

Soporte al usuario

AYUDA E INSTALADOR

Soporte al usuario: Ayuda

La documentación de un producto forma parte de la interfaz de usuario

La documentación debe considerarse al diseñar los test de usabilidad

La documentación incluye:

- Información de la instalación
- Ayuda en línea
- Mensajes
- Tutoriales
- Cualquier otra información de soporte para el producto

Documentación básica

Se considera documentación básica para el usuario, o soporte básico:

- Guía de Referencia Rápida
 - Recordatorio al usuario de determinadas acciones con herramientas con las que ya está familiarizado
- Ayuda en línea
 - Solicitada por ejemplo cuando el usuario ha encontrado un problema al realizar una tarea
- Explicación Completa (Manual)
 - Explicación completa sobre una herramienta o comando
- Tutorial
 - Particularmente útil para los nuevos usuarios de una herramienta. Proporciona instrucciones paso-a-paso de cómo usar la herramienta

Características de la Ayuda (I)

Disponibilidad

- El usuario ha de poder acceder a la ayuda en cualquier momento de la interacción con el sistema sin tener que salir de la aplicación.

Precisión y Detalle

- La ayuda ha de estar acorde con la versión actual del producto.
- Ha de abarcar todo el sistema, ya que no se conoce en qué partes del mismo podrá requerir ayuda el usuario.

Consistencia

- Todas las partes de la ayuda (y la documentación) han de ser consistentes en términos de Contenidos, Terminología y Estilo.

Características de la Ayuda (I)

Robustez

- Es importante que la ayuda proporcione un buen soporte incluso cuando la aplicación no se comporta correctamente

Flexibilidad

- En muchos sistemas la ayuda es rígida, es decir, se muestra siempre el mismo mensaje independientemente de la experiencia de la persona que busca la ayuda.
- La ayuda ha de permitir que los diferentes tipos de usuarios interaccionen de manera apropiada según sus necesidades.
- Para ello se emplea **hipertexto**

No Obstructiva

- La ayuda no se debe interponer en el uso normal de la aplicación

Ayuda en Línea

A la hora de planificar el desarrollo de la ayuda en línea hay que tener en cuenta:

- Audiencia de la aplicación
- Temas o epígrafes a tratar en la ayuda
- Contenidos de los temas de la ayuda
- Estructura de los mismos
 - Referencias cruzadas
 - Definición de términos
- Uso de ayuda sensible al contexto

Audiencia de la Aplicación

El usuario al que va dirigida la aplicación determina la información que tiene que aparecer en el sistema de ayuda y como ha de ser presentada:

- Usuario Novel
 - Completamente inexperto en informática
- Usuario Novel en la aplicación
 - Tiene algunos conocimientos de informática, pero completamente nuevo en la aplicación
- Usuario Intermedio en la aplicación
 - Tiene conocimientos de la aplicación
- Usuario Experto en la aplicación
 - Tiene una amplia experiencia.

Contenidos de las Preguntas

- Los usuarios noveles necesitan la ayuda para aprender las tareas y las definiciones de los términos.
 - Ej. Si se trata de un juego será necesario explicarle las reglas del juego así como la forma de jugar mediante la aplicación.
- Los usuarios intermedios acudirán a la ayuda más a modo de recordatorio de órdenes y funciones.
- El experto solamente pide ayuda en la sintaxis de órdenes y funciones, teclas aceleradoras o atajos para desenvolverse más rápidamente.

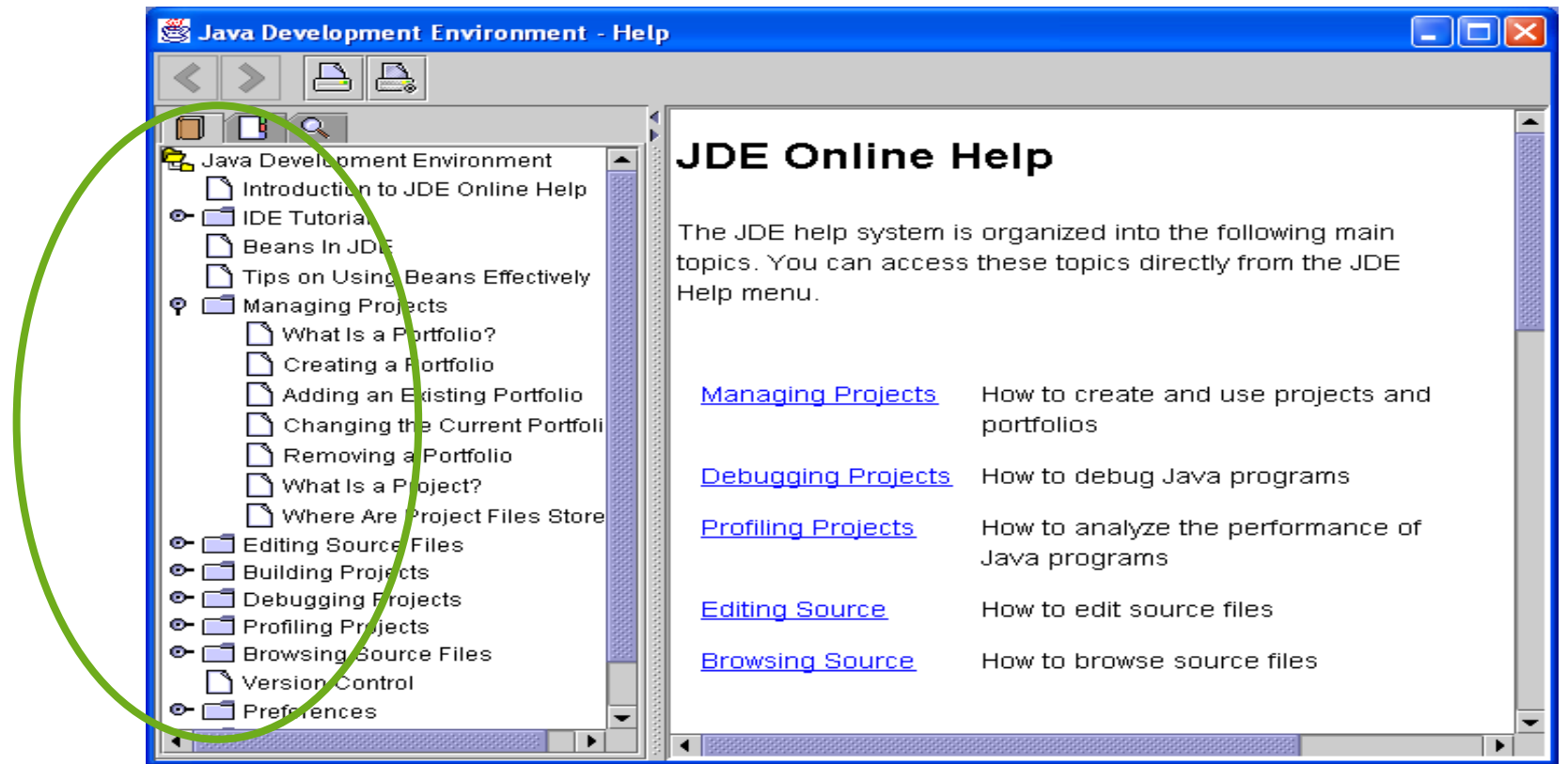
Estructura de la Ayuda

- Organización Jerárquica
 - Tabla de contenidos
 - Índice
- Recorrido Secuencial
- Referencias Cruzadas
- Búsqueda

Organización Jerárquica

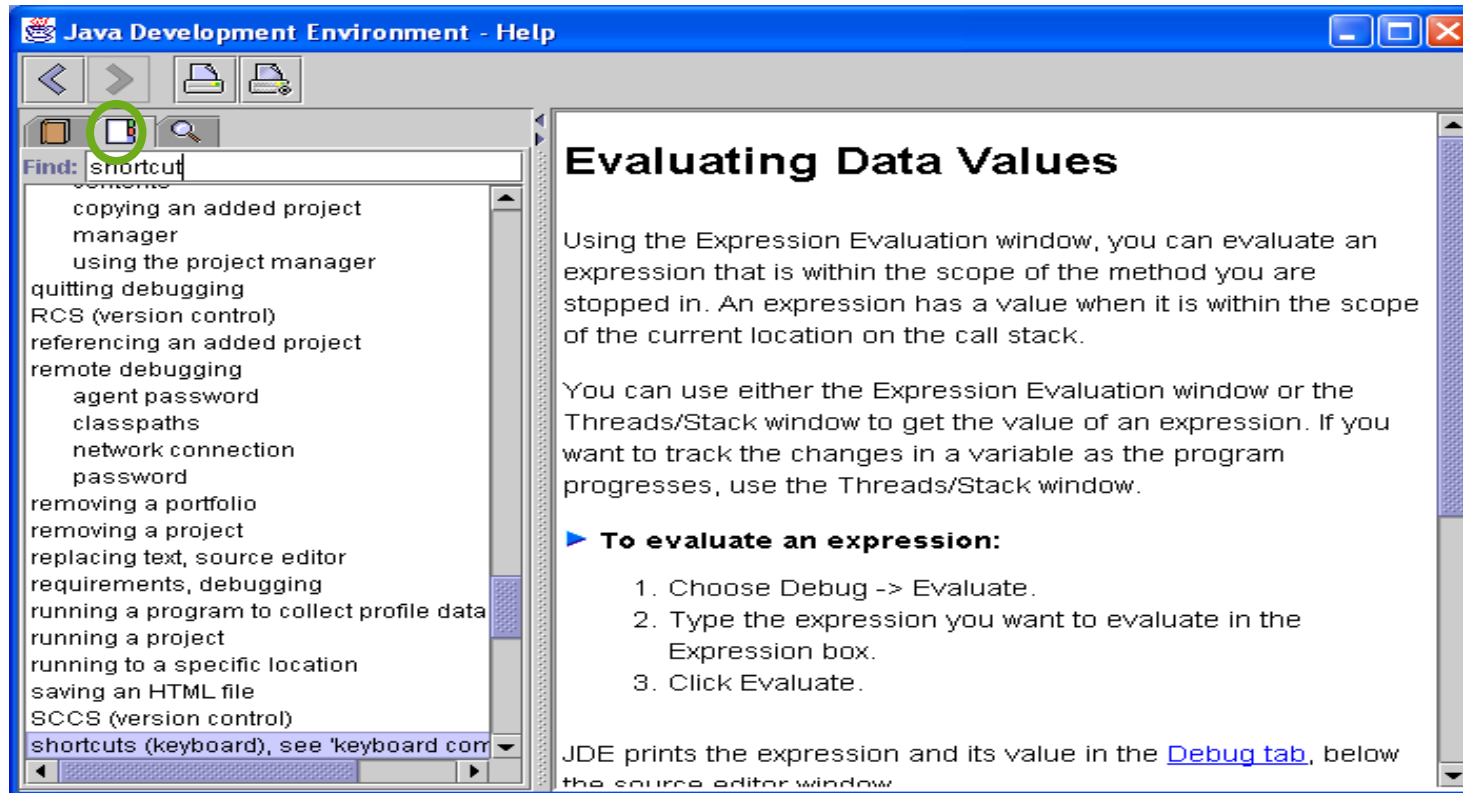
Tabla de contenidos

- ▶ Tanto la tabla de contenidos como el índice proporcionan una lista de todas las cuestiones a disposición del usuario



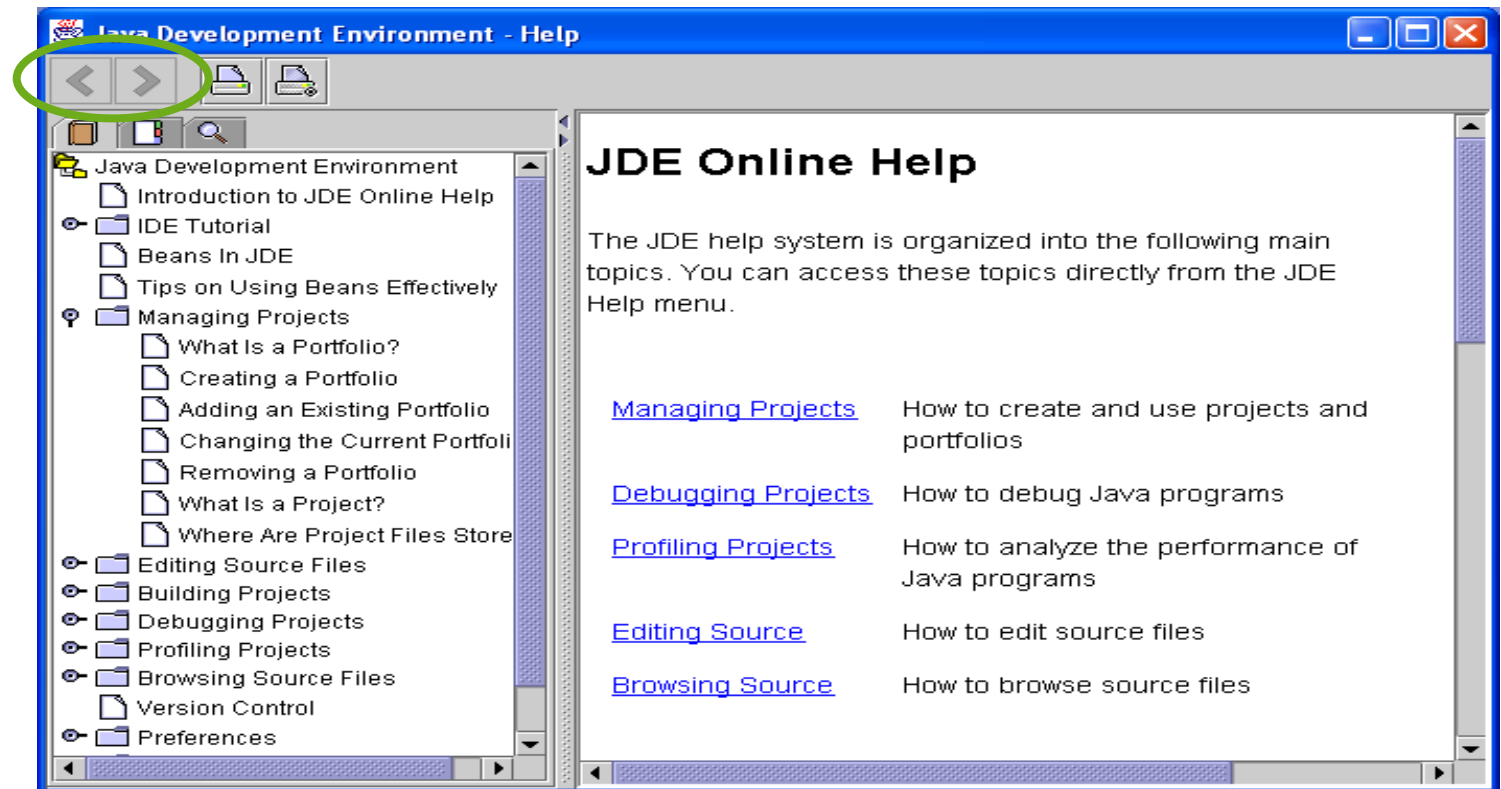
Organización Jerárquica

Indice



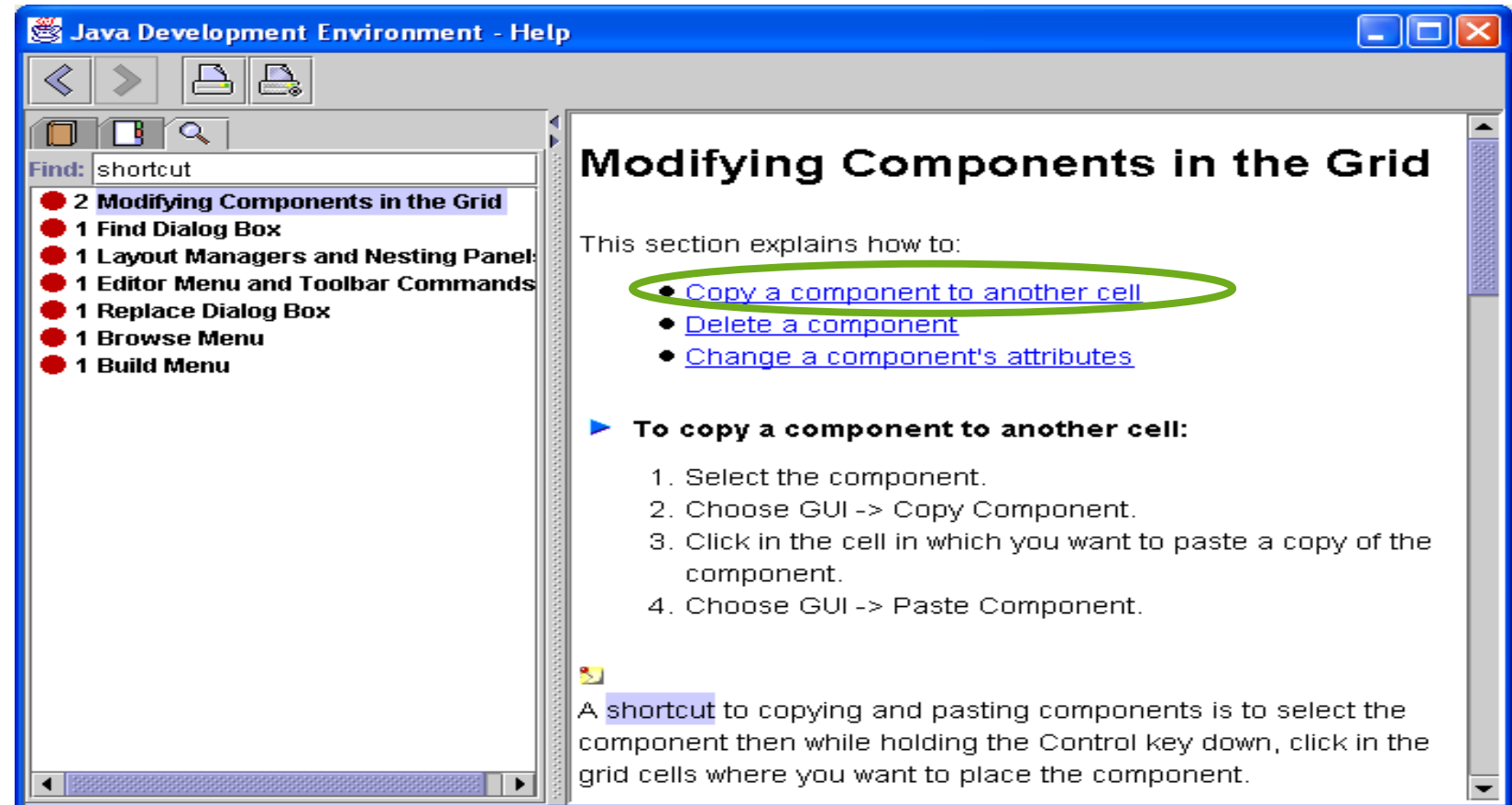
Recorrido Secuencial

- ▶ Permite una navegación secuencial por la ayuda
- ▶ Proporciona una visión de las funcionalidades del sistema
- ▶ Es especialmente útil para usuarios noveles

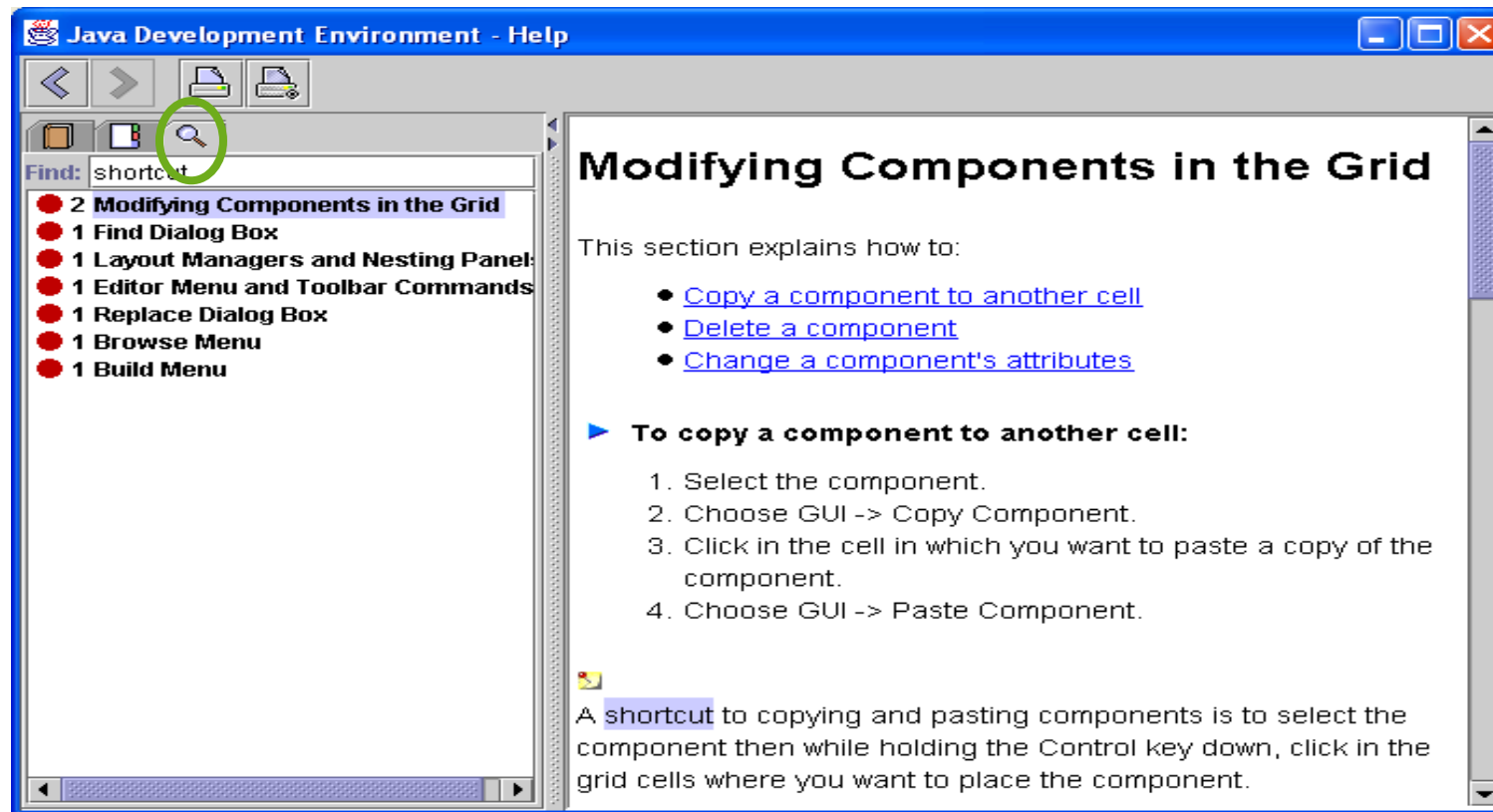


Referencias Cruzadas

- ▶ Permiten el hipertexto
- ▶ Son fragmentos del texto que enlazan con otras páginas



Opción de Búsqueda



Organización del Texto de la Ayuda

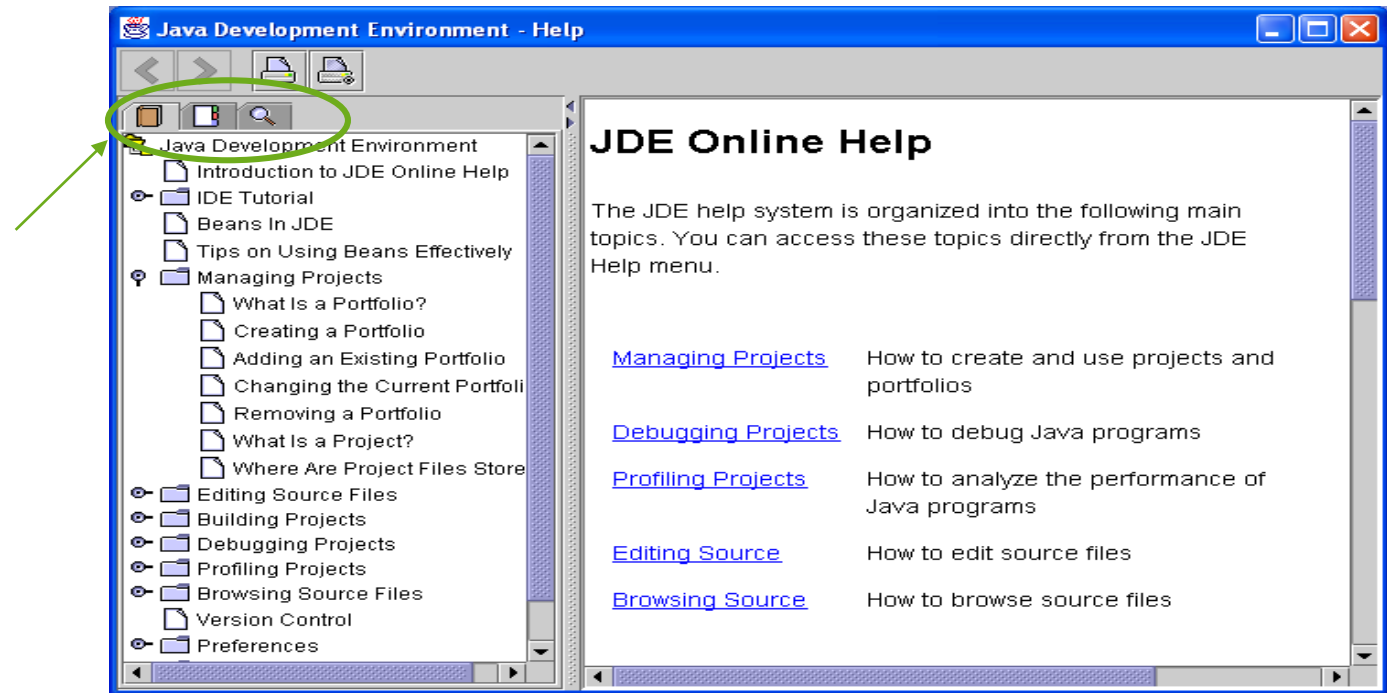
- Lenguaje claro y conciso
- Cantidad de texto
 - No excesiva, la velocidad de lectura es un 30% más lenta en la pantalla que en el papel.
 - Párrafos cortos
- Espacios en blanco : Importantes para hacer el texto más legible
- Sobreindicar el texto: Uso de tipos de letra, tamaños, color, etc. para destacar conceptos
- Gráficos: Conviene utilizar gráficos en la ayuda ya logran transmitir la información de una forma efectiva.
- Diseño consistente: Títulos con mismos colores, mismo tipo de letra...

JavaHelp

- Es una extensión de Java que facilita el desarrollo de ayuda on-line
- Es independiente de la plataforma ya que está implementado en Java.
- Las ventanas de ayuda de JavaHelp se configuran por medio de varios ficheros en formato XML. Los textos de ayuda que se quieran mostrar se escribirán en ficheros con formato HTML.

Características Básicas

- Visor de Ayuda
 - Panel de contenidos
 - Panel de navegación
- Diferentes vistas de la información
 - Tabla de contenidos
 - Índice
 - Búsqueda



Herramientas

- `hsviewer`
 - 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.

Creación del sistema de ayuda: tareas

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

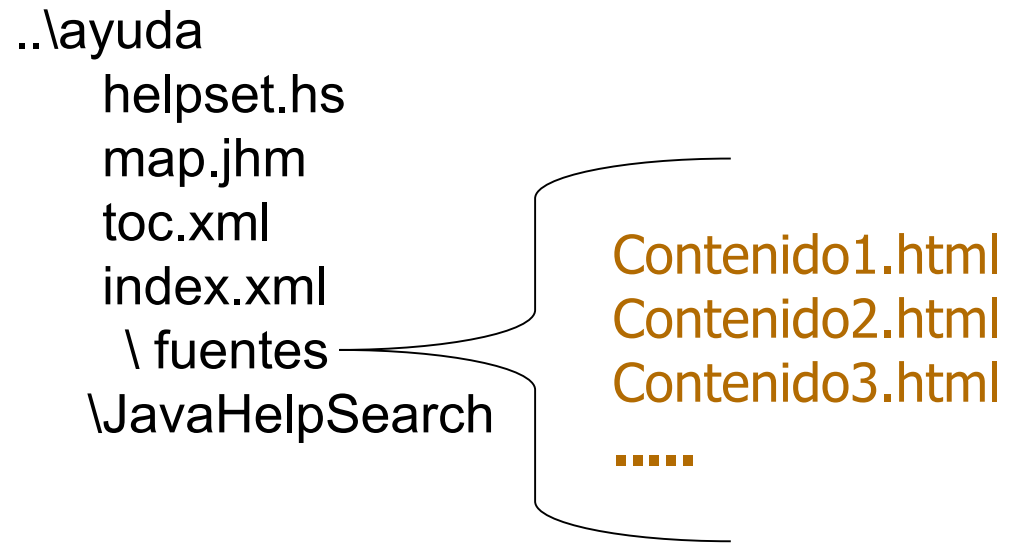
Pasos Detallados

1. Crear **archivos 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**



Imprescindibles

Organización



1. Archivos HTML

- Se pueden crear directamente en un editor de texto o bien emplear cualquier herramienta que genere HTML.
- Conviene agrupar temas relacionados y disponerlos jerárquicamente.
 - Si no, al menos, es altamente recomendable crear un subdirectorio donde se incluyan todos los html.

2. Fichero Map

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

```
<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>
```

Fichero html asociado



Identificador o "alias"

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)

```
<helpset>
```

```
  <title> Ejemplo de Ayuda </title>
```



Título

```
<maps>
```

```
  <homeID>introduccion</homeID>
```



Página a visualizar 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 (tooltip) 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
```

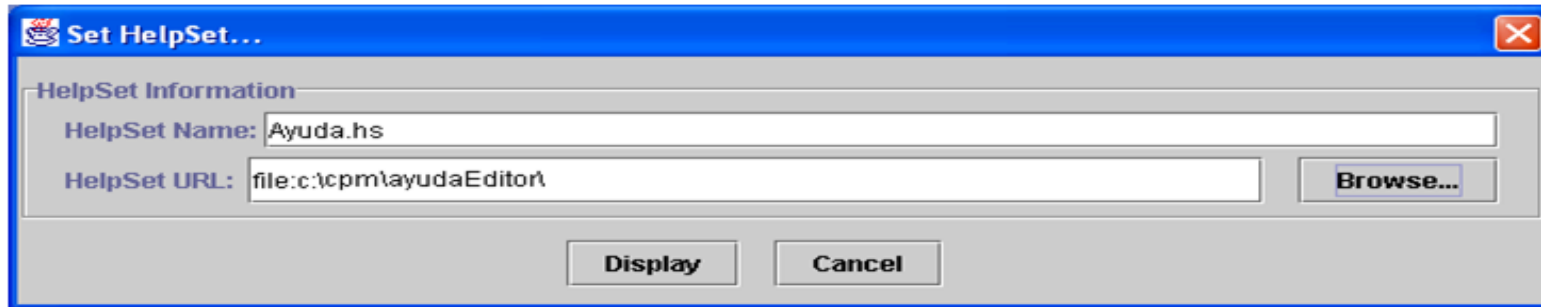
```
</view>
```

```
....
```

```
</helpset>
```

Visualización del HelpSet

- En línea de comandos
 - `hsvviewer -helpset c:\cpm\ayudaEditor\ayuda.hs`
- En Windows



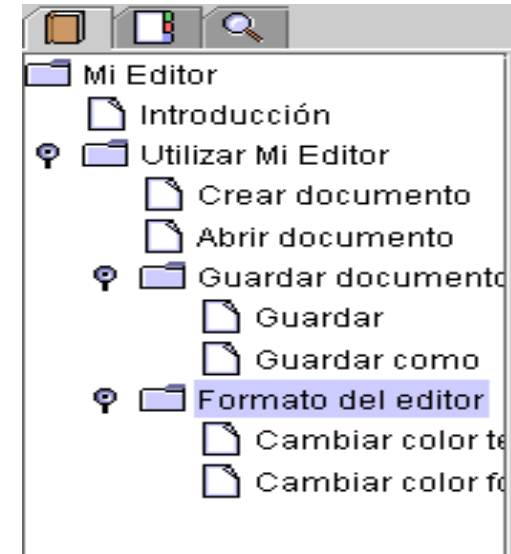
4. Fichero TOC

- 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>
  </tocitem>
</toc>
```

Especifica el texto para la entrada en la tabla de contenidos



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

5. Fichero Índice

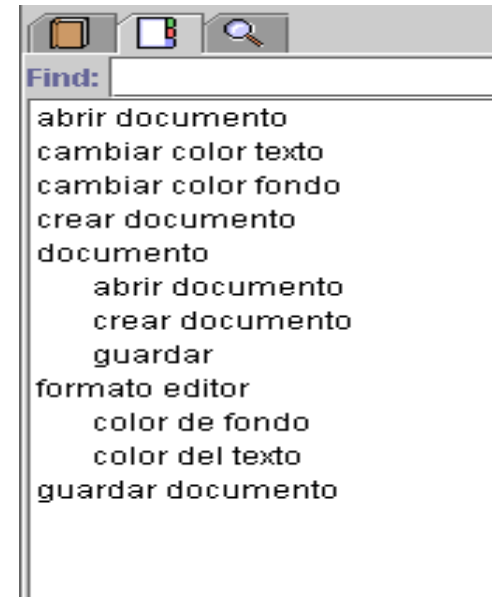
- Este fichero describe la **distribución y contenido del índice**
- Sintaxis basada en XML

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



6. Búsqueda

- Se emplea *jhindexer* para generar la base de datos de búsqueda
- La base de datos está formada por seis ficheros localizados en una nueva carpeta *JavaHelpSearch*

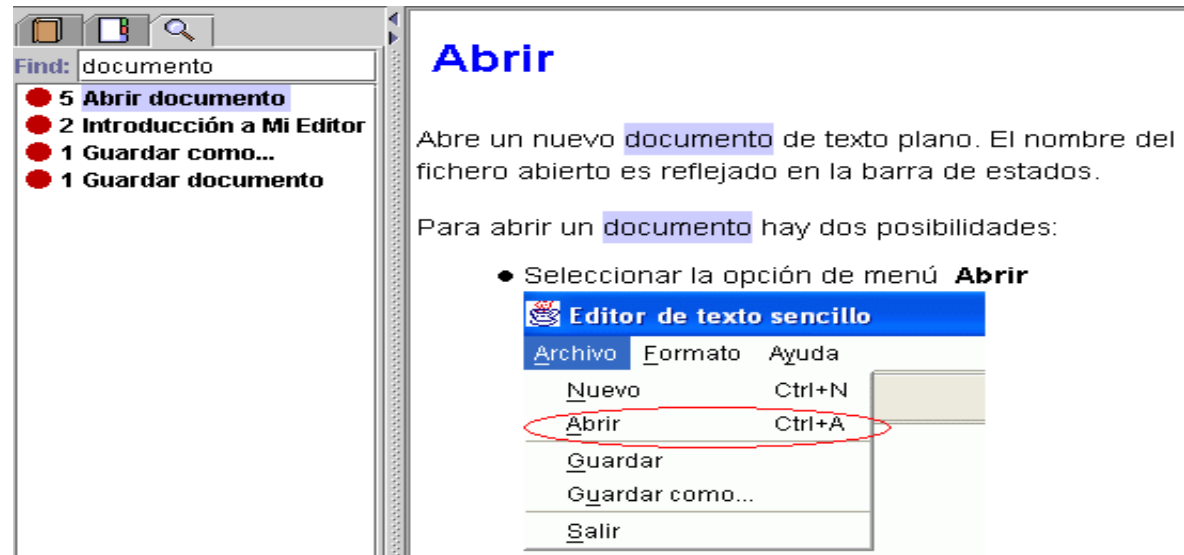
Búsqueda (II)

- Para crear la base de datos de búsqueda
 - *jhindexer fuentes*
 - Siendo fuentes el directorio donde están todos los archivos html
 - *jhindexer fuentes1 fuentes2*
 - Si tenemos más de un directorio con los html. En este caso serían 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:



Incorporación de la ayuda en la aplicación

- Clases básicas:

- URL
 - HelpSet
 - HelpBroker
- } **Pertenecen a la librería jhall.jar**

- 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 implicadas de jhall.jar

- Clase HelpSet
 - 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
- Clase HelpBroker
 - 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 F1 para mostrar la ayuda
 - **enableHelpOnButton** // Hace que se despliegue la ayuda cuando se presiona un determinado botón o elemento de menú.
 - **enableHelp** // Indica el html a mostrar cuando se pulse la tecla de ayuda sobre un determinado componente

Incorporación de la ayuda en la aplicación (I)

1. Incorporar la librería `jhall.jar` a la aplicación
2. Importar los paquetes que son necesarios en la aplicación

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

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;  
}
```

Incorporación de la ayuda en la aplicación (IV)

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

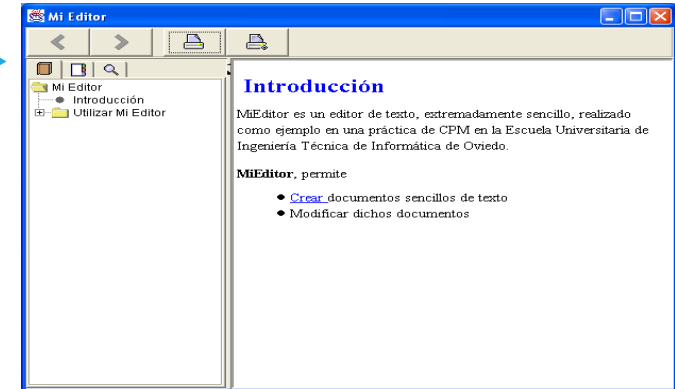
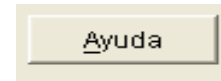
```
hb = hs.createHelpBroker ();
```

- 5. Hacer que la ventana 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
```

Incorporación de la ayuda en la aplicación (V)

6. Asociar la ayuda a un botón o elemento de menú
`hb.enableHelpOnButton(btAyuda,"introduccion",hs)`

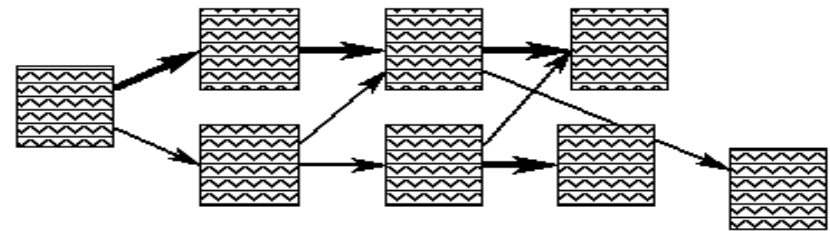


7. Asociar ayuda sensible al contexto a un componente. Para ello se asociará a cada componente una página que será la que se muestre cuando se pulse la tecla de ayuda (F1) sobre él:

```
hb.enableHelp(btGuardar,"guardar",hs);  
hb.enableHelp(btAbrir,"abrir",hs);
```


Hipertexto

- Presentar texto en una forma lineal, una página después de otra, no es apropiado para el recorrido de una ayuda en línea ya que puede desear saltar de una parte de la ayuda a otra.
- El hipertexto intenta evitar estas limitaciones del texto estructurándolo en una malla. Esto permite acceder desde una página a diferentes páginas.
- Hipertexto se puede definir como una forma no lineal de presentar información; de esa manera el usuario puede navegar del modo que prefiera a través de dicha información.
- Un sistema de hipertexto consta de un número de páginas y un conjunto de enlaces que son usados para conectar páginas. Los enlaces pueden unir cualquier página a cualquier otra, y puede haber más de un enlace por página.



Hipermedia

- Los sistemas hipertexto incorporan habitualmente elementos como imágenes, vídeos y otros medios, además del texto. Se conocen entonces como sistemas *hipermedia*.
- Hipermedia es el resultado de la combinación de hipertexto y multimedia.
- Los sistemas hipermedia son apropiados para la creación de un sistema de ayuda. De hecho, muchos sistemas de ayuda comerciales emplean un sistema de ayuda de tipo hipermedia.

HTML

- HTML (*Hypertext Mark-Up Language*) es un lenguaje que permite el desarrollo de documentos de hipertexto. Se trata por tanto de un lenguaje para describir documentos de texto que contienen referencias a otros documentos.
- Fue propuesto por Tim Berners-Lee en 1991.
- El organismo encargado de estandarizar este lenguaje es el W3C. La última versión es HTML5.



Etiquetas HTML

- HTML está compuesto por etiquetas que definen **la estructura** del documento que verá el usuario.
- Para ello encierra cada porción de texto entre un *par de etiquetas*, una de *apertura* y otra de *cierre*.
- as etiquetas se delimitan por los signos < y >:

<etiqueta > .. </etiqueta>

<etiqueta atributo1=valor1 atributo2=valor2> .. </etiqueta>

```
<h1> Esto es un encabezado </h1>  
Esto no.
```

Estructura básica de un documento HTML

```
<!DOCTYPE HTML>
```

```
<html>
```

```
    <!-- Esto es un comentario -->
```

```
    <head>
```

```
        <title>Título de la página</title>
```

```
    </head>
```

```
    <body>
```

```
        Contenido de la página
```

```
    </body>
```

```
</html>
```

Jerarquía del Documento

Títulos

- HTML dispone de 6 niveles jerarquizados, de mayor a menor nivel de relevancia:
 - `<h1> Título nivel 1 </h1>`
 - `<h2> Título nivel 2 </h2>`
 - ...
 - `<h6> Título nivel 6 </h6>`

Párrafos

- `<p>.... </p>`

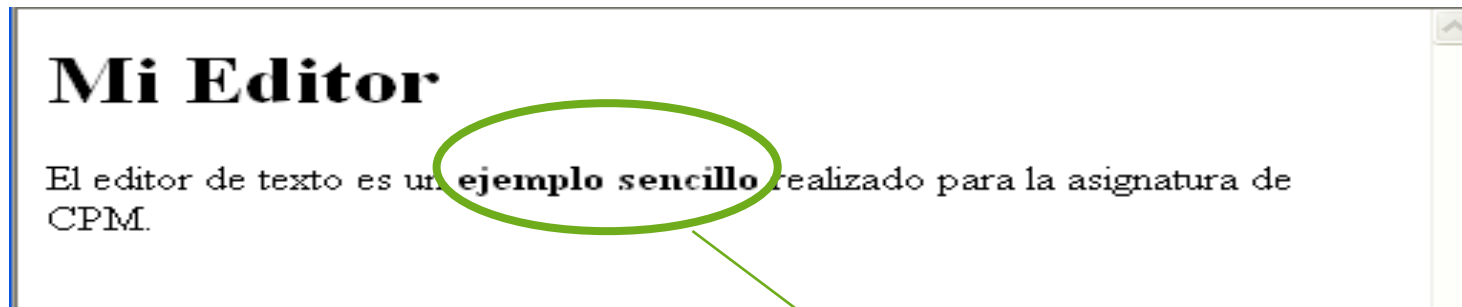
Saltos de línea

- Etiqueta `
` (no es necesaria etiqueta de cierre)

Estilos

	<i>Negrita</i>	<i>Cursiva</i>	<i>Subrayado</i>
<i>Etiqueta</i>	<code>... </code>	<code>... </code>	<code><u>...</u></code>

Ejemplo



`ejemplo sencillo`

Listas Ordenadas y no Ordenadas

	<i>Ordenada</i>	<i>No Ordenada</i>
<i>Etiqueta</i>	<pre> Uno Dos Tres </pre>	<pre> Uno Dos Tres </pre>

Ejemplo

Mi Editor

El editor de texto es un **ejemplo sencillo** realizado para la asignatura de CPM.

- Nuevo
- Abrir
- Guardar

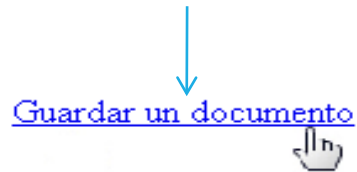
```
<ul>
  <li> Nuevo
  <li> Abrir
  <li> Guardar
</ul>
```


Uso de Enlaces

Para establecer enlaces se utiliza la etiqueta `<a>...` con el atributo HREF. Éste toma como valor el identificador del recurso al que se desea apuntar.

Ejemplo:

```
<a href="fuentes/guardar.html"> Guardar un documento</a>
```



Inclusión de Imágenes

La etiqueta `` permite la inclusión de gráficos en un documento HTML.

Atributos:

- `src`: gráfico a visualizar.
- Anchura y altura (opcional): en pixels o en porcentaje (ej: `width="300" height="150"`)

```
<p>
```

```
  
```

```
  <!-- No es necesario cerrar la etiqueta -->
```

```
</p>
```

CSS (Cascading Style Sheets)

- Las hojas de estilo en cascada (CSS) definen la **presentación** de un archivo html.
- El estilo a aplicar en el documento html puede indicarse:
 - En una etiqueta asociada al elemento:

```
<p style="color: red; margin-left: 20px">Texto que va a cambiar de estilo</p>
```
 - En la cabecera del propio documento:

```
<head>  
  <style type="text/css">  
    p{color:red; margin-left:20 px}  
  </style>  
</head>
```
 - En un fichero independiente, con extensión css. En el documento html se incluye la referencia al fichero css de la siguiente manera:

```
<head>  
  
  <link rel="stylesheet" type="text/css" href="fichero.css" />  
  
</head>
```

Reglas CSS

- Las hojas de estilo están compuestas de reglas.
- Cada regla tiene tres partes:
 - ▶ El selector, que indica al navegador la parte del documento que se verá afectada por la regla
 - ▶ La propiedad, que especifica qué aspecto del diseño va a cambiarse;
 - ▶ El valor que se establece a dicha propiedad

`selector {property: value;}`

↑
A qué etiqueta(s)
HTML se aplica
la propiedad (por
ejemplo, "body")

↑
La propiedad, por ejemplo,
podría ser el color de fondo
("background-color")

← el valor de la propiedad
"background-color"
podría ser, por ejemplo,
rojo ("#FF0000", valor
en hexadecimal)

`body { background-color: #FF0000 }`

CSS – Propiedades principales

▶ Texto (Text):

- ▶ color: [color]
- ▶ text-align: [left / right / center / justify]

Los colores se pueden introducir como valores hexadecimales (#ff0000) o se pueden usar los nombres de los colores (red) o bien como valores rgb: (rgb(255,0,0))

▶ Tipo de letra (Font):

- ▶ font-family: [fuente]
- ▶ font-size [xx-small / ... / small / medium / large / % / size (pt/px)]
- ▶ font-weight: [extra-light / ... / medium / ... / bold / ...]

▶ Listas:

- ▶ list-style-type: [disc / circle / square / ...]

```
body {  
    font-family: Georgia, "Times New Roman", Times, serif;  
    color: purple;  
    background-color: #d8da3d }  
h1 {  
    font-family: Helvetica, Geneva, Arial, sans-serif;  
    font-size: 12pt }
```

Si el nombre de la fuente tiene espacios en blanco, se debe escribir entre doble s comillas

XML

- XML (Extensible Markup Language) es otro lenguaje de marcas pero extensible, es decir, en el que el usuario establece qué marcas o etiquetas emplear.
- XML no es realmente un lenguaje en particular. Se trata de un metalenguaje ya que permite definir lenguajes para diferentes necesidades

```
<?xml version="1.0" encoding="UTF-8"?>
<films>
  <pelicula>
    <titulo>Braveheart</titulo>
    <director>Mel Gibson</director>
    <genero>épico</genero>
  </pelicula>
  <pelicula>
    <titulo>Blade Runner</titulo>
    <director>Ridley Scott</director>
    <genero>Ciencia Ficción</genero>
  </pelicula>
</films>
```

Soporte al usuario: Instalador

- Como ya se ha mencionado, la usabilidad no sólo se ocupa de aspectos relacionados con el desarrollo del sistema software propiamente dicho si no que interviene en el sistema de ayuda y del procedimiento de instalación.
- La facilidad de instalación del producto software es un aspecto valorado en las pruebas de usabilidad.
- Utilizaremos una herramienta para facilitar la instalación del producto software: InstallAnywhere.

Instalador: pasos

1. Establecer el directorio donde almacenar el instalador
2. Indicar el nombre del producto, del instalador, carpeta que contendrá la aplicación y el nombre de la aplicación
- 3. Añadir al instalador los ficheros que vamos a distribuir**
- 4. Indicar la clase que tiene el método principal**
- 5. Establecer el classpath**
- 6. Indicar la plataforma destino y la incorporación o no de la máquina virtual de java**

Variables de entorno: Path, ClassPath

- **Path:** contiene la ruta de acceso de los programas que se utilizan con más frecuencia, de modo que para ejecutar un programa bastará con escribir su nombre sin tener que escribir toda la ruta de directorios hasta llegar a él.
- **ClassPath:** indica al JDK dónde debe buscar los archivos a compilar o ejecutar, sin tener que escribir en cada ejecución la ruta completa.
- Así se ejecutaría java sin haber cambiado las variables, en MS-Dos y con el JDK instalado en el directorio C:\jdk1.8 y la clase a ejecutar de nombre 'Miclase.class' en C:\proyectosJava:

```
C:>C:\jdk1.8\bin\java C:\proyectosJava\Miclase.class
```

- Así se escribiría si se cambian adecuadamente las variables:

```
C:>java Miclase.class
```

Es decir, no sería necesario incluir la ruta de acceso del programa ni del archivo a ejecutar

ClassPath

- La variable de entorno classpath indica al JDK dónde debe buscar los archivos a compilar o ejecutar, sin tener que escribir en cada ejecución la ruta completa.
- Varios casos:
 1. La clase no pertenece a ningún paquete
 2. La clase pertenece a un paquete
 3. La clase está en un .jar

ClassPath: clase no pertenece a paquete

- Caso 1: La clase no pertenece a ningún paquete
 - Se incorpora al classpath la ruta completa de llegada al directorio (incluido) que contiene la clase
 - Ejemplo: queremos ejecutar la clase

c:\archivos de programa\utilidades\graficos\aplicacion.class

Introduciremos en el classpath

c:\archivos de programa\utilidades\graficos

ClassPath: clase pertenece a paquete

- Caso 2: La clase pertenece a un paquete => existe un directorio que contiene las clases del mismo nombre que el paquete
 - Se incorpora al classpath la ruta completa de llegada **al directorio que contiene el paquete**
 - Ejemplo: queremos ejecutar la clase *aplicacion.class* del paquete *graficos* situado en *c:\archivos de programa\utilidades*

Esto implica que hay un directorio:

c:\archivos de programa\utilidades\graficos que contiene los .class

Introduciremos en el classpath:

c:\archivos de programa\utilidades

ClassPath: clase en jar

- Caso 3: La clase está en un .jar
 - Se incorpora al classpath la ruta completa de llegada al .jar (incluido) que contiene la clase
 - Ejemplo: queremos ejecutar la clase aplicacion.class contenida en graficos.jar

c:\archivos de programa\utilidades\graficos.jar

Introduciremos en el classpath:

c:\archivos de programa\utilidades\graficos.jar

Conocer el valor de ClassPath

- Para saber qué valores contiene el CLASSPATH:
 - El línea de comandos: bastará con teclear: `C:>set`
 - En Windows:
Inicio – Panel de Control – Sistema – Configuración avanzada del sistema - Variables de entorno
- Si no aparece la variable CLASSPATH quiere decir que no contiene ningún valor.

ClassPath: Asignación

- Para asignarle un valor teclearemos lo siguiente en línea de comandos:

```
C:>set CLASSPATH = C:\ClasesDeJava
```

Siendo ClasesDeJava el directorio donde tenemos nuestras clases de Java.

- Para introducir varios directorios los separaremos por punto y coma:

```
C:>set CLASSPATH = C:\ClasesDeJava;C:\OtrasClasesDeJava
```

- Finalmente si la variable CLASSPATH ya contiene valores y queremos añadir más tendremos que hacerlo del siguiente modo:

```
C:>set CLASSPATH = C:\ClasesDeJava;%CLASSPATH%
```

- En **Windows** se modifica la variable de entorno directamente mediante el diálogo que nos muestra el valor y permite cambiarlo.