TEMA 2 – COMPONENTES JAVAFX – APP CON ACCESO A BASE DE DATOS

Hemos visto el diseño, usabilidad, componentes y eventos principales para la generación de una APP General de Escritorio, por tanto, la idea es recrear exactamente el diseño previo (Mockup) añadiendo elementos de forma progresiva:

1. Generar todo el código necesario para que la APP funcione correctamente. Para aprobar, tendremos que cumplir de forma **obligatoria** los siguientes puntos:
    a. Se tendrá muy en cuenta el **trabajo constante en clase**, la **dificultad** de la APP en **cuanto a base de datos** (tipos de campos y número) y la dificultad en sí del **diseño de ventanas** de la APP. Además, el código deberá ser **eficiente**, por ejemplo, no configurando constantemente el conector de la base de datos: se conecta 1 vez y se pasa el objeto de conexión por las clases (Controladores) o se pasan los datos leídos entre ventanas mediante objetos.
    b. Asegurarse que tiene todo lo necesario para ser **compilado** (de no poder compilarse no se corregirá dicha entrega, influyendo negativamente en la nota). Evitar rutas absolutas cuando se referencie a cualquier recurso.
    c. Que la APP sea **totalmente funcional**, es decir, que funcione la gestión de todas las ventanas y errores de forma correcta tal y como se ha diseñado en el Mockup previo.
    d. El diseño tiene que ser adecuado, **cumpliendo los criterios de Usabilidad y pautas de diseño** de Escritorio planteados en el Mockup previo. Dicho diseño deberá contener:
        i. Los componentes más adecuados para cada tipo de dato: ComboBox, ChoiceBox, TableView, ListView, ImageView, etc... Se tendrá en cuenta que se use el mayor tipo de componentes posibles.
        ii. Tooltips, taborder correcto y textos de accesibilidad en cada caja de inserción de datos y en cada botón de la APP.
        iii. En cada botón que lo permita (las etiquetas de tab no), poner icono además de texto. Para colocar el texto con respecto al icono se utiliza el atributo Content Display (posición) o la función .setContentDisplay(posición).
    e. **Obligatoriamente**, se tendrá que hacer TODAS las ventanas con **FXML** (no con código). Se empezó haciendo por código para una toma de contacto con los nodos y su funcionamiento pero no es lo más versátil.
    f. Se debe realizar el **CRUD completo de las 3 tablas**. Los listados deberán mostrarse al menos mediante un método (TableView o contenedores personalizados en GridPane [CARDS]), teniendo que enlazar a modelos leídos desde la base de datos. Se tendrá en cuenta si se programan varios métodos (TableView y además CARDS). También se tendrá en cuenta la variedad y número de tipos de datos de la BBDD. Aclaración sobre la inclusión de imágenes:
        i. Las imágenes se pueden guardar directamente en binario (tener en cuenta que según tipo de aplicación podría implicar que la BBDD fuera demasiado grande). Tipo de dato BLOB y derivados.

ii. Se pueden guardar reescaladas y codificadas por ejemplo en Base64 (formato versátil de intercambio de datos). Tipo de dato TEXT y derivados.

g. La **navegabilidad** ha de ser total (desde cualquier punto de la APP se puede ir a cualquier punto).

h. La APP deberá contener **eventos de distinto tipo**: ventana, teclado, ratón, etc.. Se tendrá en cuenta el número, variedad y originalidad. Ejemplos: que se pueda hacer doble click sobre una fila de un TableView para editar o también seleccionar fila y darle a la **e** para hacer lo mismo, los eventos de ratón también se pueden utilizar para mostrar menús contextuales, se pueden hacer filtros en los eventos de teclado para que actúen de distinta forma según componente que esté escuchando, etc..

2. Habrá que realizar **3 entregas parciales** con sus correspondientes defensas (ver fechas en Moodle). Una entrega suspensa o no entregada se podrá recuperar en la siguiente, pero teniendo en cuenta que la nota de la parte no entregada será un APTO si es que se supera la defensa. Hay que tomarse cada entrega como lo que es, un examen:

   a. **Entrega 1 de 3**: El Diseño completo de todas las ventanas con su navegabilidad, acceso a BBDD y listado en las 3 tablas (tomando datos de la BBDD). Enviar un fichero comprimido con el nombre **JAVAFX_<NOMBREALUMNO> o un enlace a GDRIVE/ONEDRIVE/GITHUB (Fecha en Moodle)** que contenga:

      1) El proyecto de VSCODE exportado a ZIP

      2) El fichero SQL con la BBDD exportada (desde PHPMyadmin)

      3) Una pequeña ficha (en un txt o word) donde se detallen datos como:

         a. El usuario/pwd para acceder a la base de datos y el puerto utilizado. Indicar si se ha utilizado algún fichero de properties.

   b. **Entrega 2 de 3**: Diseño + CRUD completo de las 3 tablas+ Validación de campos. Enviar un fichero comprimido con el nombre **JAVAFX_<NOMBREALUMNO> o un enlace a GDRIVE/ONEDRIVE/GITHUB (Fecha en Moodle)** que contenga además todo lo que se pide en la Entrega 1 de 3. Si se borra la carpeta build ocupará poco y se podrá enviar siempre vía moodle.

      i. **Comprobación y validación de campos**: se tendrán que validar todos los campos de entrada de datos utilizando cualquiera de los métodos vistos en clase, pero será uno que sea vistoso (cambiando diseño de componente, añadiendo adornos o variando el estilo).

   c. **Entrega 3 de 3**: Diseño + CRUD + Validación + Aspectos que se tendrán que ir añadiendo a la práctica de forma progresiva conforme se vayan explicando en clase (deberá contener también todo lo referente a las dos primeras entregas):

      i. **Uso de CSS**: se utilizará al menos 1 hoja de estilo CSS y se aplicarán todos los tipos: estilo directo, clase e identificador. Se tendrá que dar estilo a todos los componentes de la interfaz.

      ii. **Animaciones**: el número de animaciones no está limitado, la nota dependerá del número de elementos a los que se le aplique y de la variedad y vistosidad de los mismos.

      iii. **Menús Popups (Contextuales)**: se utilizarán varios menús contextuales en distintas partes de la aplicación que hagan de acceso directo a opciones o que hagan de puente para opciones ya hechas (por ejemplo: menú con opción de borrar dato cuando ya existe un botón de borrar en alguna parte de esa ventana para eliminar dicho dato)

      iv. **Investigar** y añadir todas las **funcionalidades extra** que se quiera utilizando componentes o métodos no vistos en clase, influyendo positivamente en la nota, por ejemplo:

         1. La realización de buscadores (de cualquier tipo) no necesarios para el funcionamiento base de la APP.

2. Ventanas de “Ver”
3. Importar/Exportar datos a la BBDD vía cualquier origen (API externa) o tipo de fichero: XML, JSON, TXT, CSV, etc..
4. Cualquier capa gráfica que se pueda importar para representar la interfaz, los gráficos o cualquier otra funcionalidad. Existen muchas librerías basadas en JavaFX (veremos algunos ejemplos en la parte 2-7 LIBRERÍAS FX).

v. Enviar un fichero comprimido con el nombre **JAVAFX_<NOMBREALUMNO> o un enlace a GDRIVE/ONEDRIVE/GITHUB (Fecha en Moodle)** que contenga todo lo que se pide en las entregas anteriores y además:

1. Explicación de dónde y cómo se ha aplicado cada uno de los apartados del punto c. Es decir: qué métodos se han usado para validar campos, cuántos ficheros CSS se han utilizado, dónde se han aplicado animaciones, dónde están los menús contextuales y qué opciones lanzan, etc.
2. Una explicación completa de cada una de las funcionalidades extra: librerías FX extra utilizadas, si se hace acceso a datos de distinto tipo indicar origen y método de carga, dónde están los buscadores extra si se han hecho, etc..

Nota: para que desde NetBeans se puedan leer todo tipo de recursos, hay que indicárselo en build.gradle incluyendo carpeta y extensión siguiendo el formato “**/*.extensión”:

```
sourceSets {
 main {
  resources {
     srcDirs = ["src/main/resources"]
     includes = ["**/*.fxml","**/*.css","**/*.png","**/*.html"] //Incluir cualquier fichero de recurso
  }
 }
}
```

1.1 NORMAS DE ENTREGA, EVALUACIÓN Y RÚBRICA DE CORRECCIÓN

Las fechas de entrega son muy importantes por lo que serán inamovibles (son sustitutos de examen). Una entrega suspensa o no entregada según fecha y hora, se podrá recuperar en la siguiente, pero teniendo en cuenta que la nota de la parte no entregada será un APTO (si es que se supera la defensa).

Muy importante: La práctica será calificada si y solo sí se ha defendido con éxito de forma parcial o en la entrega final, en cuyo caso la nota final se calculará siguiendo la siguiente rúbrica de corrección.

<table>
<tr><th>APARTADOS DE LA PRÁCTICA y ENTREGA ASOCIADA</th><th colspan="2">PESO(EN %)</th></tr>
<tr><td>Trabajo diario, defensa constante en clase y envío a tiempo de las entregas parciales. Este apartado solo sumará si se tiene aprobada la práctica teniendo en cuenta solo el resto de los apartados.</td><td colspan="2">15%</td></tr>
<tr><td>Punto a): Dificultad de la APP en general, del diseño de la base de datos (número y tipo de campos) y número de roles, porque todo esto afecta al resto de apartados. Además, el código deberá ser eficiente. Este apartado solo sumará si se tiene aprobada la práctica teniendo en cuenta solo el resto de los apartados.</td><td colspan="2">5%</td></tr>
<tr><td>Puntos b,c, d y f): La APP compila, es completamente funcional y se ha desarrollado de acuerdo al diseño y a los criterios de Usabilidad expuestos en la entrega de la Práctica 3 del Tema 1. Se deberá asegurar también la navegabilidad. PRIMERA ENTREGA</td><td colspan="2">10%</td></tr>
<tr><td>Apartado e) Se han realizado distintas y variadas ventanas (ventanas generales, formularios y contenedores de datos) para gestionar las 12 Operaciones CRUD (las cuáles deben funcionar correctamente). PRIMERA ENTREGA (Listar en las 3 Tablas)</td><td>10%</td><td rowspan="2">40%</td></tr>
<tr><td>Apartado e) Se han realizado distintas y variadas ventanas (ventanas generales, formularios y contenedores de datos) para gestionar las 12 Operaciones CRUD (las cuáles deben funcionar correctamente). SEGUNDA ENTREGA (Resto de CRUD)</td><td>30%</td></tr>
<tr><td>Apartado 2,c,i): Se realiza la validación de campos de la forma más vistosa posible. SEGUNDA ENTREGA</td><td colspan="2">5%</td></tr>
<tr><td>Apartado g) y 2,c,iii): La APP cuenta con múltiples eventos de distinto tipo. Además se utilizan menús contextuales que hagan de puente a opciones ya hechas. TERCERA ENTREGA</td><td colspan="2">5%</td></tr>
<tr><td>Apartado 2,c,i): Se utilizan hojas de estilo CSS utilizando todos los tipos de aplicación (estido directo, clase e identificador) para dar estilo al mayor número de componentes de la interfaz. TERCERA ENTREGA</td><td colspan="2">5%</td></tr>
<tr><td>Apartado 2,c,ii): Se utilizan animaciones en todos los elementos posibles. TERCERA ENTREGA</td><td colspan="2">5%</td></tr>
<tr><td>Apartado 2,c,v): Investigación y funcionalidades extra TERCERA ENTREGA</td><td colspan="2">10%</td></tr>
</table>