

PROYECTO DE USABILIDAD Y DISEÑO

APLICACIÓN DE DIBUJO

Realizado por Ana Rodríguez González

Indice

Introducción a la aplicación a diseñar	3
Distribución de las opciones de la aplicación en menús	6
Creación de una barra de herramientas	6
Creación de un formulario complejo	7
Mensajes necesarios para la comunicación con el usuario	13
Propuesta de una prueba de usabilidad para tu aplicación	15

Introducción a la aplicación a diseñar

Descripción de la aplicación diseñada:

Para poder aplicar los conceptos aprendidos en esta unidad se ha diseñado la interfaz de una aplicación para dibujo digital educativa.

La aplicación consiste en cargar imágenes en blanco y negro para que el usuario las coloree. Estas imágenes se obtendrán de una galería incluida en la propia aplicación y una vez seleccionada la imagen a dibujar, ésta aparecerá en la pantalla principal de la aplicación junto con la referencia de la obra original y algunos datos de interés sobre la obra elegida.

Los objetivos para esta aplicación son el acercar a los usuarios las obras más reconocidas del mundo del arte en sus distintas épocas mediante la experimentación personal con las mismas. Podría utilizarse en escuelas o para usuarios interesados en acercarse al arte por medios menos clásicos.

Es importante que la obra original aparezca junto con la imagen que coloreará el usuario ya que, además de servir como referencia para dibujo (que el usuario puede seguir o colorear a su gusto), sirve para tener contacto continuo con el nombre del autor, el año, el título y la técnica y así familiarizar al usuario con esta información y culturizarlo sobre las especificaciones de la obra elegida.

Los intereses que me han movido a diseñar este tipo de aplicación se basan en la motivación personal por profundizar en el conocimiento del desarrollo de programas de diseño y pintura digital y la capacidad para poder aplicar mis conocimientos formales en el desarrollo de una aplicación. Soy licenciada en Bellas Artes con el máster habilitante para profesorado de dibujo por lo que el desarrollo de una aplicación educativa de arte es algo que creo nace de manera natural en un perfil como el mío. Considero que los conocimientos profundos sobre un tema ayudan a que el diseño de una aplicación sea más acertado por lo que finalmente me he decantado por aprovechar la práctica de esta tarea para aprender más sobre el desarrollo de aplicaciones de este tipo y sobre los estándares aplicados para su diseño.

Introducción a la aplicación a diseñar

La aplicación contará con las siguientes funciones:

-**Barra de menú:** La barra de menú contará con las siguientes funciones:

Un elemento de menú para las funciones que contiene “**Archivo**” y que desplegará las siguientes opciones:

- **Abrir:** Para abrir una imagen guardada con anterioridad y que se desee continuar o modificar.
- **Cerrar:** Cerrará la imagen actual con la que se está trabajando y quedará el espacio del lienzo y de la plantilla de dibujo vacíos.
- **Guardar:** Opción de guardado rápido en el formato predeterminado por el programa.
- **Guardar como...:** Opción de guardado en la que el usuario podrá seleccionar características como el formato o el tamaño en el que se guardará la imagen.
- **Imprimir:** Función que abrirá el JDialog que se nos pide para la tarea y donde el usuario podrá seleccionar diversas opciones que afectarán al modo en el que el archivo se imprime o guarda como pdf (en caso de no tener impresora).
- **Salir:** Para cerrar el programa por completo.

Un elemento de menú para las funciones que contienen las “**Herramientas**” y que desplegará lo siguiente:

- **Pincel:** Si se selecciona esta opción del menú el usuario podrá colorear la plantilla de la imagen elegida.
- **Goma:** Con esta herramienta el usuario podrá borrar las zonas de color que haya dado a la imagen.
- **Color:** Esta herramienta permite al usuario saber el color actual que ha seleccionado de la paleta de colores.
- **Paleta:** Permite al usuario seleccionar un color de entre una amplia variedad. Lo ideal es que le permita también seleccionar nuevos tonos y agregarlos a una paleta personalizada si así lo desea. Esta herramienta deberá de abrir una nueva ventana ya que estas acciones no se pueden controlar con una sola acción.
- **Deshacer:** Deshace la ultima acción realizada por el usuario en su plantilla de dibujo.
- **Rehacer:** Rehace la última acción realizada por el usuario si se ha utilizado “Deshacer” con anterioridad. Si no se ha utilizado “Deshacer” no hará nada.

Introducción a la aplicación a diseñar

Galería”

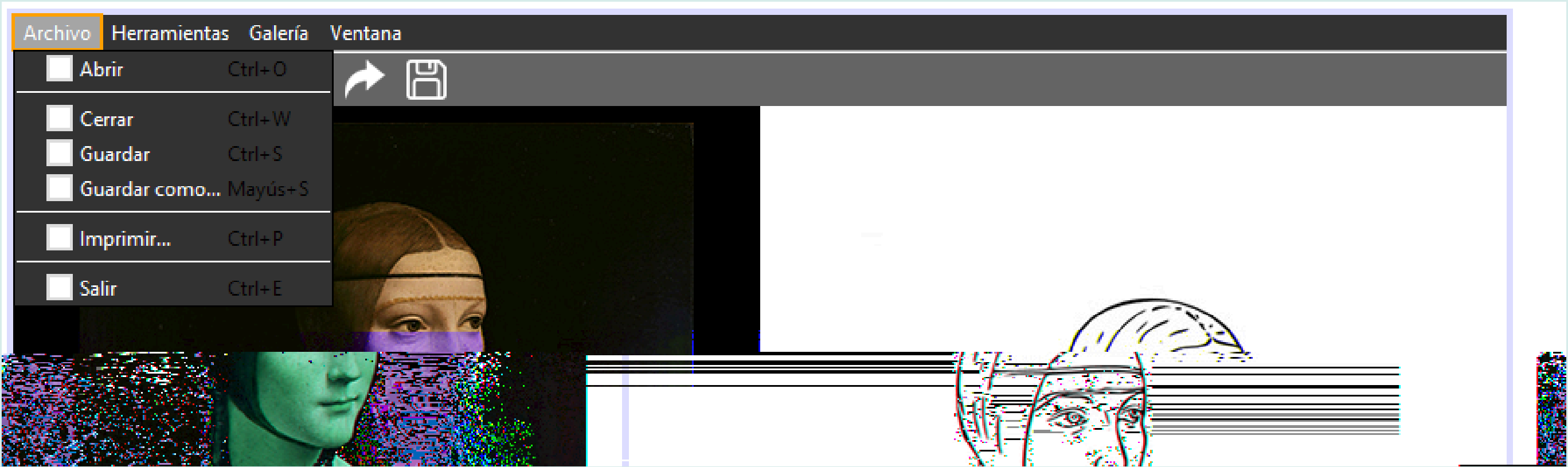
- Ver:
- Seleccionar obra... :

Ventana

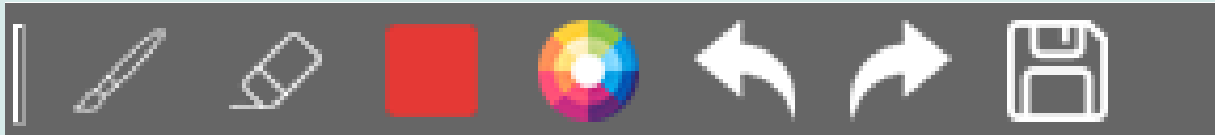
- Zurdos:
- Diestros:

Distribución de las opciones de la aplicación en menús

Detalle del diseño de la barra de menú del proyecto con las opciones para el menú “Archivo” desplegadas:



Detalle del diseño de la barra de herramientas:



Opción de Menú	Icono
Pincel	
Goma	

Opción de Menú	Icono
Color	
Paleta	

Opción de Menú	Icono
Deshacer	
Rehacer	

Opción de Menú	Icono
Guardar	

Creación de un formulario complejo

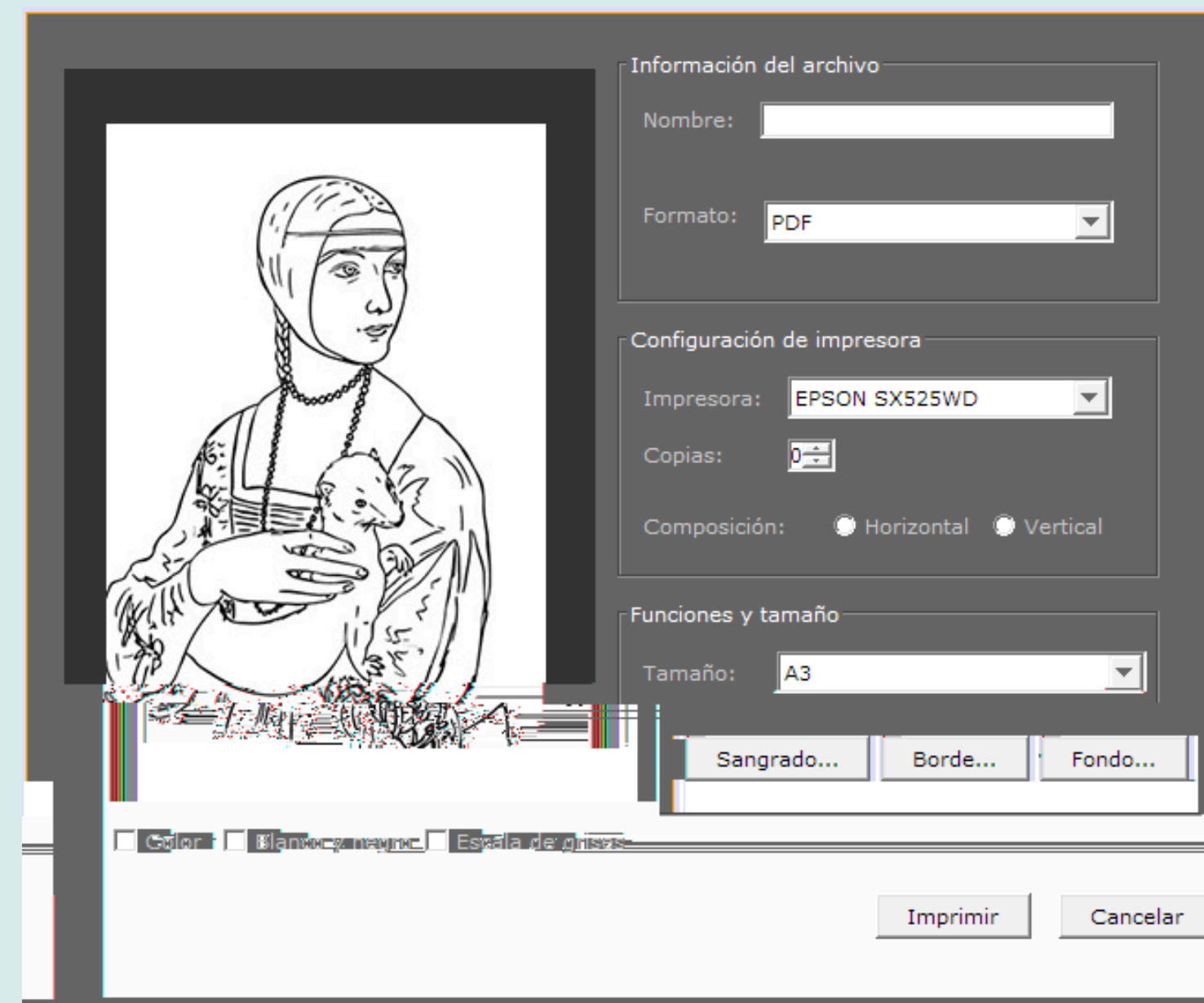
a) Justificación de la distribución de los componentes.

Para la creación del JDialog de este diseño de aplicación se ha creado un “formulario” que permite al usuario definir una serie de opciones para la impresión de la plantilla realizada. La idea es que, una vez que el usuario haya finalizado su obra pueda imprimir el resultado si así lo desea, o guardarlo en PDF para imprimir más adelante, si es que no tiene impresora (esta opción aparece al seleccionar la impresora, como suele aparecer por defecto en programas de este tipo).

La disposición de los elementos se ha realizado de manera que queden bien diferenciados por bloques gracias al uso del JPanel. Se ha utilizado la opción de “TitledBorder” para poder agregar un título a cada bloque y que el usuario sepa el tipo de configuración que contiene cada bloque. Dentro de cada bloque se encuentran organizadas las opciones a seleccionar.

Es importante destacar que, a la izquierda, contamos con un JPanel que contiene un JLabel de la vista previa de la imagen que se desea imprimir y sobre el que se podrían ir previsualizando los cambios que el usuario realice sobre la imagen al seleccionar algunas de las opciones (color, blanco y negro, escala de grises, sangrado, borde, etc). Este se ha realizado por lo práctico que resulta para el tratamiento y previsualización de la imagen.

Vemos, pues, que la información a manipular aparece a la derecha, la vista previa a la izquierda y debajo de ésta las opciones que realizan un cambio directo en la imagen y que se aplicaría a la vista previa (color, blanco y negro y escala de grises). Las opciones de sangrado, borde y fondo deben de abrir nuevos paneles de configuración por lo que se han organizado en otro bloque aunque afecten también directamente la previsualización de la imagen. Para finalizar, abajo a la derecha, siguiendo los estándares, se ha incluido el botón de imprimir y de cancelar. El botón de imprimir permite también guardar el archivo, con las opciones seleccionadas, si se ha clicado sobre “...print to PDF” en las opciones de “Impresora”.



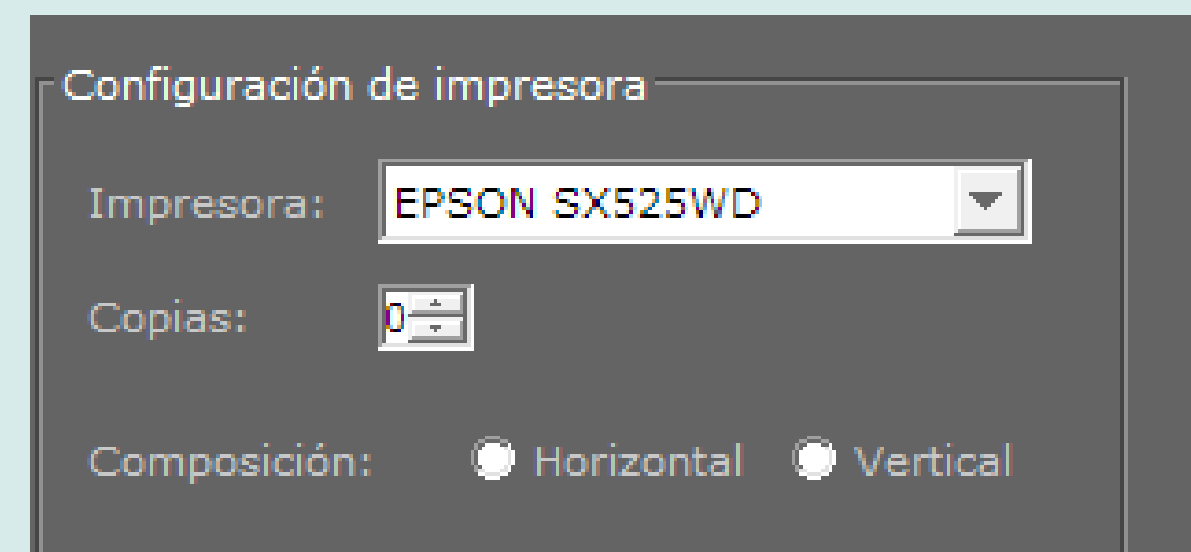
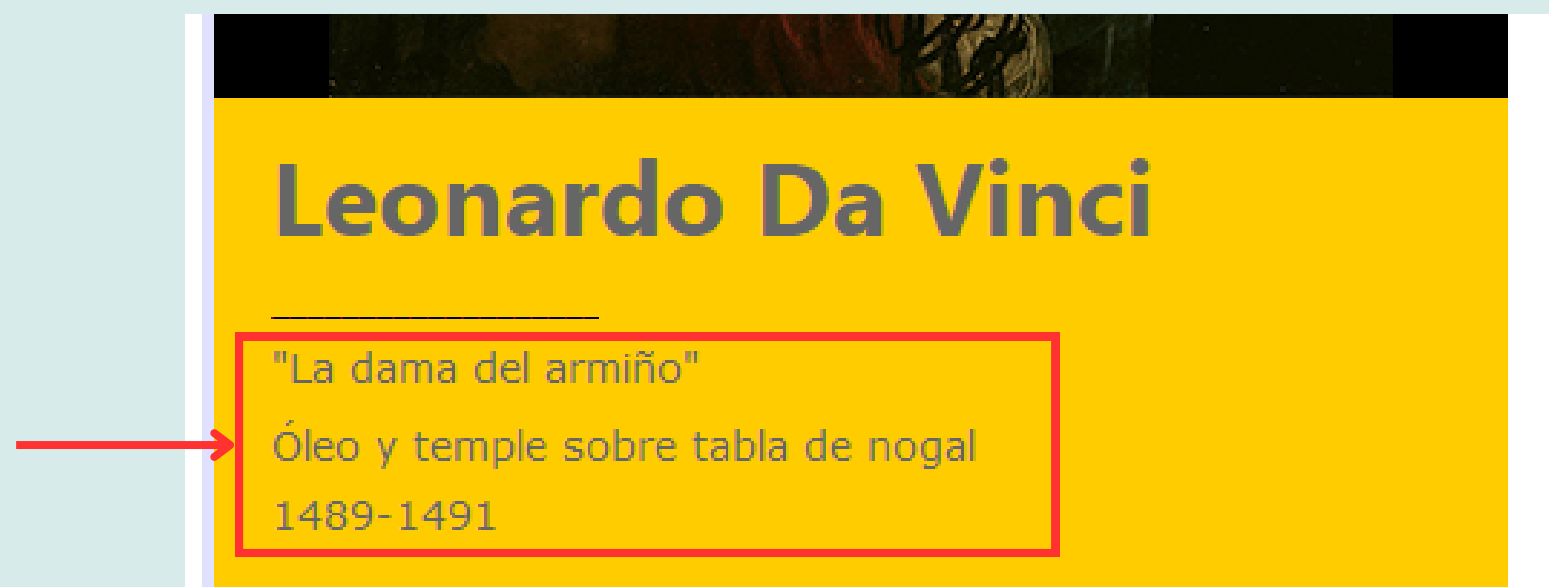
Creación de un formulario complejo

b) Justificación del uso de la tipografía y el color en la interfaz.

En cuanto a los colores y la tipografía, se ha trabajado sobre fondo gris [102,102,102], el mismo que la barra de herramientas del JFrame principal de la aplicación, a excepción del contenedor de vista previa, que el tono es más oscuro [51,51,51], el mismo que la barra de menú.

Aparte de la profundidad que añade el uso de un color oscuro sobre otro algo más claro, como es el caso del contenedor de la imagen de vista previa frente al panel principal de opciones, también se pretende mantener la armonía entre el JFrame y el JDialog ya que forman parte de una misma aplicación y hacer uso de otros colores podría sacar de contexto al usuario y transmitirle una idea de desorden.

Tal y como se comentó, para el JFrame se utilizaron colores oscuros por la seriedad y limpieza visual que aportan al diseño y porque nos permitían dar valor a la imagen. El uso de estos colores nos incita a utilizar un color más claro para la tipografía, para que destaque sobre el fondo, por lo que se ha utilizado el color blanco. El estilo es “Verdana” ya que fue el tipo de tipografía utilizada para el texto que contiene la información de la obra y se seleccionó por su claridad y fácil lectura que permite hacer llegar el mensaje al lector sin entorpecimientos. Al volver a utilizarla en el formulario conseguimos beneficiarnos de las características propias de la tipografía mencionadas y además, conseguir armonía y unidad entre los componentes de nuestra aplicación.



Creación de un formulario complejo

b) Tabla de componentes del formulario

Nombre de la clase del componente	Justificación	Alternativas	Recomendaciones de usabilidad
jPanel	Su uso nos permite organizar los componentes en contenedores y así poder controlar mejor el espacio de diseño de nuestra aplicación.	JScrollPane si la imagen fuese grande o quisiéramos que el usuario pudiera ampliarla para verla con más detalle.	<ul style="list-style-type: none">• Usos Recomendados: Para agrupar y organizar componentes relacionados.• No Recomendado: sobrecargar con demasiados componentes. Mantener la coherencia en la disposición.
jLabel	Su uso nos permite insertar imágenes y texto que el usuario no puede modificar. Si se necesita modificar el contenido del JLabel se puede manejar mediante el código interno del programa, lo que nos resulta útil en nuestro caso para aplicar los cambios en la imagen de vista previa o el texto de las obras.	TextField o JTextArea para texto extenso. Aunque estos elementos se pueden modificar e inhabilitar para usarse como una etiqueta realmente no tiene mucho sentido hacerlo cuando contamos con el JLabel para este uso.	<ul style="list-style-type: none">• Usos Recomendados: Para etiquetar campos, mostrar información estática.• No Recomendado: para entradas de usuario directas.

Creación de un formulario complejo

Nombre de la clase del componente	Justificación	Alternativas	Recomendaciones de usabilidad
jTextField	Su uso nos permite que el usuario pueda modificar la información de manera personal ya que puede escribir en él. Es común su uso cuando necesitamos obtener información por parte del usuario, como es el caso para nombrar el archivo.	JTextArea , aunque esta sería más conveniente para campos de texto más amplios de los que nosotros necesitamos.	<ul style="list-style-type: none"> • Usos Recomendados: Entradas cortas como nombres, contraseñas, etc. Etiquetar claramente para guiar al usuario. • No Recomendado: usar para grandes cantidades de texto.
jComboBox	Su uso permite que el usuario pueda seleccionar una serie de items de una lista. Es muy útil si no deseamos ocupar un espacio innecesario en la interfaz ya que se despliega y en el cuadro queda la selección realizada por el usuario.	JList sería una opción pero el espacio que ocupa en comparación al jComboBox es innecesario en nuestro caso.	<ul style="list-style-type: none"> • Usos Recomendados: Cuando hay un conjunto finito de opciones. Limitar opciones para no abrumar al usuario. • No Recomendado: uso para listas largas o dinámicas.
JRadioButton	Su uso nos permite que el usuario pueda elegir entre varias opciones. Al agregar los botones a un grupo se puede elegir una única opción entre varias.	JCheckBox para selecciones múltiples, pero no es nuestro caso.	<ul style="list-style-type: none"> • Usos Recomendados: Cuando solo se permite una opción. Etiquetar claramente para evitar confusiones. • No Recomendado: Evitar para selecciones múltiples.

Creación de un formulario complejo

Nombre de la clase del componente	Justificación	Alternativas	Recomendaciones de usabilidad
jCheckBox	Su uso permite al usuario seleccionar opciones múltiples. La elección de este componente frente a <code>JRadioButton</code> , a pesar de que quizás este habría sido más conveniente, ha sido por diferenciar visualmente las opciones de control de color frente a los contenedores de ajustes de impresión.	JRadioButton , quizás hubiese sido más acertado pero como se ha comentado, se ha pensado que el <code>jCheckBox</code> era más conveniente para marcar una diferencia visual entre las opciones dadas al usuario.	<ul style="list-style-type: none">• Usos Recomendados: Para selecciones múltiples. Agrupar opciones relacionadas.• No Recomendado: Evitar para opciones únicas.
jButton	Su uso permite realizar una acción que generará algún tipo de evento. Como las opciones de imprimir o cancelar en nuestro caso.	JToggleButton que al ser un botón de dos estados permite que el botón quede pulsado o no. En este caso solo nos interesa la acción del clic, no que se mantenga pulsado o no, por lo que era preferible el uso del <code>jButton</code> .	<ul style="list-style-type: none">• Usos Recomendados: Para realizar acciones al hacer clic, como enviar formularios o iniciar procesos. Colocar de manera estratégica para resaltar acciones importantes.• No Recomendado: Evitar el uso excesivo para no sobrecargar la interfaz.

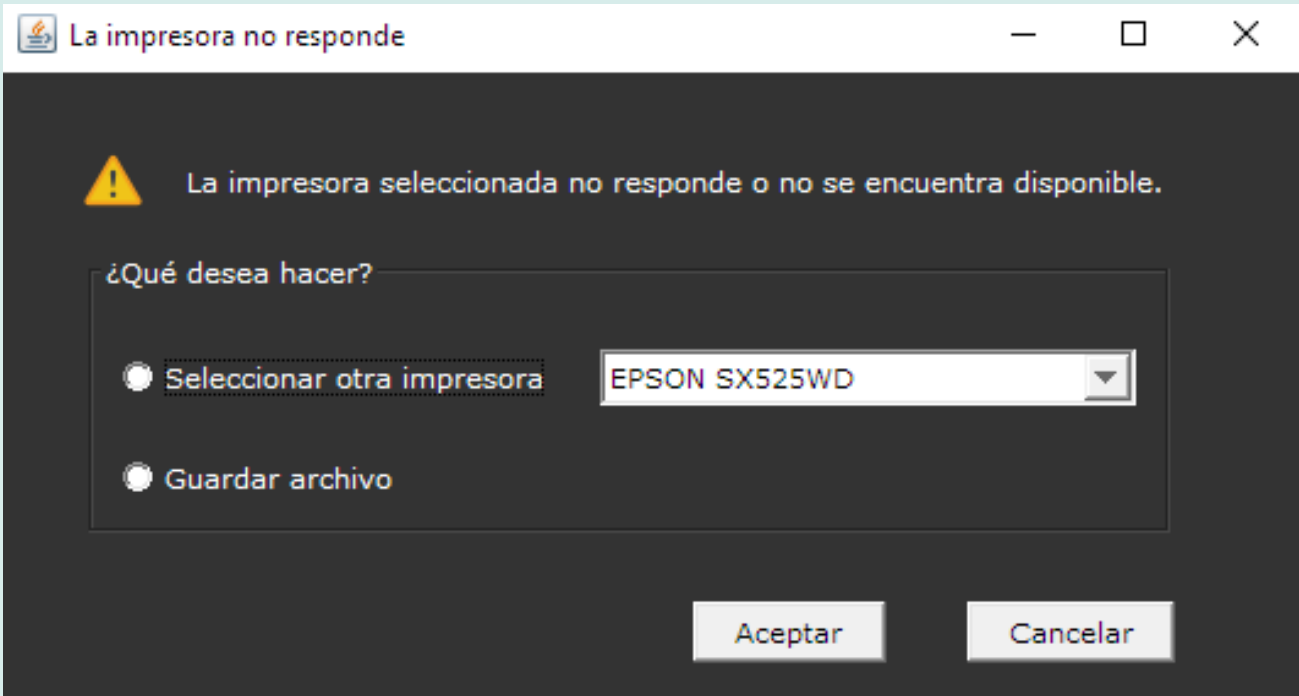
Creación de un formulario complejo

Nombre de la clase del componente	Justificación	Alternativas	Recomendaciones de usabilidad
jSpinner	Su uso permite que el usuario pueda marcar el numero de copias con comodidad. Su diseño ya nos aporta facilidades puesto que nos permite seleccionar un valor dentro de un rango del mismo tipo con las flechas de arriba o abajo.	JSlider aunque es más útil para otros casos y quedaría algo forzado su uso en este caso. Quizás podría usarse un jTextField pero habría que controlar la entrada por código para que fuese el tipo de dato que queremos y estuviese en el rango deseado cuando el jSpinner ya nos ofrece esto sin necesidad de complicarnos demasiado por lo que es más cómodo en casos como éste.	<ul style="list-style-type: none">• Usos Recomendados: Para selección de valores en un rango. Proporcionar límites claros y etiquetas.• No Recomendado: Evitar para valores continuos o deslizables. Evitar también para valores no restringidos por un rango o para permitir al usuario ingresar al usuario ingresar distintos tipos de datos de manera libre.

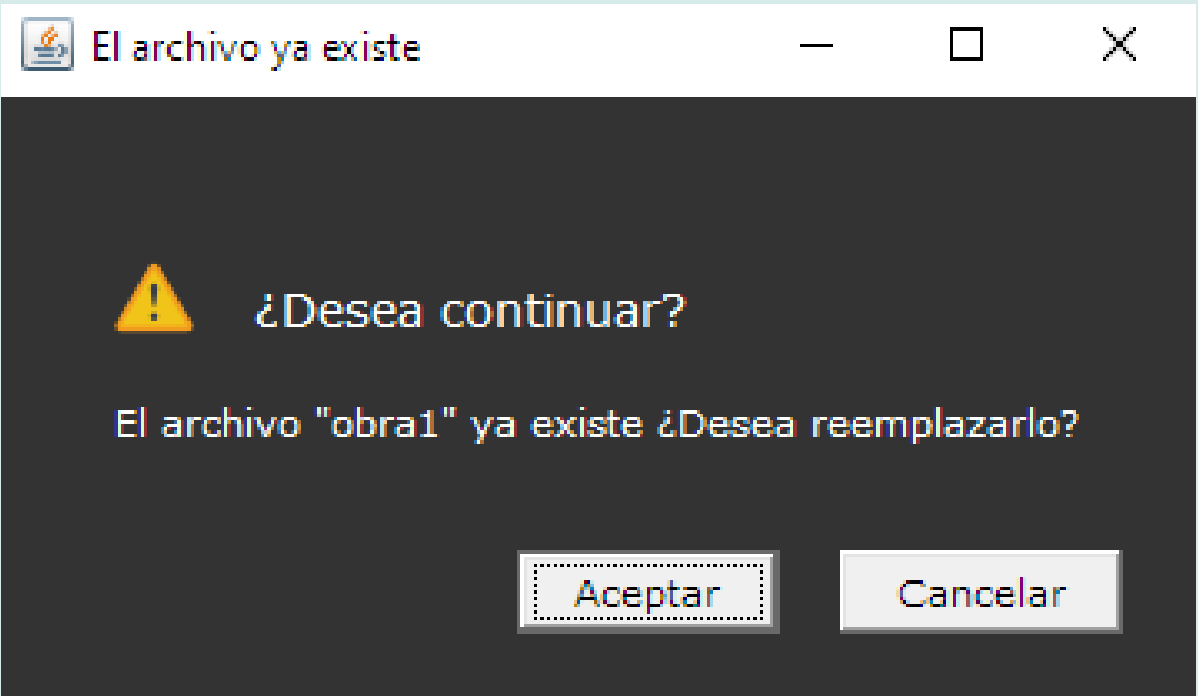
Mensajes necesarios para la comunicación con el usuario

Se muestra dos ejemplos de mensajes que se utilizarían para advertir de algún error que pudiese cometer el usuario en el manejo de la aplicación.

Ejemplo de dos tipos de mensajes:



Mensaje informativo: Este mensaje se lanzará cuando la impresora esté desconectada o no esté disponible por algún motivo. Da la información al usuario y le ofrece opciones para realizar otra acción que solvante el problema actual.



Mensaje de confirmación: Este mensaje se lanzará cuando el nombre de un archivo guardado corresponda con el nombre del archivo actual que se desea guardar. Aparece cuando el usuario selecciona la opción “guardar” del menú “Archivo” y pretende servir como medida de seguridad que informa al usuario de la acción a realizar y solicita doble confirmación.

Mensajes necesarios para la comunicación con el usuario

Tabla de justificación:

Mensaje incorrecto	Mensaje correcto	Justificación
“No se puede realizar la acción”	“La impresora seleccionada no responde o no se encuentra disponible ¿Qué desea hacer?” <ul style="list-style-type: none">• Seleccionar otra impresora• Guardar archivo”	Si utilizásemos un mensaje como el indicado en el “Mensaje incorrecto”, el usuario no sólo no tendría información sobre el problema que se puede estar dando y que no le permite realizar la acción que quiere sino que tampoco se le estarían dando opciones para solucionar el problema. Informar adecuadamente al usuario de las posibles situaciones por las que una acción no está funcionando y ofrecer alternativas para la misma sería lo ideal en casos como el de este ejemplo.
“Se ha reemplazado el archivo”	“¿Desea continuar? El archivo “obra1” ya existe ¿Desea reemplazarlo?”	Si el usuario selecciona la opción de guardar cuando un archivo tiene el mismo nombre que otro ya guardado y el programa no le avisa adecuadamente y le pide confirmación de esta acción es muy posible que se realicen reemplazos no deseados de archivos. Si simplemente se le informa con un mensaje una vez realizada la acción, como el que se pone de ejemplo en el “Mensaje incorrecto” de esta tabla, no sólo no se estaría avisando al usuario con antelación y pidiendo su confirmación sino que, al haberse realizado ya la acción, podría llegar a ser muy molesto.

Propuesta de una prueba de usabilidad para tu aplicación

Propuesta de prueba de usabilidad:

Para verificar el funcionamiento de la aplicación se propone la realización de una prueba que nos permita ver el recorrido natural de los usuarios a la hora de interactuar con la aplicación y si nuestro diseño permite una interacción usuario/aplicación cómoda y eficiente.

Se pretende que la experiencia para el usuario a la hora de interactuar con nuestra aplicación sea positiva, intuitiva y fácil de usar. Para ello los mejores jueces serán los usuarios finales pero antes de sacar la aplicación es conveniente que un grupo de potenciales usuarios finales realice una serie de pruebas de usabilidad diseñadas para analizar su comportamiento con nuestra interfaz y tomar la información reportada por los mismos mediante un formulario específicamente diseñado para tal caso.

Se propone la realización de una prueba de usabilidad a un grupo de usuarios de entre 5 y 10 personas a los que, mediante una prueba de diseño de escenarios, se les pedirá probar ciertas funcionalidades de la aplicación y contestar una serie de preguntas relacionadas.

La prueba se organizará en 4 fases que pasaremos a describir.



Propuesta de una prueba de usabilidad para tu aplicación

Fase 1. Planificación:

En esta fase se definirán las tareas que van a realizar los usuarios y se establecerán los objetivos para las tareas seleccionadas. Se definirán las variables que se van a medir durante el proceso y se planificará cómo se van a recoger los datos.

Para esta prueba se van a planificar la realización de tareas relacionadas con el uso general de la aplicación. Se les pedirá a los usuarios realizar acciones en la aplicación tales como dibujar una imagen utilizando el pincel y la goma, cambiar el color utilizado con la herramienta “paleta”, deshacer y rehacer acciones, guardar la imagen, volver a cargarla, etc. Las variables que se medirán incluyen el tiempo necesario para completar cada tarea, el porcentaje de errores cometidos y el número de clics necesarios para completar cada tarea. También se comprobará si el usuario echa en falta alguna otra herramienta de acceso rápido.

Fase 2. Ejecución:

En esta fase se realizarán las pruebas con los usuarios. Se les reunirá y se les pedirá que realicen las tareas definidas durante la fase de planificación. Se les pedirá rellenar el formulario correspondiente por cada tarea realizada. También se dejará un apartado en el que los usuarios que participan en la prueba puedan realizar comentarios y sugerencias de mejora sobre la interfaz y las funcionalidades de la aplicación.

Fase 3. Análisis:

En esta fase se analizarán los datos recogidos durante la fase anterior y se buscarán problemas de usabilidad en la aplicación. Se pueden detectar las tareas que resultan más difíciles para los usuarios, los elementos de la interfaz que son menos comprensibles y se puede obtener una lista de prioridades para el rediseño de la interfaz. Gracias a los comentarios y sugerencias obtenidos por los usuarios durante la fase de ejecución se pueden plantear, a su vez, mejoras para la interfaz y las funcionalidades de la aplicación.

Fase 4. Rediseño:

En esta fase se procederá al rediseño de la interfaz según la lista de prioridades obtenida en la fase de análisis. Se buscará dar solución a los puntos más conflictivos detectados en la fase anterior y aplicar las mejoras y recomendaciones que se han decidido implementar gracias a los comentarios obtenidos por los usuarios. Se pueden realizar pruebas adicionales con los usuarios para comprobar si las mejoras realizadas han solucionado los problemas de usabilidad detectados y las mejoras cumplen con las expectativas.

Fin