

***DESARROLLO DE INTERFACES***  
***2º CURSO DAM***  
***I.E.S. HERMENEGILDO LANZ***  
***JOSÉ MARÍA MOLINA***



**TEMA 4-1 – PRUEBAS Y  
DOCUMENTACIÓN**

# TEMA 4 – DOCUMENTACIÓN, DISTRIBUCIÓN Y PRUEBAS

El último paso en el desarrollo de una APP-GUI es (en este orden lógico): las **pruebas**, la **documentación** (manual de usuario), y la distribución MULTIPLATAFORMA (creación del instalador).

- ✓ 1 PRUEBAS
- ✓ 2 DOCUMENTACIÓN
  - FICHEROS DE AYUDA (Formatos y tipos)
  - HERRAMIENTAS DE GENERACIÓN DE AYUDAS (JavaHelp y Webview)
  - TIPOS DE MANUALES
  - ESTRUCTURA DE UN MANUAL



# 1 PRUEBAS

## 1. OBJETIVO, IMPORTANCIA Y LIMITACIONES DEL PROCESO DE PRUEBA. ESTRATÉGIAS.

- No es necesario indicar la importancia de realizar pruebas al software. Hay distintas estrategias y tipos de prueba que hay que tener en cuenta.
- No todos los tipos de pruebas tienen cabida según la APP que estamos tratando( ej: una prueba de carga o estrés no tiene sentido en una aplicación de gestión)
- Las pruebas adecuadas para nuestra APP de gestión son las **FUNCIONALES** , de **SISTEMA**, **REGRESIÓN**, **ALFA** y **UAT**

# 1 PRUEBAS

## 2. PRUEBAS DE INTEGRACIÓN, ASCENDENTES Y DESCENDENTES.

- Son pruebas unitarias **FUNCIONALES** que permiten integrar módulos en una aplicación más grande. Existen principalmente dos tipos de pruebas de integración:
  - **No incrementales:** consiste en combinar todos los módulos y probar todo el programa en su conjunto. El resultado puede ser un poco caótico con muchos fallos difíciles de identificar el módulo/s de origen.
  - **Incrementales:** el programa se prueba poco a poco en las que los fallos son más fáciles de detectar. Existen dos tipos de integración incremental, ascendente y descendente. El ascendente prueba desde lo más pequeño a lo más general, el descendente lo hace de forma inversa.

Se recomiendan pruebas de integración (**FUNCIONALES**) incrementales ascendentes. 😊



# 1 PRUEBAS

## 3. PRUEBAS DE SISTEMA, CONFIGURACIÓN, RECUPERACIÓN, REGRESIÓN, VOLUMEN Y ESTRÉS Y SEGURIDAD

- Pruebas de **SISTEMA** es un nivel de prueba que valida el producto de software completo y totalmente integrado. Diferencia con funcionales?
- Pruebas de **CONFIGURACIÓN** pretende comprobar errores tras cambios en la configuración de la aplicación (necesitará de un panel de configuración para ser probado).
- Las pruebas de **RECUPERACIÓN** son pruebas no funcionales que determinan la capacidad del software para recuperarse de fallos tales como fallas de software / hardware o cualquier fallo de la red.
- Las pruebas de **REGRESIÓN** son cualquier tipo de pruebas de software con el objeto de descubrir errores (bugs), carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, causados por la realización de un cambio en el programa una vez acabado. El propósito de éstas es asegurar que los casos de prueba que ya habían sido probados y fueron exitosos permanezcan así.

# 1 PRUEBAS

## 3. PRUEBAS DE SISTEMA, CONFIGURACIÓN, RECUPERACIÓN, REGRESIÓN, VOLUMEN Y ESTRÉS Y SEGURIDAD

- Una prueba de **CARGA O VOLUMEN** es una prueba planificada para realizar un número específico de solicitudes a un sistema para probar la funcionalidad del sistema bajo niveles de rendimiento normal.
- Una prueba de **ESFUERZO O ESTRÉS** es una prueba diseñada para aumentar el número de solicitudes simultáneas en un sistema más allá de un punto donde el rendimiento se degrada, incluso hasta el punto de un fallo completo.



- Pruebas de **SEGURIDAD**: es un tipo de prueba de software que revela vulnerabilidades, amenazas, riesgos en una aplicación de software y previene ataques maliciosos de intrusos. El propósito de las Pruebas de Seguridad es identificar todas las lagunas y debilidades potenciales en el sistema de software que podrían resultar en la pérdida de información, ingresos, reputación de los empleados o personas ajenas a la Organización.



# 1 PRUEBAS

## 4. PRUEBAS DE USUARIO. PRUEBAS DE ACEPTACIÓN. VERSIONES ALFA Y BETA.

- Pruebas de **USUARIO**: las realizan usuarios externos a la programación del software: hay de dos tipos, usuarios internos de la empresa (pruebas **ALFA**) y usuarios externos (pruebas **BETA** o **UAT** o **de ACEPTACIÓN** o **de USUARIO FINAL**).
- Las pruebas BETA, se define como la prueba del software por parte del usuario o cliente para determinar si puede ser aceptado o no. Esta es la prueba final que se realiza una vez que se completan las pruebas **FUNCIONALES**, de **SISTEMA**, **REGRESIÓN** y **ALFA**

# 1 PRUEBAS

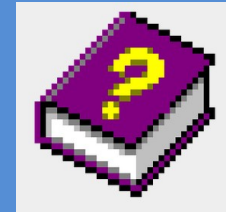
- **5. PRUEBAS MANUALES Y AUTOMÁTICAS. HERRAMIENTAS SOFTWARE PARA LA REALIZACIÓN DE PRUEBAS**
- En las **pruebas manuales** los casos de **prueba** se ejecutan manualmente (es decir, por un humano) sin el apoyo de herramientas o scripts. Pero con las **pruebas automatizadas**, los casos de **prueba** se ejecutan con la ayuda de herramientas, scripts y software (ej: JUnit)
- El problema es que al ser una APP interactiva utilizar JUnit no es tan directo ni sencillo, por lo que seguiremos con las pruebas manuales.
- Existen librerías de prueba específicas para JavaFX (TestFX)



# 1 PRUEBAS

- Para resumir, en la práctica nosotros hacemos y por este ORDEN:
  - A. **FUNCIONALES**: cada vez que se termina un módulo (ej: Update del CRUD de una tabla)
  - B. **SISTEMA**: cada vez que se acaba una versión (ej: CRUD de una tabla)
  - C. **REGRESIÓN**: se repiten todas las pruebas anteriores (funcionales sobre los módulos aceptados y todas las de sistema) cada vez que se han hecho cambios/mejoras/arreglos sobre una versión entregada.
  - D. **ALFA**: tests de guerrilla
  - E. **UAT**: (José María 😊)

## ***2 DOCUMENTACIÓN***



### **1. FICHEROS DE AYUDA.**

- **Formatos estándar:**
  - Antiguamente en Windows se utilizaba un formato estándar con extensión HLP (F1).
  - Páginas web: por ejemplo el formato CHM: consisten en un conjunto de páginas web comprimidas y compiladas en un único archivo. Hoy en día también se pueden abrir desde Linux (instalando paquetes, ej: xchm).
    - Software específico, ej: CHM Editor, HelpNDoc o HTML Help Workshop.
  - PDF/HTML: opción muy lógica cuando se trabaja con APPs multiplataforma.
- **Ayuda general y ayuda sensible al contexto:** la ayuda general consiste en un único archivo donde está todo, la sensible al contexto nos permite dar información tipo tooltip y/o “saltar” de la aplicación a la explicación dentro de la ayuda.



## ***2 DOCUMENTACIÓN***

### **2. HERRAMIENTAS DE GENERACIÓN DE AYUDAS.**

- **JAVAHelp:** es una clase oficial que permite crear ayuda contextual de forma sencilla y empaquetada en una estructura Web
- **Problema:** Está desfasado y sin soporte, y aunque se podría utilizar en JavaFX no se recomienda!
- Se ha de añadir la librería al proyecto.
- **Estructura:**
  - Páginas de ayuda en html (se puede crear en Word/Writer y exportar). Las páginas pueden ser *generales*, por *secciones* o *componente*

# 2 DOCUMENTACIÓN

## 2. HERRAMIENTAS DE GENERACIÓN DE AYUDAS.

The screenshot shows the 'Ayuda - JAVAPLATOSGUI' application window. The main content area is titled 'VENTANA PRINCIPAL' and contains the text: 'La aplicación Gestión de Platos permite gestionar una tabla de una base de datos SQLITE.' Below this, there is a sub-window titled 'Gestión de Platos - DI - 2ºDAM' with a menu bar containing 'Fichero', 'Informes', and 'Ayuda'. The sub-window has a toolbar with icons for 'Nuevo', 'Listar Platos', 'Listar Platos Tipo', and 'Recetas'. The main area of the sub-window displays a table of dishes with the following data:

ID	Plato	Tipo	Calorias	Pre...	
1	Macarrones Carbonara	primero	1100.0	9.5	Editar
2	Migas de pan	primero	950.0	7.0	Editar
3	Huevos Cabreaos	primero	1200.0	8.91	Editar
4	Papas Bravas	entrante	750.0	6.0	Editar
5	Tarta 3Chocolates	postre	1200.0	7.0	Editar
6	Ensalada Mixta	entrante	450.0	5.5	Editar
8	Sopa de Picadillo	entrante	450.0	4.5	Editar
14	Chuletón de Ávila	segundo	2000.0	22.0	Editar



# 2 DOCUMENTACIÓN

html
help_set.hs
indice.xml
map_file.jhm
toc.xml

## 2. HERRAMIENTAS DE GENERACIÓN DE AYUDAS.

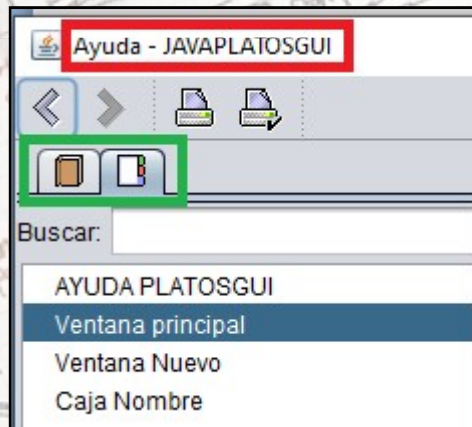
### ■ Estructura:

- **help\_set.hs**: define la estructura principal de la ventana de ayuda que a su vez llama a 3 ficheros
  - el mapa (esqueleto) de la ayuda (map\_file.jhm)
  - la tabla de contenidos (toc.xml)
  - y el índice (indice.xml)
- **map\_file.jhm**: pone nombre (target) y ruta a las páginas web.
- **toc.xml**: asigna los nombres que aparecerán en el índice de secciones/contenidos (parte izquierda).
- **indice.xml**: igual que indice.xml pero para la BÚSQUEDA por sección/contenido.
- Se puede también meter una vista SearchView pero hay que instalar un programa que genere la base de datos de ayuda y es más complejo.

# 2 DOCUMENTACIÓN

- html
- help\_set.hs
- indice.xml
- map\_file.jhm
- toc.xml

help\_set.hs:



```
1  <?xml version="1.0" encoding='ISO-8859-1' ?>
2  <!DOCTYPE helpset PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp
3  "http://java.sun.com/products/javahelp/helpset_1_0.dtd">
4  <helpset version="1.0">
5      <title>Ayuda - JAVAPLATOSGUI</title>
6      <maps>
7          <!-- Pagina por defecto al mostrar la ayuda -->
8          <homeID>aplicacion</homeID>
9          <!-- Que mapa deseamos -->
10         <mapref location="map_file.jhm"/>
11     </maps>
12
13     <!-- Las Vistas que deseamos mostrar en la ayuda -->
14     <!-- La tabla de contenidos -->
15     <view>
16         <name>Tabla Contenidos</name>
17         <label>Tabla de contenidos - PLATOSGUI</label>
18         <type>javax.help.TOCView</type>
19         <data>toc.xml</data>
20     </view>
21
22     <!-- El indice -->
23     <view>
24         <name>Indice</name>
25         <label>El indice</label>
26         <type>javax.help.IndexView</type>
27         <data>indice.xml</data>
28     </view>
29 </helpset>
```



# 2 DOCUMENTACIÓN

- html
- help\_set.hs
- indice.xml
- map\_file.jhm
- toc.xml

map\_file.hs

toc.xml

indice.xml

```
help_set.hs x map_file.jhm x indice.xml x toc.xml x
1  <?xml version='1.0' encoding='ISO-8859-1' ?>
2  <!DOCTYPE map
3    PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp Map Version 2.0//EN"
4    "http://java.sun.com/products/javahelp/map_2_0.dtd">
5  <map version="1.0">
6    <mapID target="aplicacion" url="html/main.html" />
7    <mapID target="ventana_principal" url="html/principal.html" />
8    <mapID target="nuevo" url="html/nuevo.html" />
9    <mapID target="caja_nombre" url="html/caja_nombre.html" />
10 </map>
```

```
help_set.hs x map_file.jhm x indice.xml x toc.xml x
1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <toc version="1.0">
3    <tocitem text="Ayuda GENERAL" target="aplicacion">
4      <tocitem text="Ventana Principal" target="ventana_principal"/>
5      <tocitem text="Ventana Nuevo" target="nuevo"/>
6      <tocitem text="Caja Nombre" target="caja_nombre"/>
7    </tocitem>
8  </toc>
```

```
help_set.hs x map_file.jhm x indice.xml x toc.xml x
1  <?xml version='1.0' encoding='ISO-8859-1' ?>
2  <!DOCTYPE index
3    PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp Index Version 1.0//EN"
4    "http://java.sun.com/products/javahelp/index_1_0.dtd">
5  <index version="1.0">
6    <indexitem text="AYUDA PLATOSGUI" target="aplicacion"/>
7    <indexitem text="Ventana principal" target="ventana_principal"/>
8    <indexitem text="Ventana Nuevo" target="nuevo"/>
9    <indexitem text="Caja Nombre" target="caja_nombre"/>
10 </index>
```

## 2 DOCUMENTACIÓN

### INSERCIÓN JAVADOC

**HelpSet** es el sistema de ayuda. **HelpBroker** es la ventana de ayuda. Ejemplo JavaSwing

```
private void generaAyudaJavaHelp() throws MalformedURLException{
    // Carga el fichero de ayuda JavaHelp
    File fichero = new File(System.getProperty( key: "user.dir")+"/ayuda/help_set.hs");
    URL hsURL;

    hsURL = fichero.toURI().toURL();
    try {
        // Crea el HelpSet
        helpSet = new HelpSet( cl:getClass().getClassLoader(), url:hsURL);
    } catch (HelpSetException ex) {
        Logger.getLogger( name:GUI.class.getName()).log( level:Level.SEVERE, msg:null, thrown:ex);
    }

    hb = helpSet.createHelpBroker();
    hb.setLocation( point:this.getLocation()); //Centra en el padre
    hb.setSize(new Dimension( width: 800, height: 600)); //Se da ancho y largo

    // Ayuda al pulsar F1 sobre las ventanas/elementos registradas
    hb.enableHelpKey( cmpnt:this.getContentPane(), string:"ventana_principal", hs:helpSet);
    hb.enableHelpKey( cmpnt:this.jPanell, string:"nuevo", hs:helpSet);
    hb.enableHelpKey( cmpnt:this.caja_nombre, string:"caja_nombre", hs:helpSet);
}
```



## ***2 DOCUMENTACIÓN***

### **2. HERRAMIENTAS DE GENERACIÓN DE AYUDAS.**

- **Página Web embebida (WebView):** es la solución más moderna.
- Lo más sencillo es utilizar un editor específico de HTML que genere un único archivo con toda la ayuda
- **Usamos esta opción!** (código siguiente página)

# ***2 DOCUMENTACIÓN***

## **2. HERRAMIENTAS DE GENERACIÓN DE AYUDAS. (ej: crud\_javafx\_cinefx)**

```
this.panelActores.getScene().setOnKeyPressed((KeyEvent event) -> {  
    if (event.getCode() == KeyCode.F1) {  
        WebView webView = new WebView();  
  
        WebEngine webEngine = webView.getEngine();  
        URL url = getClass().getResource(name: "/ayuda/ayuda.html");  
        webEngine.load(string: url.toExternalForm());  
  
        StackPane root = new StackPane();  
        root.getChildren().add(e: webView);  
  
        Scene scene = new Scene(parent: root, d: 300, dl: 250);  
        Stage stage = new Stage();  
  
        stage.initModality(modlt: Modality.APPLICATION_MODAL);  
  
        // Asigna la escena a la ventana y muestra la ventana  
        stage.setScene(scene);  
        stage.setTitle(string: "Ayuda");  
        stage.getIcons().add(e: icono);  
        stage.show();  
    }  
});
```



## ***2 DOCUMENTACIÓN***

### **3. TIPOS DE MANUALES:**

- **Manuales de usuario / instalación/configuración/administración:** incluye toda la información técnica para el manejo, instalación, configuración y/o administración de una aplicación
- **Tutoriales:** enseña paso a paso algo o de forma interactiva.
- **Guía de referencia:** son manuales cortos, claros, precisos y concisos.
- **Guías rápidas:** guías de referencia más resumidas aún.

## ***2 DOCUMENTACIÓN***

**4 ESTRUCTURA DE UN MANUAL.** Como mínimo un manual técnico debería incluir:

- Una sección de **INTRODUCCIÓN**: es una descripción general a la aplicación.
- Una sección de **OBJETIVOS**: un resumen de los objetivos generales de la APP.
- Una sección de **FUNCIONES DEL SISTEMA**: explicando el **FUNCIONAMIENTO EN DETALLE** de todas aquellas partes de la APP.
- Una sección de **SOLUCIÓN DE PROBLEMAS**: explicando aquellos problemas o errores que se pueden dar.