

Práctica 2.1: Desarrollar una aplicación GUI accesible.**Objetivos**

Implementar una aplicación que utilice una Interfaz gráfica atendiendo a los aspectos de usabilidad y accesibilidad, e incorporando imágenes y gráficos.

Herramientas Necesarias

- JDK 8
- IntelliJ

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.

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.
- 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 permitir ser modificada desde la sección de cada uno de los dos elementos relacionados. (Ejemplo: alumnos y asignaturas: permitir modificar las asignaturas de un alumno concreto, y también los alumnos matriculados en una asignatura concreta)
- Todos los botones tendrán características de accesibilidad/usabilidad: imágenes (ausencia de texto), tooltips y atajos de teclado. Además, toda ventana de la aplicación tendrá un botón por defecto. Todos los elementos del menú tendrán también su icono y su acelerador (atajo). Los únicos JButtons que pueden tener texto son los habituales en un cuadro de dialogo (Open, Accept, Cancel, Next, etc)
- La aplicación deberá estar completamente internacionalizada al menos en dos idiomas, permitiendo seleccionar el idioma deseado desde la propia aplicación. Al arrancar la aplicación está se iniciará en el idioma seleccionado. La internacionalización afecta a todos los textos mostrados en la aplicación, que no sean el valor de propiedades de alguno de los elementos gestionados.
- La aplicación debe permitir mostrar al menos un tipo de diagrama que muestre alguna estadística de datos de los elementos de la aplicación.

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. Cada rol tendrá acceso a más o menos aspectos de la aplicación atendiendo a su nombre.
- Crear un cuadro de diálogo que muestre diferentes gráficas (al menos dos). Solo podrá visualizarse una 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.
- Incluir una pantalla de carga (splash screen) antes de arrancar la aplicación, que contenga al menos una imagen y una barra de carga.
- 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 tendrá algún método de “deshacer acción” que permite al menos deshacer las dos últimas acciones (al menos acción de creación de nuevos elementos y eliminación de nuevos elementos)
- 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.
- Hacer un seguimiento del desarrollo de la práctica con un repositorio remoto de Git (bitbucket, github, gitlab, etc). En este caso, se utilizará la wiki de estos repositorios para documentar el uso y despliegue de la aplicación, y también la gestión de incidencias y problemas que vayan surgiendo durante el proyecto. Cada nueva funcionalidad de la aplicación irá asociada a un commit sobre el repositorio.

Entrega

Se entregará el proyecto comprimido, un Jar completamente funcional y el link del repositorio de Git (en caso de que se use)

Corrección

- La aplicación se defenderá públicamente en clase, desde el proyector, ejecutando el archivo JAR 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.