para sus ayudas.
- Con el tiempo, hizo público este formato y proporcionó un compilador, (bajo MS/DOS), para generar esta clase de ayudas.
- El compilador evolucionó hasta llegar al actual HTML Help WorkShop con el que es posible compilar los dos formatos actuales, (HLP y CHM).

# FORMATOS DE FICHEROS DE AYUDA

- **HLP**
- Generar ayudas con este sistema es muy sencillo. Las herramientas necesarias son :
  - Word para generar el cuerpo de la ayuda, (el texto), en formato RTF, (no DOC).
  - El bloc de notas o el HTML Help WorkShop para generar los archivos de configuración y automatización de la ayuda, (extensiones HPJ y CNT).
  - El compilador HTML Help WorkShop para generar el archivo HLP resultante de la combinación de los archivos descritos anteriormente, (RTF, HPJ y CNT).

# FORMATOS DE FICHEROS DE AYUDA

- **CHM**
- Con la aparición de Windows 98, Microsoft creó un nuevo estándar de ayudas.
- Los archivos CHM son un sistema de ayudas cuya principal diferencia con su antecesor, (el HLP), es que no utiliza el formato RTF para generar los cuerpos de la ayuda sino que, claro está, utiliza HTML.

# FORMATOS DE FICHEROS DE AYUDA - HLP

- HLP utiliza un “lenguaje” de marcas en un archivo con formato RTF creado con Word.
- Para crear una ayuda es necesario conocer dichas marcas. Dichas marcas se referencian en el texto como notas al pie y, en esas notas se insertan informaciones tales como :
  - ❖ Cadenas de búsqueda.
  - ❖ Nombres de variable para crear hipervínculos.
  - ❖ Apuntes de índice y contenido.
- El archivo resultante con RTF es compilado con la ayuda de dos archivos mas, que se generan en un editor de textos sencillo, (bloc de notas). Estos archivos :
  - ❖ Dan información al compilador y configuran el HLP resultante, (HPJ).
  - ❖ Crean el índice de contenido de la ayuda, (CNT)

# FORMATOS DE FICHEROS DE AYUDA - CHM

- CHM utiliza un lenguaje de marcas completamente diferente al famoso RTF. HTML es un lenguaje muy probado y nos ofrece mas posibilidades y versatilidad que RTF.
- Una vez creados los diferentes capítulos de nuestra ayuda en diferentes archivos HTML, estos son aglutinados en un archivo de proyecto con extensión HHP. A este proyecto se le añaden :
  - ❖ Archivos H : Añaden un ID para identificador de capitulo dado, que luego será aprovechado por nuestras aplicaciones Visual Basic.
  - ❖ Archivos HHK : Indices.
  - ❖ Archivos HHC : Tablas de contenido.
- Con todo esto, compilamos el proyecto y obtenemos un archivo de ayuda con extensión CHM.



# FORMATOS DE FICHEROS DE AYUDA - CHM

- **Archivos HHC** En un archivo HHC se define la estructura del libro de contenidos de nuestra ayuda. El formato de un archivo de contenido es HTML.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
```

```
<HTML>
```

```
<HEAD>
```

```
<meta name="GENERATOR" content="Microsoft® HTML Help Workshop 4.1">
```

```
<!-- Sitemap 1.0 -->
```

```
</HEAD><BODY>
```

```
<OBJECT type="text/site properties">
```

```
    <param name="Window Styles" value="0x800025">
```

```
    <param name="ImageType" value="Folder">
```

```
</OBJECT>
```

```
<UL>
```

# FORMATOS DE FICHEROS DE AYUDA - CHM

```
<LI> <OBJECT type="text/sitemap">
    <param name="Name" value="Principal">
    </OBJECT>
<UL>
    <LI> <OBJECT type="text/sitemap">
        <param name="Name" value="Principal">
        <param name="Local" value="PruebaHTML1.htm">
        <param name="Comment" value="Comentario">
        <param name="ImageNumber" value="10">
        </OBJECT>
    <LI> <OBJECT type="text/sitemap">
        <param name="Name" value="Capitulo I">
        <param name="Local" value="PruebaHTML2.htm">
        </OBJECT>
    </UL>
</UL>
</BODY></HTML>
```

# FORMATOS DE FICHEROS DE AYUDA - CHM

- **Archivos HHK** Un archivo de índices también está descrito en formato HTML. En el referenciamos todas las cadenas de búsqueda para cada capítulo.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
```

```
<HTML>
```

```
<HEAD>
```

```
<meta name="GENERATOR" content="Microsoft® HTML Help Workshop 4.1">
```

```
<!-- Sitemap 1.0 -->
```

```
</HEAD><BODY>
```

```
<UL>
```

```
  <LI> <OBJECT type="text/sitemap">
```

```
    <param name="Name" value="Capítulo I">
```

```
    <param name="Name" value="Capítulo I">
```

```
    <param name="Local" value="PruebaHTML2.htm">
```

```
    <param name="URL" value="PruebaHTML2.htm">
```

```
  </OBJECT>
```

# FORMATOS DE FICHEROS DE AYUDA - CHM

```
<LI> <OBJECT type="text/sitemap">  
    <param name="Name" value="Principal">  
    <param name="Name" value="Capitulo I">  
    <param name="Local" value="PruebaHTML2.htm">  
    <param name="Name" value="Principal">  
    <param name="Local" value="PruebaHTML1.htm">  
    </OBJECT>  
</UL>  
</BODY></HTML>
```

# FORMATOS DE FICHEROS DE AYUDA - CHM

- **Archivos H**
- En un archivo H definimos, (con el comando #define), los alias que referenciaremos a cada capítulo, (archivos HTM), de nuestra ayuda.
- Un ejemplo de definición de estos alias seria :  
    #define IDH\_Top1 1  
    #define IDH\_Top2 2

# FORMATOS DE FICHEROS DE AYUDA - CHM

- **Compilar con Help WorkShop**
- Compilar con Help WorkShop es una tarea extremadamente sencilla. Simplemente hemos de cargar el archivo de proyecto, (HHP o HPJ dependiendo del formato de nuestra ayuda), e ir a la opción Compile del menú File.
- Aparecerá una ventana en que se nos pedirá confirmación del proyecto a compilar, (por defecto ofrece el que esté cargado en ese momento). En el proceso de compilación aparece una ventana de log en la que se nos informa del estado de dicha compilación. Si ocurre un error aparece reflejado en esta ventana.
- Otra utilidad práctica de Help WorkShop es la opción de View Compiled Help File del menú View. Con esta opción vemos cualquier archivo de ayuda; sea este CHM o HLP.

# AYUDA EN VBASIC

- **Inserción de la ayuda en Visual Basic.**
- En nuestras aplicaciones en Visual Basic es posible referenciar un archivo de ayuda en cualquiera de los formatos explicados, (HLP o CHM). La potencia de esta relación estriba en que en un control de nuestro formulario, (o en el propio formulario), es posible referenciar un capítulo concreto de nuestra ayuda. De esta forma, al pulsar F1 sobre ese control, se muestra la ayuda concreta sobre el tema referenciado para ese control.
- **Propiedades del proyecto.**
  - En la opción de Propiedades del proyecto del menú Proyecto de Visual Basic podemos determinar el archivo de ayuda para esa aplicación.
  - El archivo de ayuda destinado para la aplicación puede ser en formato HLP o CHM.

# AYUDA EN VBASIC

- Propiedad HelpContextID
  - Entre las propiedades del formulario o de cualquier control contenido en el hay una de nombre HelpContextID. Esta propiedad acepta cualquier valor numérico entero.
  - En el archivo H de la ayuda CHM o en la sección MAP de la ayuda HLP referenciamos estos números con los capítulos de la ayuda que deseemos.



# AYUDA EN VISUAL C++

- Agregar una matriz de pares identificador a la aplicación para asignar identificadores de controles a identificadores de contexto de ayuda. El último par debe ser de 0. Por ejemplo:

```
static const DWORD aMenuHelpIDs[] =  
{  
    ID_FILE_NEW, IDH_FILE_NEW ,  
    ID_FILE_OPEN, IDH_FILE_OPEN ,  
    ID_FILE_SAVE, IDH_FILE_SAVE ,  
    ID_EDIT_CUT, IDH_EDIT_CUT,  
    ID_EDIT_COPY, IDH_EDIT_COPY,  
    ID_EDIT_PASTE, IDH_EDIT_PASTE,  
    ID_FILE_PRINT, IDH_FILE_PRINT,  
    ID_APP_ABOUT, IDH_APP_ABOUT,  
    ID_FILE_PRINT_PREVIEW, IDH_FILE_PRINT_PREVIEW,  
    0, 0  
};
```

## AYUDA EN VISUAL C++

- Debajo de la matriz de pares de ID, declare una variable que determina el número total de pares de ID. Por ejemplo:

```
// Subtract 1 from the total number of ID pairs to account
```

```
// for the NULL pair at the end of the array.
```

```
DWORD numHelpIDs = (sizeof(aMenuHelpIDs)/sizeof(DWORD))/2 - 1
```

# AYUDA EN VISUAL C++

- Reemplace el método CWinApp:: WinHelp para llamar a la API de Ayuda HTML en lugar de WinHelp. Por ejemplo:

```
void CTestApp::WinHelp( DWORD dwData, UINT nCmd )
{
    DWORD i;
    switch (nCmd)
    {
        case HELP_CONTEXT:

            // If it is a help context command, search for the
            // control ID in the array.
```

# AYUDA EN VISUAL C++

```
for (i= 0; i < numHelpIDs*2; i+=2)
{
    if (aMenuHelpIDs[i] == LOWORD (dwData) )
    {
        i++; // pass the help context id to HTMLHelp
        HtmlHelp(m_pMainWnd->m_hWnd,"sample.chm",
            HH_HELP_CONTEXT,aMenuHelpIDs[i]);
        return;
    }
}
// If the control ID cannot be found,
// display the default topic.
if (i == numHelpIDs*2)
    HtmlHelp(m_pMainWnd->m_hWnd,"sample.chm",HH_DISPLAY_TOPIC,0);
break;
}
}
```

## AYUDA EN VISUAL C++

- Nota En el ejemplo anterior, debe ser Sample.chm en la misma carpeta que el archivo de proyecto de aplicación MFC, porque no se ha especificado ninguna ruta de acceso.
- Incluir un prototipo para la función WinHelp como sigue:

```
virtual void WinHelp( DWORD dwData, UINT nCmd = HELP_CONTEXT );
```

- Incluir un archivo que contiene las definiciones de ID de ayuda contextual. Los valores numéricos de estos Id deben coincidir con los valores definidos en la sección [MAP] del archivo de proyecto de Ayuda HTML. El siguiente es un ejemplo de cómo se define la Ayuda contextual de IDs:

## AYUDA EN VISUAL C++

- Incluir un archivo que contiene las definiciones de ID de ayuda contextual. Los valores numéricos de estos Id deben coincidir con los valores definidos en la sección [MAP] del archivo de proyecto de Ayuda HTML. El siguiente es un ejemplo de cómo se define la Ayuda contextual de IDs:

```
#define IDH_FILE_NEW 1
#define IDH_FILE_OPEN 2
#define IDH_FILE_SAVE 3
#define IDH_EDIT_CUT 4
#define IDH_EDIT_COPY 5
#define IDH_EDIT_PASTE 6
#define IDH_FILE_PRINT 7
#define IDH_APP_ABOUT 8
#define IDH_FILE_PRINT_PREVIEW 9
```

# HERRAMIENTAS PARA GENERAR AYUDAS ( DR EXPLAIN, HTML HELP WORKS, CHMEDITOR )

# HERRAMIENTAS – DR EXPLAIN

- **¿Qué es Dr. Explain?**
- Dr. Explain es un software para crear archivos de ayuda, guías de usuario, manuales en línea y documentación de aplicaciones.
- Dr. Explain produce documentación en formatos HTML (manuales en línea), CHM (archivos de ayuda MS Windows®), RTF y PDF desde una sola fuente.
- **El programa analiza la aplicación y produce automáticamente las capturas de pantalla de las ventanas junto con la secuencia de llamadas explicativas para cada control.**



# HERRAMIENTAS – DR EXPLAIN

- **¿Cuáles son las características más útiles de Dr. Explain?**
  - ❖ La herramienta de captura de pantallas integrada, que analiza las ventanas o las páginas web de la aplicación y que automáticamente crea gráficos de las capturas de pantalla con anotaciones.
  - ❖ La herramienta de anotaciones para hacer ilustraciones técnicas sensacionales para los manuales de ayuda y la documentación de software.
  - ❖ Un editor de contenidos con muchas funcionalidades, optimizado para crear documentación de software.
  - ❖ La capacidad de ampliar el contenido con texto variable y con fragmentos de código HTML.
  - ❖ La creación de archivos de ayuda CHM, de manuales en línea en HTML, de documentos RTF y de documentación en formato PDF desde una única fuente.

# HERRAMIENTAS – DR EXPLAIN

- **¿Cuáles son las características más útiles de Dr. Explain?**
  - ❖ La actualización sencilla de ilustraciones cuando se lanza una nueva versión del producto. La herramienta de actualización de imágenes sabe reemplazar las imágenes subyacentes, al mismo tiempo que mantiene todos los metadatos y las anotaciones explicativas.
  - ❖ La compatibilidad con Help ID (“ID de ayuda”) para hacer archivos de ayuda que distinguen el contexto.
  - ❖ La capacidad de agregar funciones de búsqueda e índices de palabras clave a los manuales en línea sin necesidad de instalar programas, scripts ni bases de datos en el servidor.
  - ❖ La compatibilidad con idiomas multibyte y que se escriben de derecha a izquierda.
  - ❖ El seguimiento del avance del proyecto con estados de los temas y bloqueo.

# HERRAMIENTAS – DR EXPLAIN

- [Manual de ayuda](#)

# CONSTRUCCIÓN DE FICHEROS DE AYUDA PARA JAVA CON JAVAHELP

# JAVAHELP

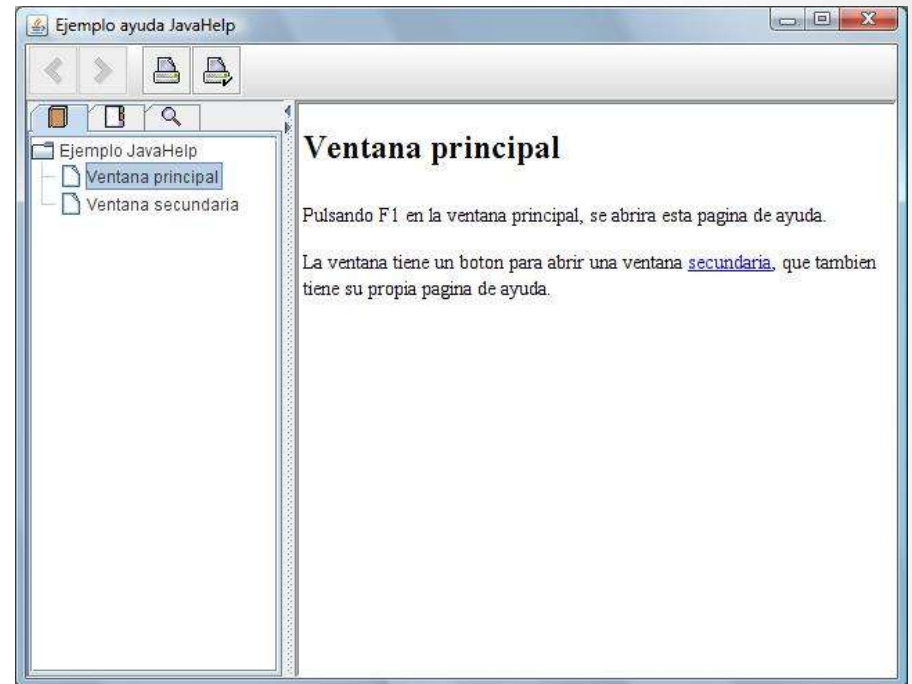
- [?] Qué es JavaHelp
- [?] Características Básicas
  - ❖ [?] Herramientas
  - ❖ [?] Librerías
- [?] Pasos en la construcción de la ayuda
- [?] Incorporación de la ayuda en la aplicación

## JAVAHHELP – QUÉ ES

- ¿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

# JAVAHHELP – QUÉ ES

- [?] Visor de Ayuda
  - ❖ [?] Panel de contenidos
  - ❖ [?] Panel de navegación
- [?] Tabla de contenidos
- [?] Índice
- [?] Búsqueda



# JAVAHELP – Herramientas

- hswviewer
  - ❖ ⓘ 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 jhindexe



## JAVAHELP – 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.

## JAVAHELP – Pasos Construcción de la Ayuda (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
  - ❖ [?] Indice
  - ❖ [?] ....

## JAVAHELP – Pasos Construcción de la Ayuda (detalles)

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).

# JAVAHELP – Organización

## ..\help

mi\_helpset.hs

mi\_map.jhm

mi\_toc.xml

mi\_index.xml

\ fuentes\

\JavaHelpSearch

**Topic1.html**

**Topic2.html**

**Topic3.html**

## JAVAHHELP – 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.

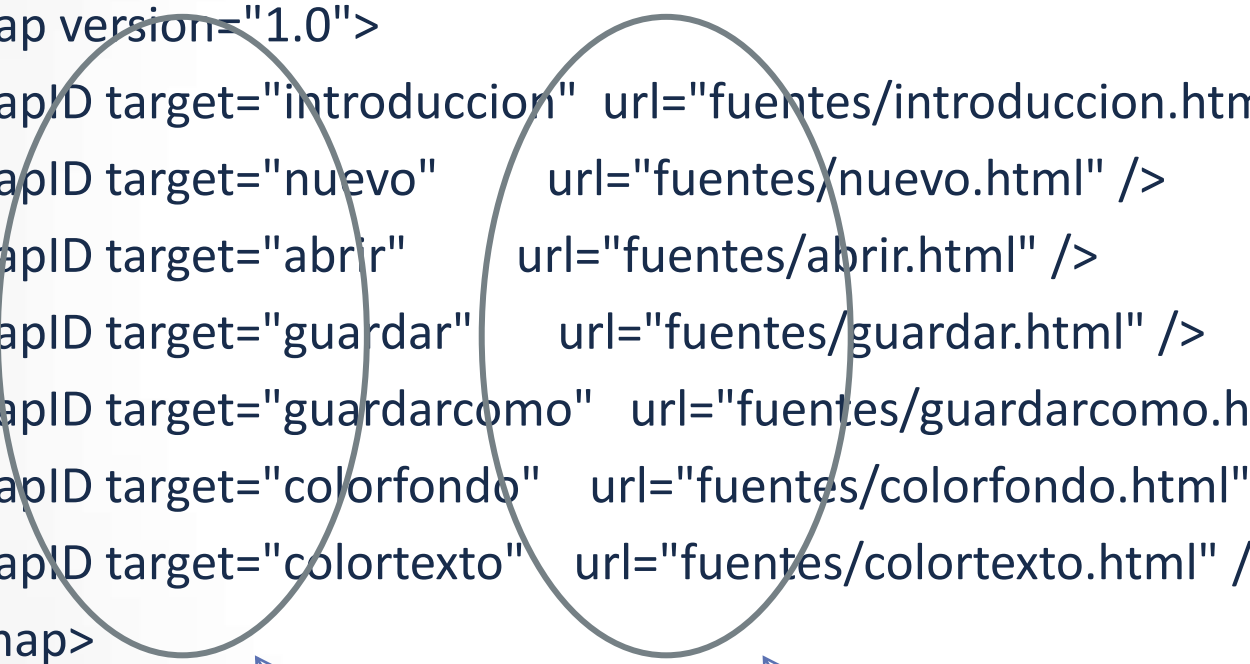
## JAVAHELP – 2. Fichero Map

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

## JAVAHHELP – 2. Fichero Map

<...

```
<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>
```



Identificadores

Fichero asociado html

## JAVAHELP – 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



## JAVAHELP – 3. 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

## JAVAHHELP – 3. Fichero HelpSet (II)

<.....

<helpset>

<title> Ejemplo de Ayuda </title>



Título

<maps>

<homeID>introduccion</homeID>



Página que visualiza por defecto

<mapref location="Map.jhm"/>



Fichero map

</maps>

<!-- views -->

<view>

<name>Tabla de contenidos</name> // nombre de la vista

<label>Tabla de Contenidos</label> // etiqueta asociada con la vista


<type>javax.help.TOCView</type> //camino a la clase navegador

<data>TOC.xml</data> //fichero que contiene la tabla de contenidos

</view> ....

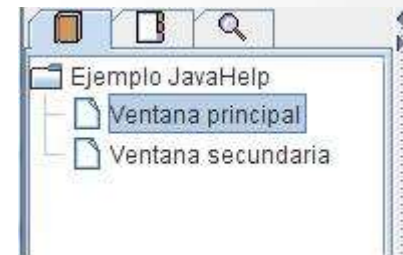
/helpset>

## JAVAHHELP – 4. FICHERO TOC

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

## JAVAHELP – 4. 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>
  </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

## JAVAHELP – 5. FICHERO Índice (I)

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

## JAVAHHELP – 5. 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

## JAVAHELP – 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

## JAVAHELP – 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



## JAVAHHELP – 6. 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

## JAVAHELP – 6. Búsqueda (III)

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

## JAVAHELP – 7. Construcción de fichero 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.

## JAVAHELP – 7. Construcción de fichero 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**

JAVAHHELP

# Incorporación de la Ayuda en la Aplicación

## JAVAHELP – Clases Básicas Implicadas

### ■ Clase URL

- ❖ ¿Paquete a importar: `import java.net.*;`
- ❖ ¿Esta clase se utiliza para representar una URL (UniformResource 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.

## JAVAHELP – 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

# JAVAHELP – 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



## JAVAHELP – Pasos para incorporar la ayuda en la aplicación

- Incorporar la librería jhall.jar a la aplicación
- Importar los paquetes que son necesarios en la aplicación
  - `import javax.help.*;`
  - `import java.net.*;` // para la clase URL
- 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;  
}
```

## JAVAHELP – Pasos para incorporar la ayuda en la aplicación

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

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

- Asociar la ayuda a un botón o elemento de menú

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

- 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.

## JAVAHELP – Pasos para incorporar la ayuda en la aplicación

- 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);`

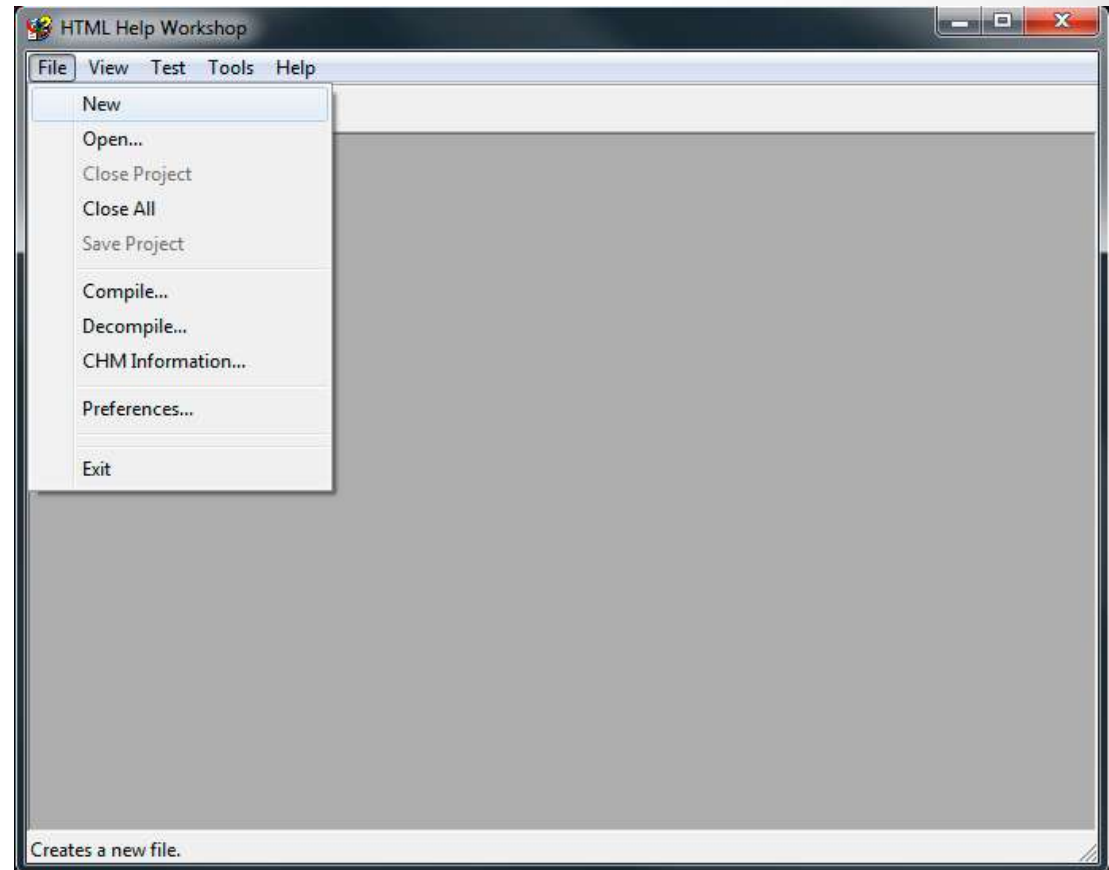
# CONSTRUCCIÓN DE FICHEROS DE AYUDA PARA C, C++, VISUAL BASIC, ETC CON HTML HELP WORKSHOP

## HTML HELP WORKSHOP

- USANDO HTML HELP WORKSHOP
- HTML Help Workshop es una aplicación diseñada para proporcionar un sistema simple para crear y administrar proyectos de ayuda y sus archivos relacionados de creación de ayuda.
- Puedes descargarlo de forma gratuita
- <http://www.microsoft.com/en-us/download/details.aspx?id=21138>
- HTML Help Workshop utiliza los componentes subyacentes de Microsoft Internet Explorer para mostrar el contenido de la ayuda.
- Es compatible con HTML, ActiveX, Java, secuencias de comandos (JScript y Microsoft Visual Basic Scripting Edition) de idiomas y formatos de imagen HTML (archivos .jpeg, .gif y .png).

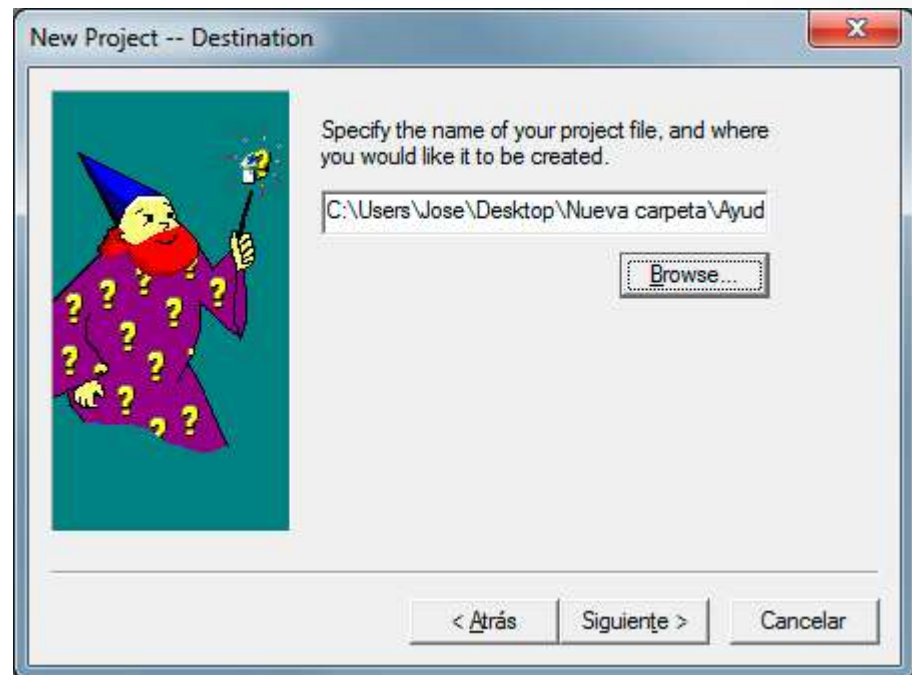
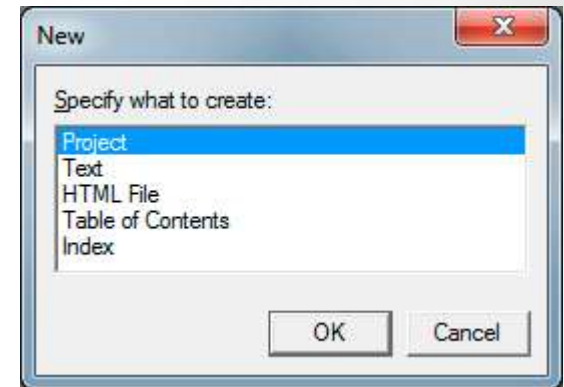
# HTML HELP WORKSHOP

- TUTORIAL BÁSICO  
HTML HELP  
WORKSHOP



## HTML HELP WORKSHOP

- Inicia el programa y selecciona la opción de nuevo proyecto en la barra de herramientas. En la ventana que aparece pon el nombre de tu "Proyecto" y elige la localización del mismo.



## HTML HELP WORKSHOP

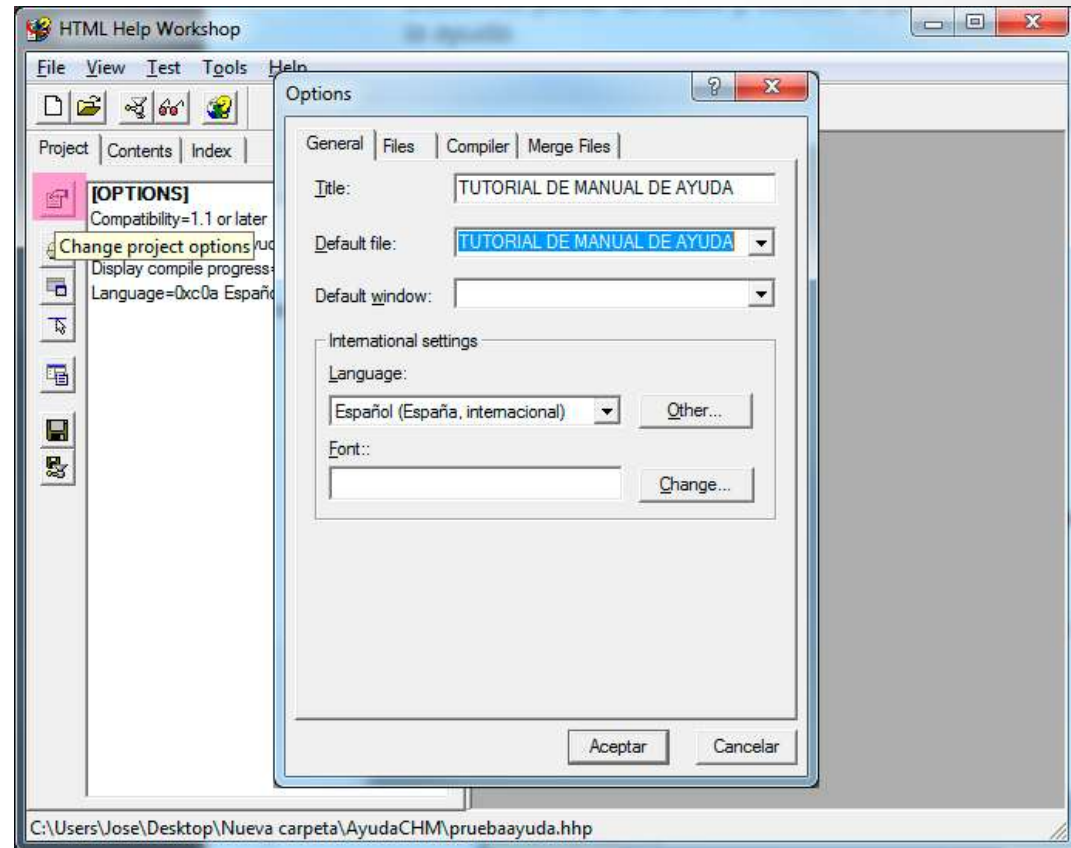
- Pulsa en siguiente y aparecerá esta ventana; como nuestro proyecto es nuevo no marcamos nada.





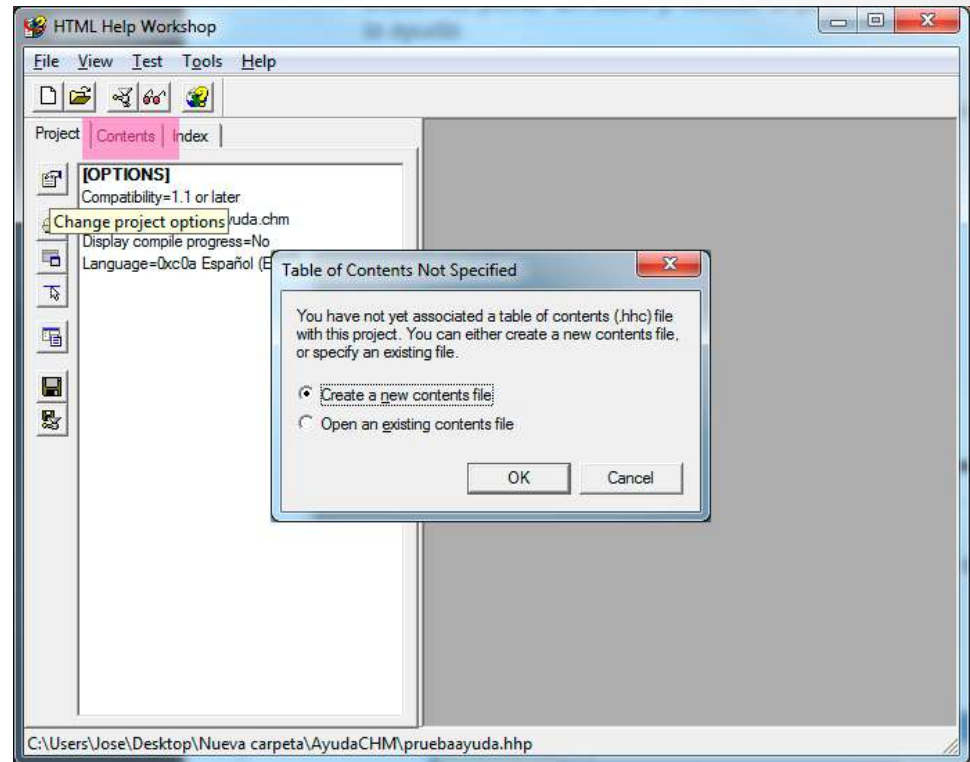
## HTML HELP WORKSHOP

- Te encuentras en la ventana principal de tu proyecto.
- Debemos poner un título y colocar la página por defecto en la que comenzara la ayuda.
- Presionamos el primer botón que aparece en la barra vertical en el lado izquierdo y ponemos el título.



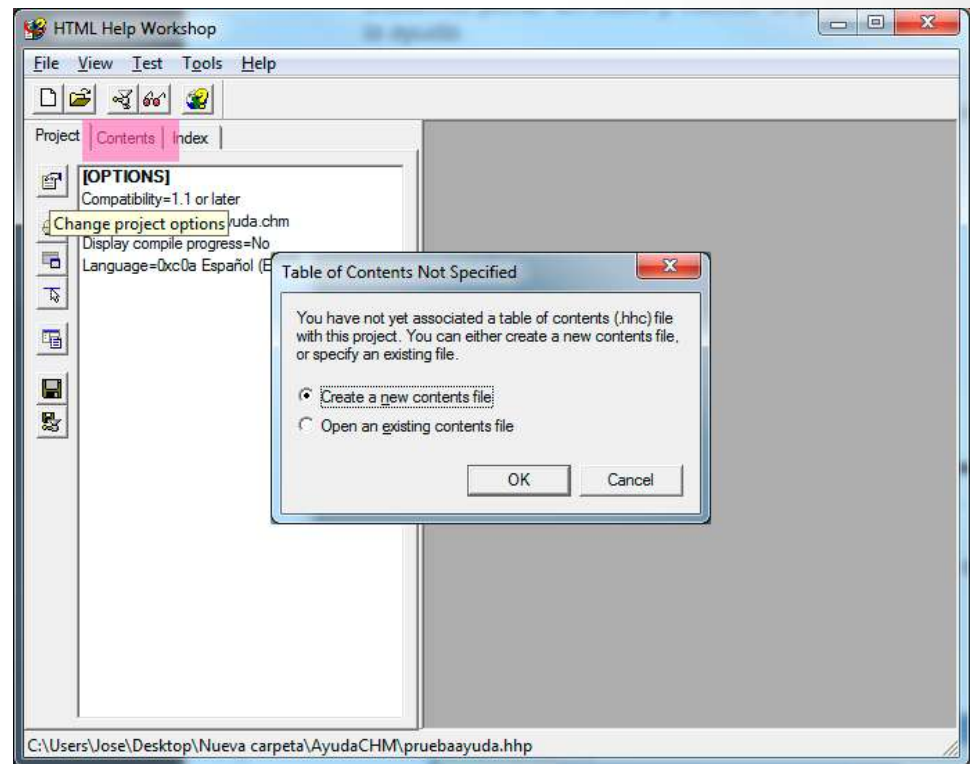
## HTML HELP WORKSHOP

- El campo “Archivo por defecto” (Default file), es tu archivo inicial del proyecto. Tienes que tener en cuenta que este archivo debe pertenecer al proyecto.
- Una vez cambiadas las opciones de proyecto, pulsaremos la pestaña Contents que se encuentra en el menú principal junto a la pestaña Project.



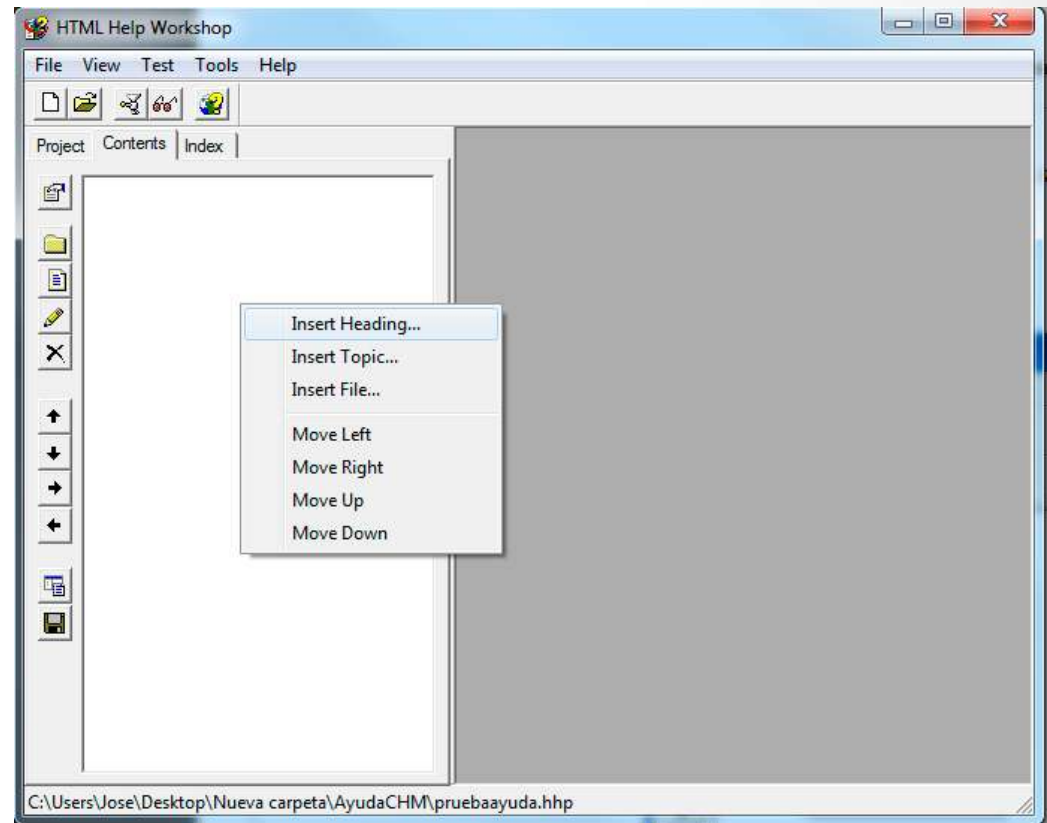
## HTML HELP WORKSHOP

- Marcaremos la opción Create a new contents file ya que no disponemos de ficheros. Aparecerá una ventana para guardar tu archivo de contenidos; en este caso lo guardare como tutorial.



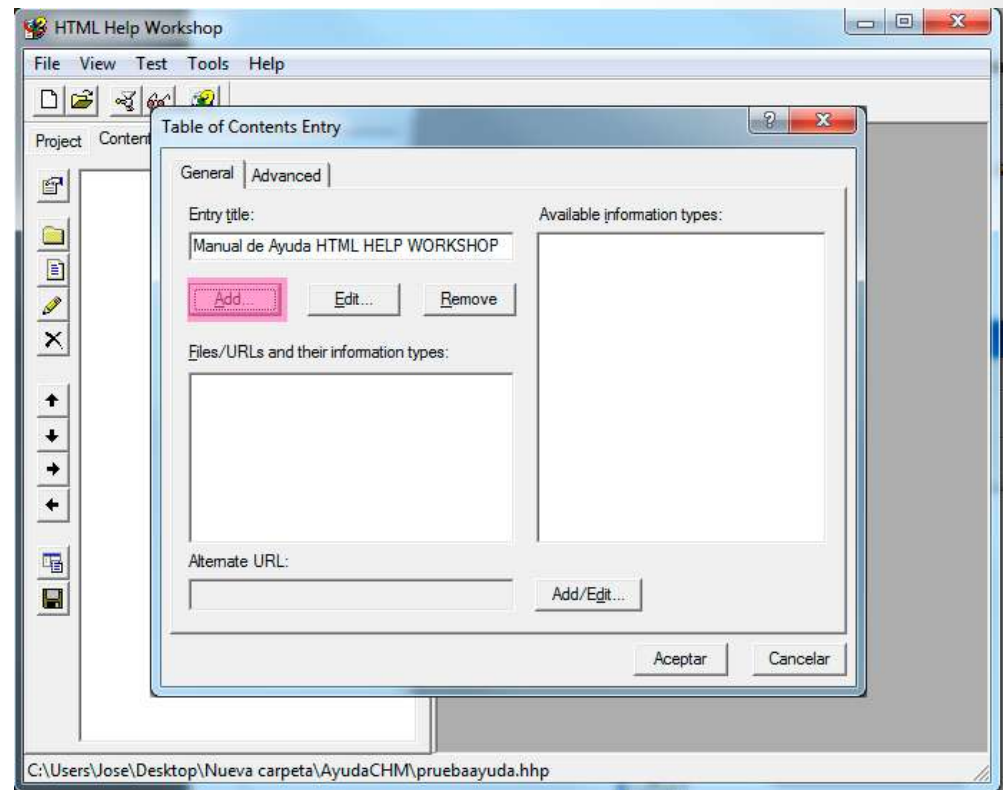
## HTML HELP WORKSHOP

- Pulsa con el botón derecho en el lugar blanco de contenidos y selecciona la primera opción del menú.
- Insert Heading crea una cabecera para tu archivo de contenidos, un HTML que contiene una descripción general de lo hace tu programa o del conjunto al que representa.

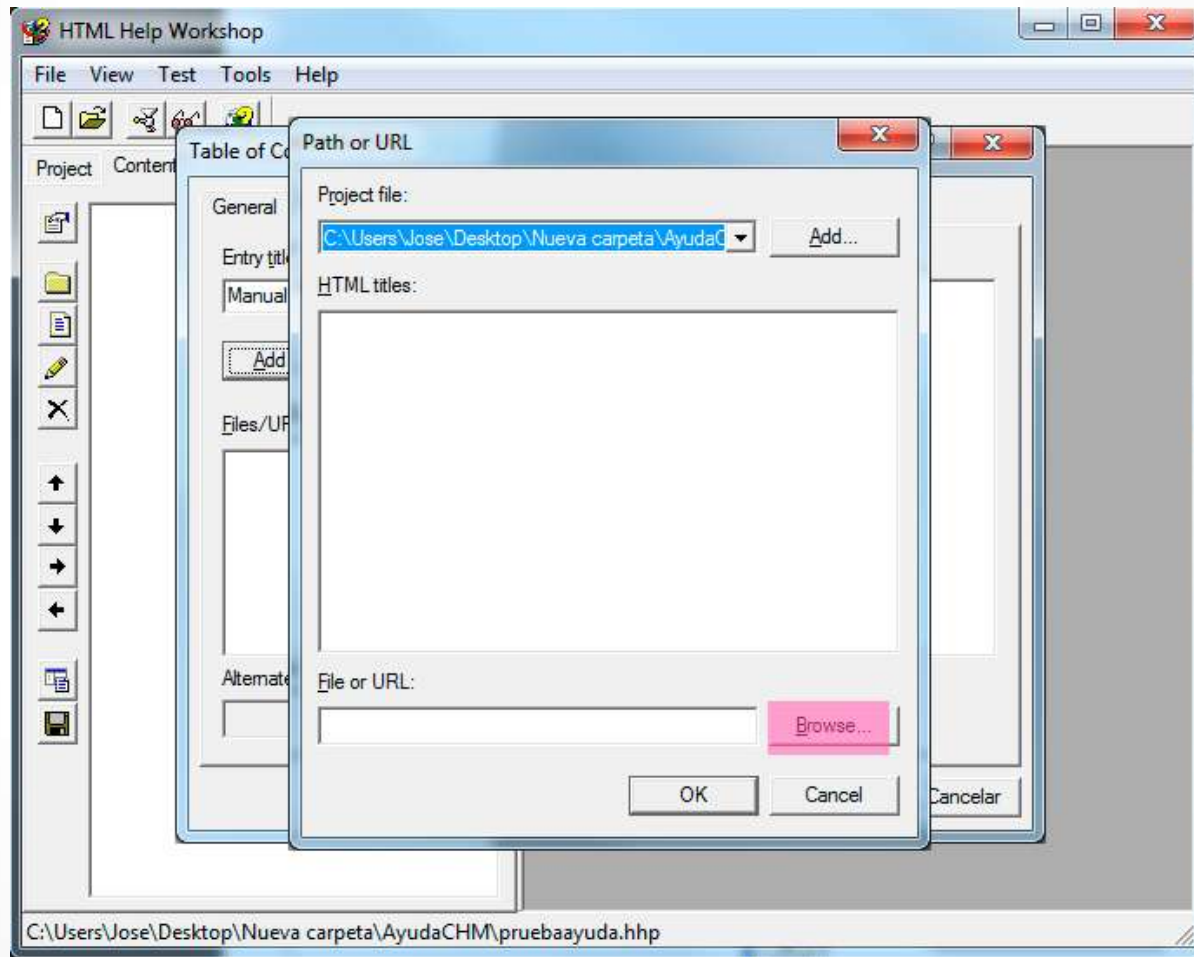


# HTML HELP WORKSHOP

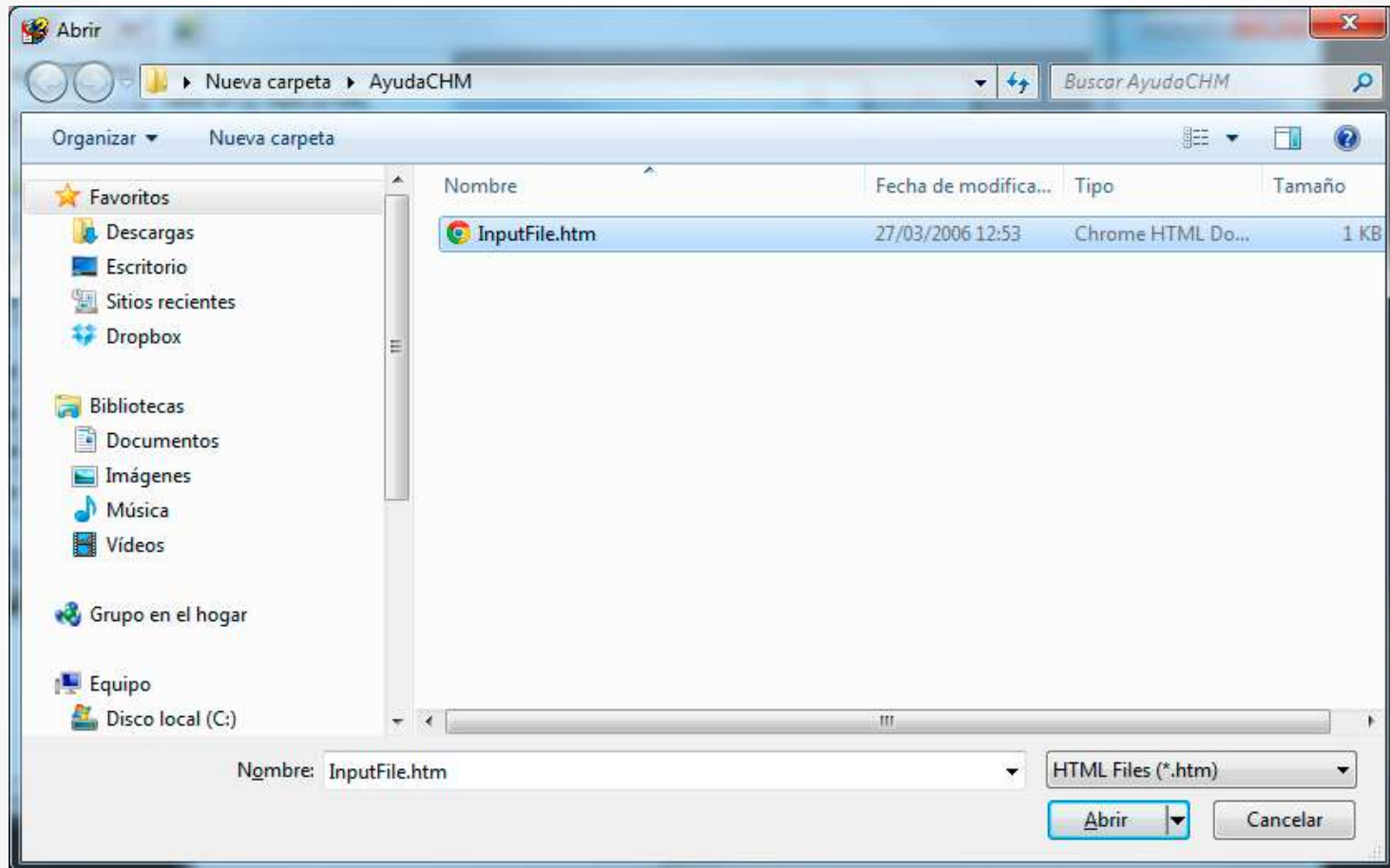
- Introduce a continuación el título que desees y pulsa add; se abrirá la ventana de la imagen. Pulsa sobre la opción browse y elige tu archivo HTML creado.



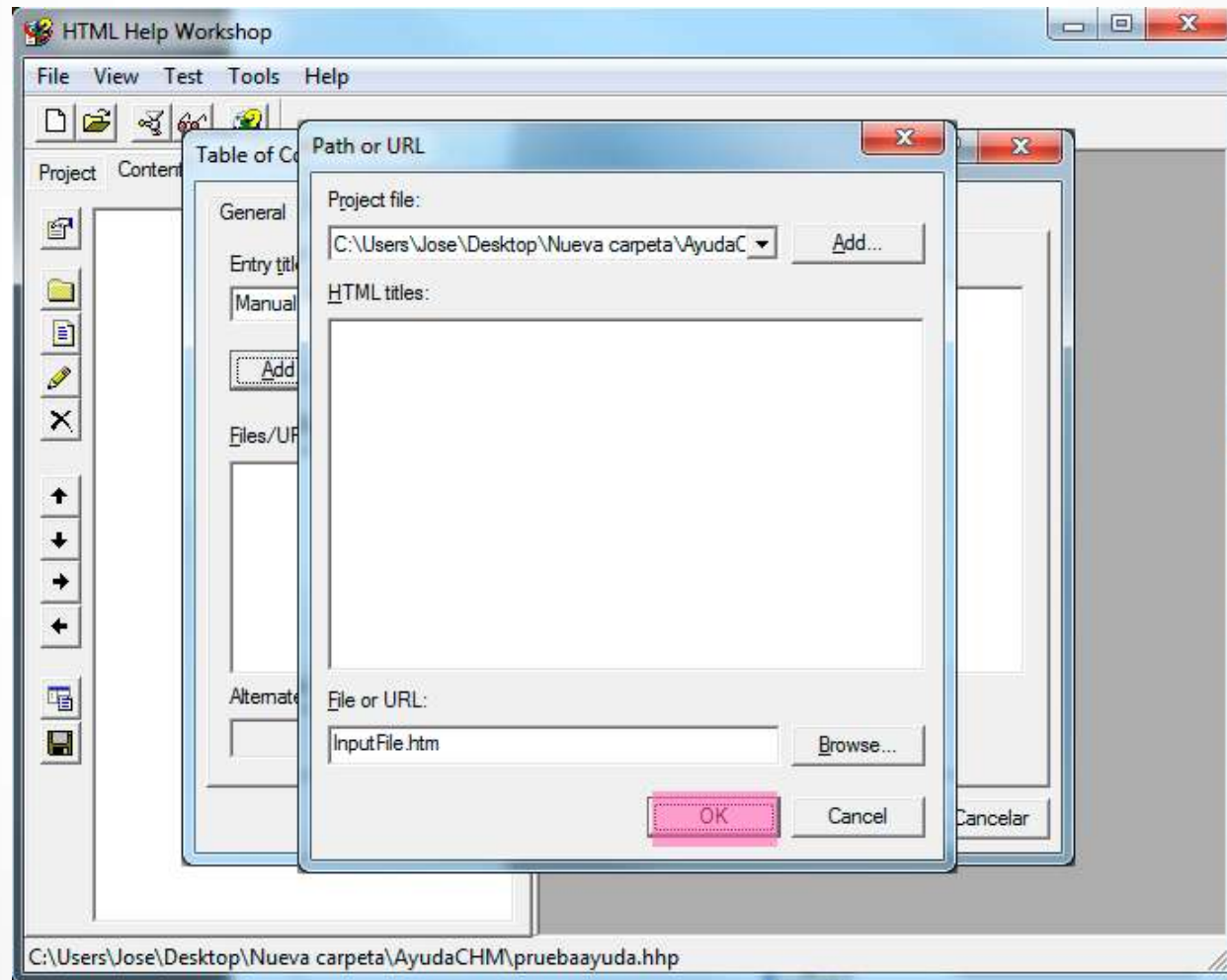
# HTML HELP WORKSHOP



# HTML HELP WORKSHOP

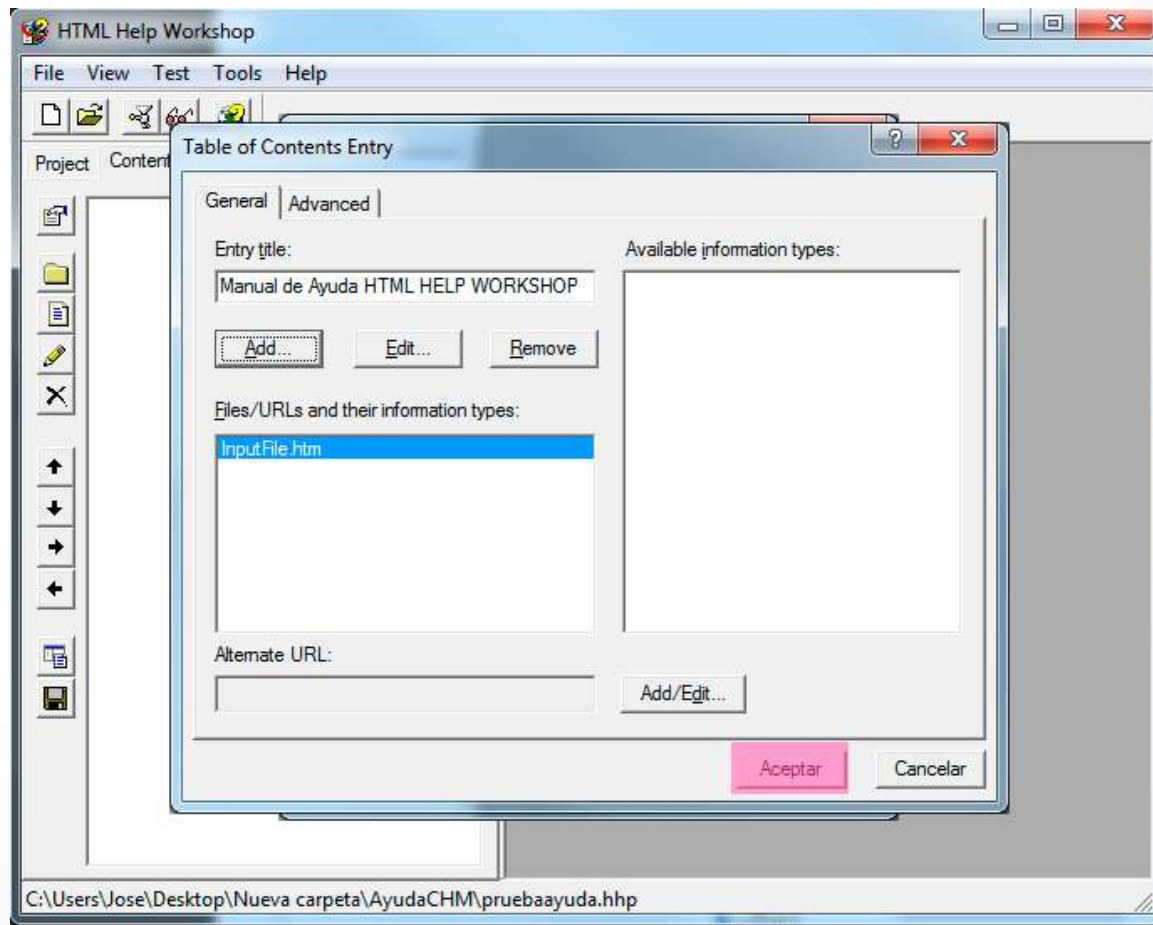


# HTML HELP WORKSHOP



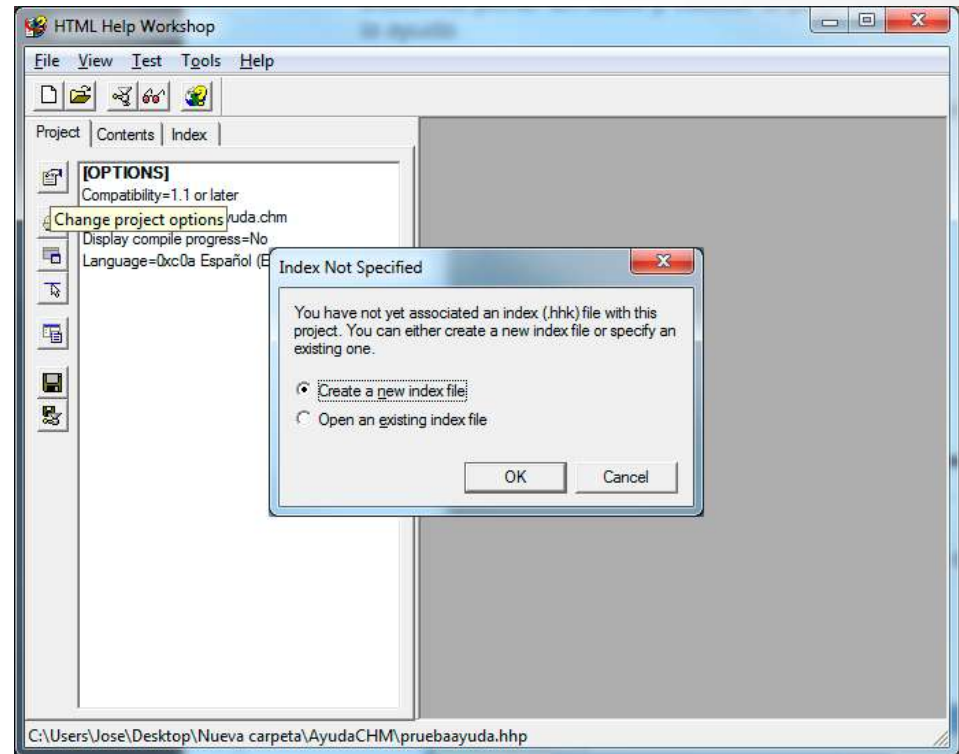


# HTML HELP WORKSHOP

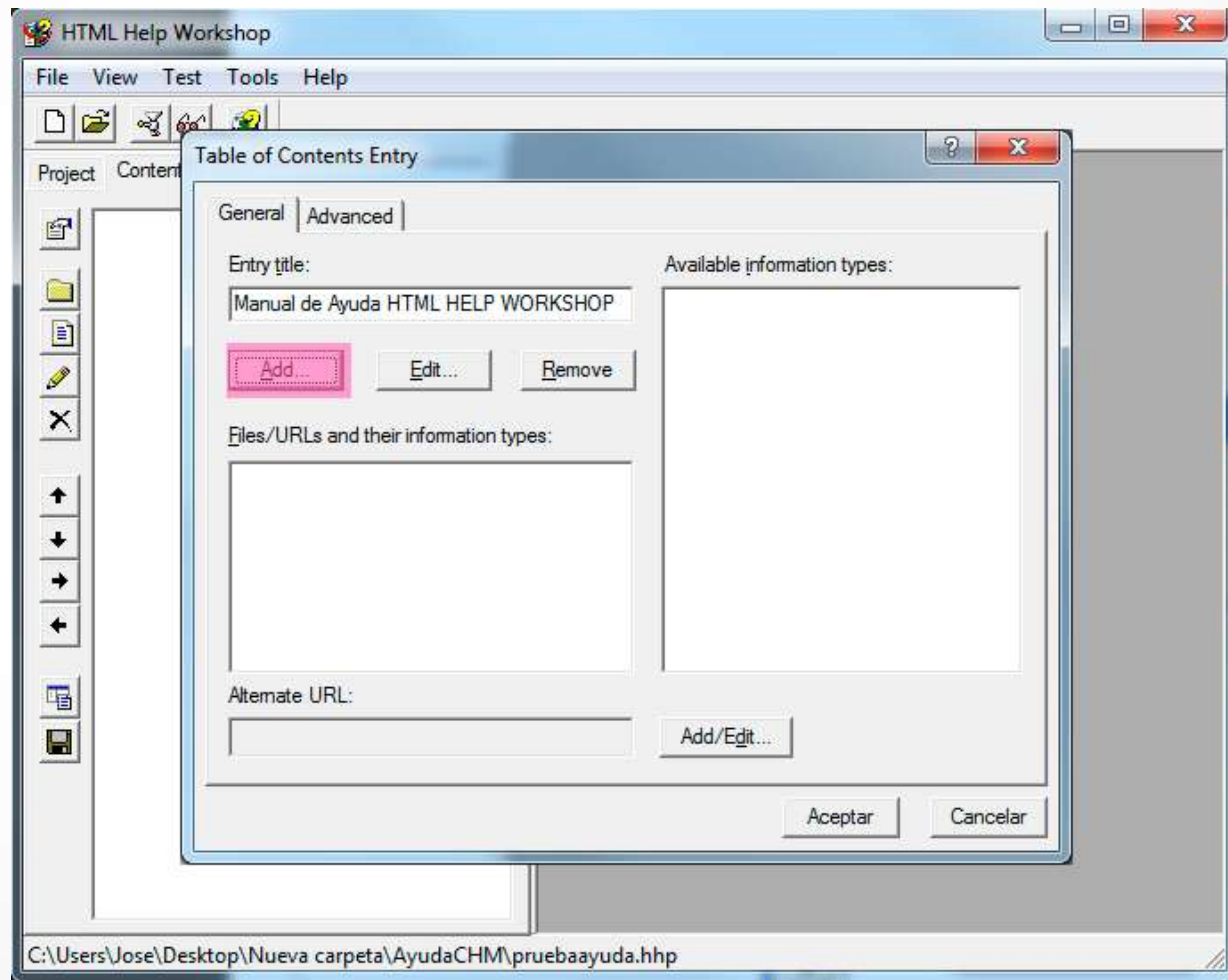


## HTML HELP WORKSHOP

- Puedes insertar HTML unos dentro de otros de forma sencilla escogiendo inserttopic y realizando el mismo proceso.
- Pulsa sobre la pestaña índices (index)

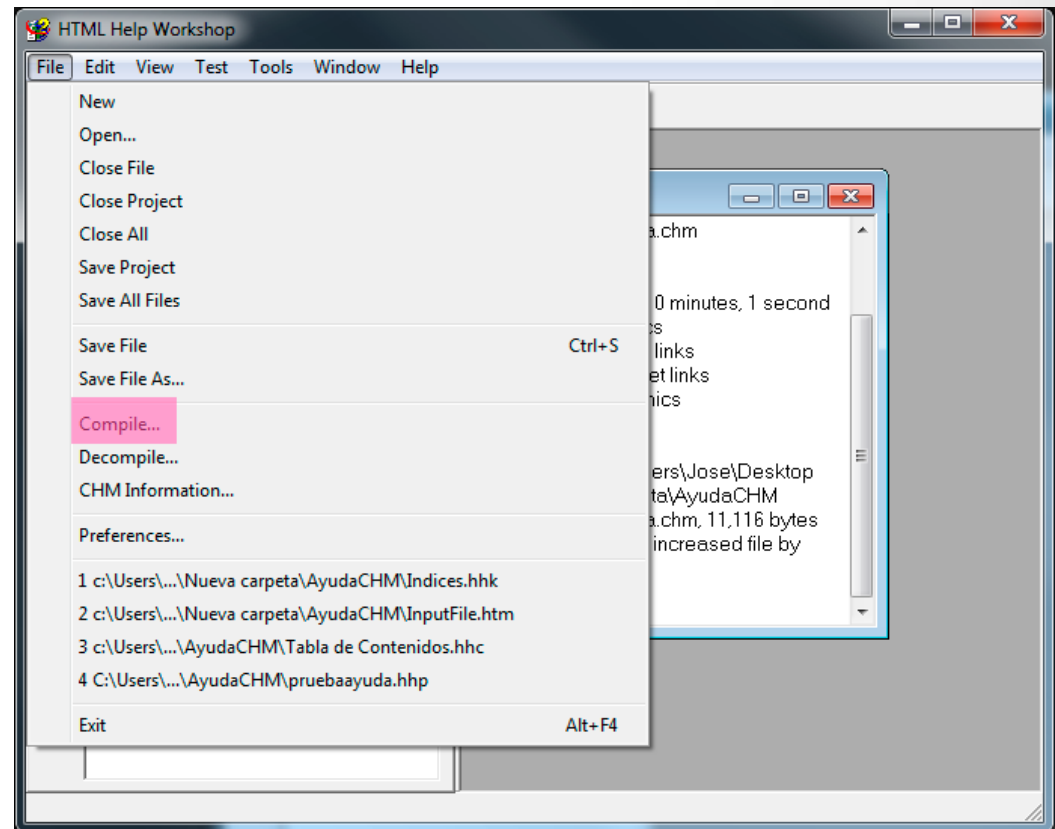


# HTML HELP WORKSHOP

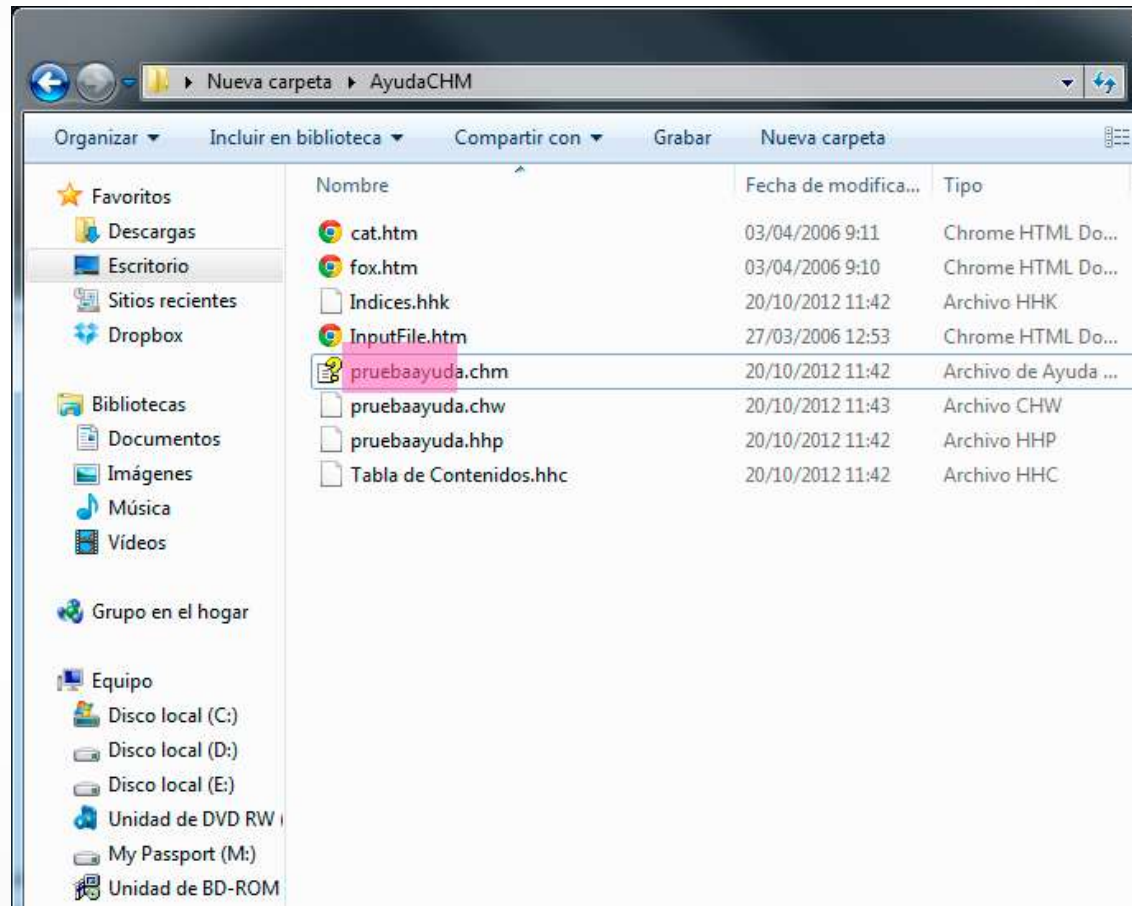


## HTML HELP WORKSHOP

- Compilamos el proyecto, y ya tendremos generado nuestro fichero .chm listo para ser insertado en nuestra aplicación.



# HTML HELP WORKSHOP



# HTML HELP WORKSHOP

