

**Práctica 3.1: Desarrollar una aplicación completa****Objetivos**

Implementar una aplicación que utilice una Interfaz gráfica atendiendo a usabilidad/accesibilidad, trabajando con imágenes, diagramas e informes, y que permita su distribución completa.

**Herramientas Necesarias**

- JDK 8
- IntelliJ
- JFreeChart
- JasperReport
- JasperSoftStudio
- Git
- Creador de paquetes auto-instalables

**Enunciado**

Se debe implementar una aplicación que administre elementos y utilice una interfaz gráfica de usuario para gestionarla. Se pensará en un supuesto real en el que existan, al menos, 3 tipos de objetos diferentes **relacionados entre sí**, y se creará la aplicación para gestionarlos. Se debe documentar el código fuente, generar su documentación de clases, su documentación de uso e instalación, generar un instalador y distribuir la aplicación mediante un servidor de repositorios de Git.

**Requisitos (1 pto cada uno)**

- Se debe gestionar información de al menos 3 objetos con 5 atributos cada uno. Deben aparecer, al menos una vez, datos de tipo cadena, número (enteros y coma flotante), fechas e imágenes. La aplicación se diseñará utilizando el patrón MVC (Model-View-Controller). Los datos se deben poder guardar y cargar desde fichero seleccionando su ubicación mediante un cuadro de diálogo. Todos los botones tendrán características de accesibilidad/usabilidad: imágenes (ausencia de texto), tooltips y atajos de teclado. Todos los elementos del menú tendrán también su icono y su acelerador (atajo). La aplicación debe permitir visualizar todos los objetos creados, sus propiedades y los elementos con los que están relacionados, crear nuevos, relacionarlos, eliminarlos, así como modificar **todas** sus propiedades y sus relaciones. Además, al menos una de las relaciones entre dos objetos debe **muchos a muchos**, y debe permitir ser modificada desde cada uno de los dos elementos relacionados, pudiendo relacionar varios elementos en una misma operación. (Ejemplo: alumnos y asignaturas: permitir añadir y quitar uno ó varios exámenes a un alumno concreto en una sola operación, y permitir añadir y quitar varios alumnos de un examen concreto).
- Mostrar como mínimo dos tipos de gráficas de distinto tipo, y al menos en una se debe de mostrar alguna estadística en la que intervengan los datos relacionados de los objetos.
- Mostrar como mínimo dos tipos de informes diferentes. Debe aparecer al menos una vez: una gráfica en el informe, imágenes de los objetos de mi programa y más de un campo de algún elemento relacionado.
- Se debe generar la documentación de la aplicación: documentación de clases (Javadoc que muestre también los miembros privados) incluida en el proyecto, manual de usuario y manual de instalación, mostrando capturas de pantalla. Se debe permitir acceder a la documentación de la aplicación desde algún elemento de la aplicación. Toda la documentación (tanto si está en un sitio web, como si está en formato pdf) deberá estar alojada en el repositorio de git.
- Se debe crear un instalador para Windows que permita desplegar la aplicación, desinstalarla y crear accesos directos a la aplicación y al manual de usuario (tanto si es en pdf como si es un enlace web html)

**Otras funcionalidades (1 pto cada una)**

- La aplicación tendrá un sistema de control de usuarios (login-password), que permita crear usuarios con al menos 3 roles distintos. Existirá un rol que solo pueda cargar datos y visualizarlos. Otro rol que podrá hacer modificaciones en los datos, y otro que podrá acceder a la creación de usuarios.
- Crear un cuadro de diálogo que muestre diferentes informes (al menos dos). Solo podrá visualizarse uno en cada momento y podremos cambiar de una a otra pulsando su botón correspondiente.
- Permitir que la aplicación notifique mediante un dialogo si hay datos modificados sin guardar en el momento de cerrar la aplicación. Si los hay ofrecerá al usuario guardarlos desde ese mismo dialogo. Si no hay datos sin guardar, la aplicación no notificará nada.
- Añadir la posibilidad de realizar un guardado automático de los datos. Esta opción se activará o desactivará desde la aplicación, y será recordada una vez cerrada la aplicación. Permitirá que los datos se guarden al salir y se carguen al arrancar de forma automática en una ruta por defecto que se podrá modificar desde la aplicación.
- Permitir modificar el tamaño de fuente de todo el texto de la aplicación y que el cambio persista aunque la aplicación se cierre.
- La aplicación contará con una opción de búsqueda (al menos de dos objetos). Se podrán buscar elementos por más de uno de sus campos y por un campo de sus elementos relacionados.
- Crear un Renderer para una lista o combo box que permita mostrar la imagen almacenada de alguno de los objetos.

**Entrega**

Se entregará el proyecto comprimido, la documentación, el instalador y el link del repositorio de Git.

**Corrección**

- La aplicación se defenderá públicamente en clase, desde el proyector, **ejecutando el archivo de instalación entregado**.
- Los 5 requisitos obligatorios deben funcionar completamente para poder aprobar la práctica.
- Las funcionalidades extra añadirán más nota a una práctica aprobada.
- El código de la práctica debe ser personal y genuino, penalizando el plagio con una nota de 0 puntos.
- Toda práctica no aprobada se debe recuperar, volviéndola a entregar con las correcciones necesarias para cumplir los 5 requisitos.