

Readme

Sketch Box

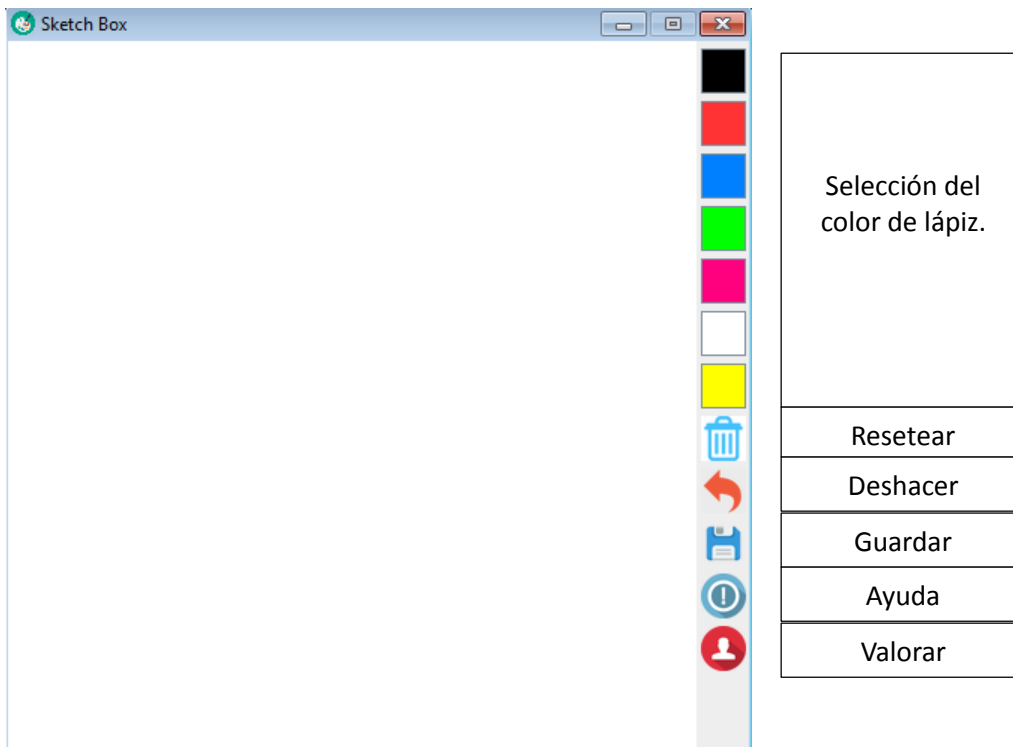
Sketch Box es un entorno de dibujo el cual permite al usuario realizar proyectos de dibujo a partir de herramientas.

Al ejecutar el programa se muestra un panel de herramientas el lado derecho de la ventana, Entre las cuales encontramos:

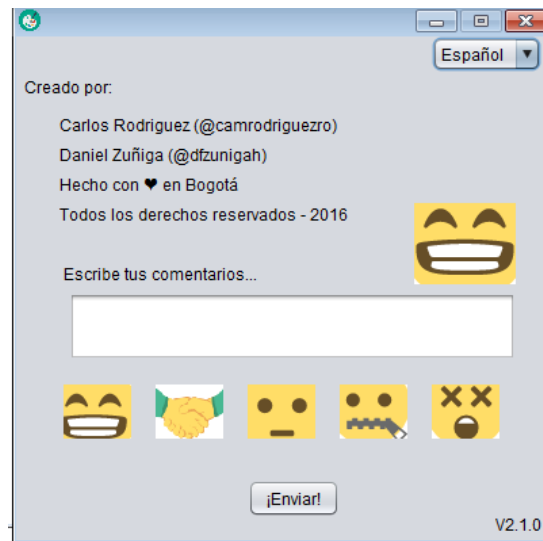
7 botones para seleccionar el color de lápiz que se desea usar, los colores que se ofrecen son:

- Negro
- Rojo
- Azul
- Verde
- Magenta
- Blanco
- Amarillo

Un botón con el icono de una papelera el cual permite resetear el dibujo y dejar el panel de dibujo en blanco. El botón para deshacer la última acción realizada. El botón para guardar el trabajo (por defecto el archivo se guarda en la misma carpeta donde se encuentra el ejecutable). Después se encuentra el botón de ayuda y finalmente el botón para valorar la aplicación.



Al presionar el botón de valorar aparece la ventana:



En esta ventana el usuario encontrará la información sobre los desarrolladores de la aplicación además de cambiar el idioma en que se encuentra este texto, puede elegir entre inglés, español y francés. Además encontrara un cuadro de texto en donde podrá dejar sus comentarios y valorar la aplicación de manera didáctica.

Este programa fue desarrollado como proyecto final de la asignatura POO (Programación orientada a objetos) de la Universidad Nacional de Colombia en el año 2016. Y fue elaborado por Carlos Mauricio Rodríguez Rodríguez (camrodriguezro@unal.edu.co) y Daniel Felipe Zúñiga Hurtado (dfzunigah@unal.edu.co).

Documentación

ToastMessage (JDialog)
- milliseconds: int
- public ToastMessage(toastString: String, time: int)
- public run(): void

SizedStack<T> (Stack<T>)
- final maxSize: int
- public SizedStack(size: int)
- public push(object: Object): Object

ReviewListener (MouseListener)
- button: JButton - statusIcon: JButton - writer: FileWriter - reader: Scanner - puntuations: ArrayList<Integer>
- public ReviewListener(one: JButton, statusIcon: JButton) - public mousePressed(e: MouseEvent): void - public mouseReleased(e: MouseEvent): void - private getPunctuation(): void - private status(): double - private static round(value: double): double - private updateImagePunctuation(): void

MouseMovementAdapter (MouseMotionAdapter)
- pad: Pad
- public MouseMovementAdapter(pad: Pad) - public mouseDragged(e: MouseEvent): void

LateralPanel (JPanel)
- blackButton: JButton - redButton: JButton - blueButton: JButton - greenButton: JButton - magentaButton: JButton - whiteButton: JButton - yellowButton: JButton - clearButton: JButton - save: JButton - info: JButton - aboutUs: JButton - drawPad: Pad
- public LateralPanel(drawPad: Pad) - private setLook(): void

Pad (JComponent)
- image: Image - pen: Graphics2D - oldX: int - oldY: int - actualX: int - actualY: int - final undoStack: SizedStack<Image>
- public Pad() - public undo(): void - private setImage(img: Image): void - private copyImage(img: Image): BufferedImage - protected saveToStack(img: Image): void - public setOldX(actualX: int): void - public setActualX(currentX: int): void - public setOldY(actualY: int): void - public setActualY(currentY: int): void - public getOldX(): int - public getOldY(): int - public getActualX(): int - public getActualY(): int - public getGraphics(): Graphics2D - public getImage(): Image - public paintComponent(g: Graphics): void - public clear(): void

Frame
- frame: JFrame
- public Frame() - public static main(args: String[]): void

ButtonListener (ActionListener)
- drawPad: Pad - button: JButton
- public ButtonListener(drawPad: Pad, button: JButton) - public actionPerformed(e: ActionEvent): void

OpinionListener (ActionListener)
- send: JButton; - opinionWriter: FileWriter - comments: JTextField - reader: Scanner - fileOnComments: File - builder: StringBuilder - dateFormat: DateFormat
- public OpinionListener(send: JButton, comments: JTextField) - public actionPerformed(e: ActionEvent): void - private readComments(): void

MousePressedAdapter (MouseListener)
- pad: Pad
- public MousePressedAdapter(pad: Pad) - public mousePressed(e: MouseEvent): void

AboutPanel (JFrame)
- Container: JPanel - sendButton: JButton - rate5: JButton - rate4: JButton - rate3: JButton - rate2: JButton - rate1: JButton - opinionImage: JButton - createdBy: JLabel - person1: JLabel - person2: JLabel - rights: JLabel - madeIn: JLabel - commentLabel: JLabel - version: JLabel - commentArea: JTextField - languageSelector: JComboBox<String> - language: Locale - translator: ResourceBundle
- public AboutPanel() - private initComponents(): void - private changeLanguage(): void - private setGraphicLayout(): void - public initiate(): void - public run(): void

Sección de Contribución

- Carlos Rodriguez Rodriguez (@camrodriguezro)
- Daniel Felipe Zuñiga Hurtado (@dfzunigah)

El proyecto posee un repositorio en GitHub, el cual se encuentra disponible en <https://github.com/camrodriguezro/testSketchBox> (Prevía autorización para contribuir) bajo una licencia "FALTA", ahí está el código Fuente, un "Read-me" y en la sección de "Lanzamientos/Releases" se encuentra un ejecutable para descargar.

Historial de Cambios

(C) = Trabajo realizado por Carlos Rodriguez.

(D) = Trabajo realizado por Daniel Zuñiga.

(C/D) = Trabajo realizado en equipo.

[FASE ALFA I - Prototipado]

- [V0.1.0][#Añadido] (D) Definida la clase Pad.java: Hace de lienzo para pintar.
- [V0.2.0][#Añadido] (C) Definida los "Getters" y "Setters" para la clase Pad.java.
- [V0.3.0][#Añadido] (C) Creadas las clases MousePressed y MouseDragged, ellas hacen de "Listeners" para los movimientos del mouse.
- [V0.4.0][#Añadido] (C) Se crea el repositorio del proyecto en GitHub.
- [V0.4.1][#Pruebas] (C) Primeras pruebas con botones incluyendo imágenes.
- [V0.5.0][#Añadido] (C/D) Creado el prototipo de la ventana "About us" siguiendo el mock-up hecho anteriormente; se emplea el editor gráfico de interfaces de NetBeans y se define como un Java Form.
- [V0.6.0][#Cambio] (C) Recreado el Java Form de la ventana "About us" mediante una simple clase de Java. Esta ventana se concibe como la primera rama del proyecto, trabajando en paralelo
- [V0.7.0][#Cambio] (C/D) Se reestructura la ventana "About us", se parte del código del Java Form y se elimina el código innecesario, se personaliza y organiza el código, importando librerías y renombrando los elementos para hacerlos más comprensibles.
- [V0.8.0][#Añadido] (D) Implementado el proceso de "Internacionalización" inicialmente con tres (3) idiomas: español, inglés y francés.

- [V0.9.0][#Añadido] (C) Asignado el primer módulo de persistencia para la ventana "About us": Permite guardar los comentarios de la caja de texto.
- [V0.10.0][#Cambio] (C) Se crea una clase aparte para el manejo de eventos de la ventana.
- [V0.11.0][#Añadido] (C) Asignado el segundo módulo de persistencia para la ventana "About us"; Permite guardar las puntuaciones hechas mediante los botones. Separación de los procesos de persistencia de comentarios y puntuaciones.
- [V0.12.0][#Añadido] (C/D) Creado el sistema de para el manejo de las puntuaciones: Asignaciones, cómputo y respuesta.
- [V0.12.1][#Añadido] (C/D) Primeras contribuciones al repositorio de GitHub: Terminado el prototipo de la ventana "About us". Finalización de la fase ALFA I.

[FASE ALFA II - Integración]

- [V0.12.2][#Cambio] (C) Definidos nombres más técnicos para los recursos utilizados.
- [V0.12.3][#Cambio] (C) Redefinidos los alcances de las variables y los métodos: Encapsulación y aumento de seguridad.
- [V0.12.4][#Cambio] (C) Redefinidos los archivos de salida, nombres más técnicos y uso de la extensión ".ser", a fin de que sean inmodificables para un usuario sin mucho conocimiento en manejo de archivos.
- [V0.13.0][#Añadido] (C) Añadidas etiquetas a los botones (ToolTipText), a fin de proporcionar ayuda al usuario.
- [V0.13.1][#Cambio] Redefinida la funcionalidad de la etiqueta para el botón "statusIcon": ahora muestra el promedio de calificaciones que ha recibido la aplicación.
- [V0.13.2][#Cambio] (D) Imágenes rediseñadas, mejor posicionadas, manejo de escalas.
- [V0.14.0][#Arreglado] (C) Arreglado el problema del sistema de cómputo para las puntuaciones: Se establecieron nuevos parámetros para definir una escala acorde.
- [V0.15.0][#Añadido] (D) Implementación, diseño y ajuste del icono de la aplicación.
- [V0.15.1][#Pruebas] (C) Creado el primer ejecutable, pruebas de funcionamiento, comprobación de errores.
- [V0.15.2][#Pruebas] (C) Retroalimentación de usuarios: Código muy desordenado, no se pueden ver los comentarios.
- [V0.16.0][#Añadido] (D) Ahora es posible visualizar los comentarios hechos mediante una ventana de dialogo plana.
- [V0.16.1][#Cambio] (D) Creadas clases independientes que funcionan como "ActionListeners" con el fin de controlar los eventos sobre los botones: Reestructuración del código.
- [V0.16.2][#Pruebas] (C) Experimentación con "Pad.java", redefinido como un JComponent a fin de poder incrustarlo en otros contenedores.
- [V0.16.3][#Pruebas] (C) Definido el marco general del programa. Integración del Pad con el marco general.
- [V0.16.3][#Añadido] (C) Definida la clase LateralPanel.java y su funcionalidad.

- [V0.16.4][#Añadido] (C) Añadidos los primeros elementos gráficos al panel lateral, aún sin funcionalidad.
- [V0.17.0][#Añadido] (D) Creación de una clase independiente para el manejo de eventos, añadidos primeras funcionalidades.
- [V0.18.0][#Añadido] (D) Implementación de cambio de color y borrar la pantalla. Funcionalidades asignadas a los botones.
- [V0.18.1][#Cambio] (C/D) Estructuración del panel lateral, definidos los tamaños y el formato.
- [V0.18.2][#Cambio] (C) Botones estilizados, pantalla centrada, se determina que la aplicación no pueda reajustar su tamaño, demás cambios estéticos.
- [V0.18.3][#Pruebas] (C/D) Intercambio de información, ahora ambas ramas del proyecto (AboutPanel y DrawPad están en un solo directorio). Integración del código, sin embargo aún no hay conexión directa entre ramas. Trabajo en paralelo continua.
- [V0.18.4][#Pruebas] (C/D) Actualización del repositorio del proyecto en GitHub. Inicio de recopilación de cambios para la elaboración de un historial de cambios.
- [V0.18.5][#Cambio] (C) Reestructuración de paquetes del proyecto: Se separan todos los recursos del código. Se renombran los archivos y variables a fin de que sea más clara la relación entre las ramas, coherencia del código.
- [V0.19.0][#Añadido] (D) Integración directa de las dos ramas: Creación del botón “Acerca de” que permite una llamada del panel de dibujo al panel “About us”. Añadidas sus funcionalidades y su estética.
- [V0.19.1][#Arreglado] (C) Se cambió la operación de cerrado por defecto de la segunda ventana (About us), a fin de que no termine la operación, sino que simplemente se cierre y vuelva al panel de dibujo o panel principal.
- [V0.19.2][#Removido] (C) Eliminación de anteriores prototipos.
- [V0.19.3][#Pruebas] (C/D) Revisión y reestructuración del código. Eliminación de elementos innecesarios, estudio de la coherencia entre los nombres de archivos. Finalización de fase ALFA II.

[FASE BETA – Diseño funcional]

- [V1.0.0][#Pruebas] (C) Creado un ejecutable del proyecto, primer uso público.
- [V1.0.1][#Pruebas] (C/D) Retroalimentación de usuarios: Mejorar la estética, agregar más funciones. Se llevó a cabo mediante el panel de comentarios.
- [V1.1.0][#Agregado] (D) Agregada la funcionalidad de borrar con su respectivo icono estilizado y formato.
- [V1.2.0][#Pruebas] (D) Agregada la funcionalidad de cambiar el color mediante la selección de botones, se estilizaron y formatearon. Inicialmente negro, azul, verde, Rosado y rojo.
- [V1.2.1][#Cambio] (D) Estilización de los elementos gráficos de la aplicación.
- [V2.0.0][#Agregado] (D) Agregada la funcionalidad para guardar la imagen producida, con su respectivo botón estilizado y formateado. Finalización de la fase BETA: La aplicación es autónoma y capaz de llevar a cabo tareas a nivel básico.
- [V2.0.1][#Cambio] (C/D) Revisión de código, reestructuración y eliminación.

[FASE GAMA - Mejoras]

- [V2.1.0][#Agregado] (D) Agregado un elemento “Toast-like” (Pequeño mensaje), asignado al botón de “Guardar” a fin de que le permita saber al usuario su acción.
- [V2.1.1][#Cambio] (C/D) Ajustados los parámetros para el proceso de persistencia de la imagen.
- [V2.2.0][#Agregado] (D) Agregados dos botones más de color: Blanco y Amarillo con su respectiva funcionalidad y elementos estilizados.
- [V3.0.0][#Agregado] (D) Agregada la función “Deshacer” mediante el uso de pilas de imágenes, asignada a su respectivo botón estilizado.
- [V3.1.0][#Agregado] (D) Agregado un botón de información cuya función consiste en abrir un archivo que permita la visualización de datos sobre el programa.
- [V3.1.1][#Pruebas] (C/D) Elaboración de archivos de soporte, retroalimentación, actualización del repositorio del proyecto, revisión de código, creación de distribuibles y pruebas generales.
- [V3.1.2][#Arreglado] (C/D) Arreglado error de la escala de puntuaciones.
- [V3.1.3][#Removido] (C) Se crea un nuevo repositorio en Github con el código final de la aplicación.
- [V3.1.4][#Arreglado] (D) Agregado comentarios explicando la función de cada método usado en el código.

[Versión corriente]

- Capacidad de guardar archivos (persistencia), imágenes concretamente, en formato “.png”
- Retroalimentación del usuario mediante comentarios y puntuaciones.
- Capacidad de realizar persistencia textual, guardando comentarios y puntuaciones en archivos “.ser”
- Implementación de la “Internacionalización”, permitiendo alternar el idioma usado, actualmente: Español, Ingles (por defecto), Francés.
- Capacidad de análisis número: Leyendo y computando los valores de las puntuaciones hechas por el usuario.
- Capacidad de generar imágenes editadas por el usuario (y editables) mediante las diversas herramientas puestas a su disposición (Color, borrador, lápiz, deshacer)
- Capacidad de cambio del color usado para editar las imágenes.
- Capacidad de deshacer el último cambio hecho, mediante el uso de pilas de imágenes.
- Implementación de componentes gráficos personalizados, como es el caso de un “Toast-like”
- Adecuada separación Modelo-Vista: Operaciones y manejo de eventos van por separado. Utilización de clases personalizadas que actúan como “ActionListener”
- Adecuada separación en el manejo de recursos: Código y recursos se ubican en diferentes paquetes.
- Prioridad en la estética: Diseño minimalista pero funcional, basado en imágenes.

- Implementación de un sistema novedoso (sino innovador) para la medición de puntuaciones hechas a la aplicación por medio de imágenes que reflejan la calificación y el uso de un icono que represente el promedio.
- Diversidad en el uso de elementos: Desde básicos como datos primitivos, pasando por flujos E/S, elementos gráficos (predeterminados y personalizados), operadores de datos numéricos, "ActionListeners" para manejo de eventos, clases personalizadas y manejo de pilas, entre otros.
- Diseño intuitivo: Basado en imágenes representativas, sin embargo se crean textos de ayuda como soporte.
- Elementos estéticos de más: Como lo son el icono de la aplicación y el "Feel-and-Look" de los marcos (Nimbus Style)
- Formato de código: Todo el código está indentado, se ha eliminado todo el código innecesario, se especifican únicamente los elementos necesarios de importar (en vez de importar toda la librería), no se presentan errores o excepciones de compilación ni en tiempo de ejecución, probado en diversas máquinas, coherencia entre clases (nombres significativos), manejo de Camel-Case y demás reglas de escritura de código.
- Capacidad para mostrar un documento en el cual se encuentra toda la información relacionada al proyecto, entre ella: Un historial de cambios, un diagrama de clases, los mock-ups iniciales, un "Read-me", una sección de contribución y anexada la primera y segunda entrega.
- Documentación, versionado y soporte en GitHub.

[Versiones futuras]

- **#PorHacer** Establecer un método que permita imprimir solo determinada cantidad de comentarios (elegidos aleatoriamente), esto con motivo de que debido a la gran cantidad de comentarios, no es posible visualizarlos todos en una caja de dialogo.
- **#PorHacer** Mejorar el manejo de excepciones: Dada la estructura del proyecto, no se esperar errores o excepciones notables, por lo que no se le da un manejo a las excepciones, sin embargo, es necesario a futuro establecer un protocolo para impedir posibles acciones no deseadas. El actual programa no es apto de tontos.
- **#PorHacer** Manejo de sockets: Integrar a la aplicación la posibilidad de compartir datos con otras máquinas o programas. Por ejemplo, enviando reportes de errores a un servidor.

***Última actualización: 24 noviembre 2016**

Anexo: Primera Entrega

Título del proyecto: Sketch Box

Objetivos:

- Poner en práctica los temas aprendidos en el curso de POO (Programación Orientada a Objetos) durante el semestre.
- Relacionarse con el desarrollo de aplicaciones en entornos grupales.
- Conocer las bases y lineamientos para el desarrollo de software a nivel profesional.
- Implementar API's que se aprendieron de manera autodidacta mediante libros y cursos online (MOOC).

Descripción del Sistema:

El programa que se busca desarrollar es un entorno de dibujo el cual le permita al usuario crear archivos de imagen, editarlos y guardarlos en su ordenador. Además tendrá la opción de usarse en varios idiomas, manejo de formas básicas y paleta de colores. Para el desarrollo del programa se implementaran interfaces y clases que heredan, además de otros temas aprendidos durante el semestre.

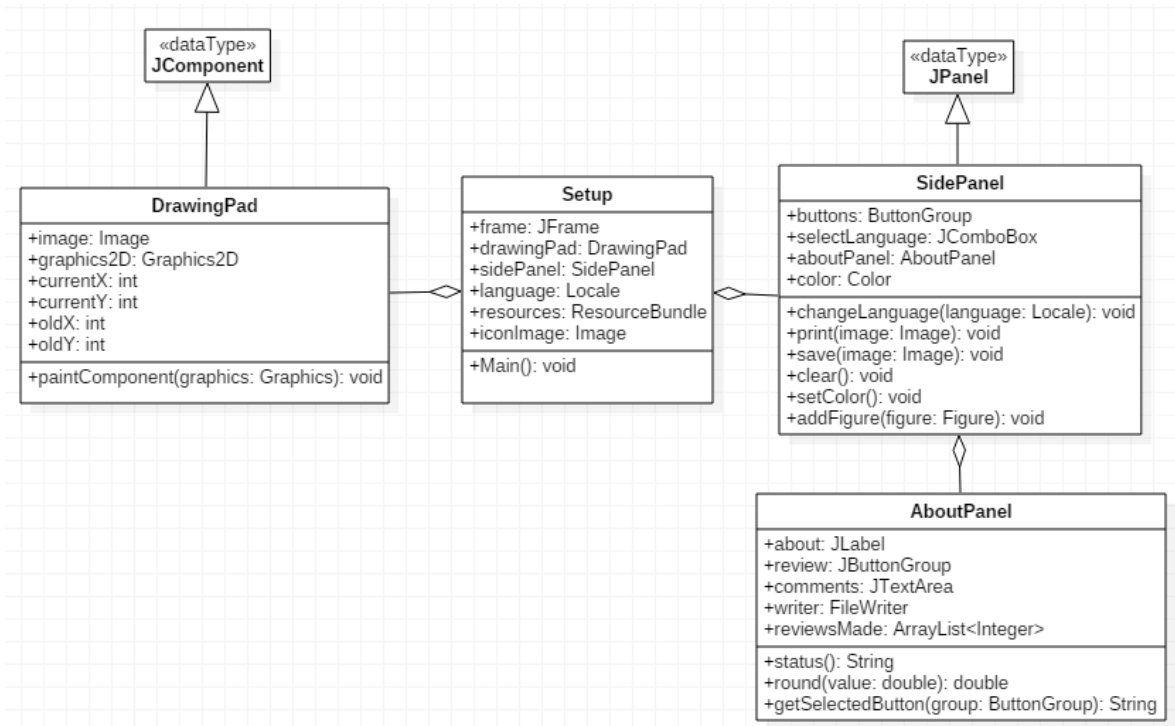
Requerimientos:

- El usuario encontrara un área de dibujo la cual abarcara cerca del 80% de la ventana y el resto corresponde a herramientas de edición y los menús.
- El entorno permitirá al usuario desarrollar y editar archivos de imagen además de guardar los cambios realizados al trabajo.
- El usuario podrá elegir el idioma de su preferencia. Inicialmente los idiomas serán: inglés, español, francés.
- El programa tendrá un panel de menú en el habrán opciones que se realizaran sobre el trabajo (Archivo, Edición, visualización, etc.) y de cada una de estas opciones se desplegara un nuevo menú.
- Dentro del panel del menú se encontrara la opción información la cual al señalarse desplegara los detalles del trabajo (Autor, Fecha de creación, cambios realizados) y un área en la cual el usuario dejara sus comentarios y calificaciones de manera interactiva.
- El programa tendrá la opción de implementar e insertar formas básicas (círculo, cuadrado, líneas rectas, etc.).
- También se podrá insertar cuadros de texto donde el usuario pueda modificar el tipo de fuente, tamaño y color de su escrito.
- Las figuras implementadas en el área de trabajo podrán ser modificadas en su tamaño.

- El usuario podrá elegir una opción lápiz con la que podrá trazar líneas libremente sobre el área de dibujo además podrá cambiarle el grosor a la línea.
- El programa implementará una opción “borrador” la cual eliminara los trazos por los cuales se sobreponga mientras se esté presionando el clic izquierdo.
- El usuario tendrá a su disposición una paleta de colores en la cual podrá el color que requiera para su trabajo.
- Las figuras podrán ser coloreadas a través de varias opciones como rellenar un área cerrada o usando un “pincel”.
- Las opciones de lápiz, borrador y color podrán ser activadas cuando el usuario realice una pulsación sobre el botón correspondiente.
- El programa contara con las opciones deshacer y rehacer.
- El programa entenderá ordenes mediante comandos de teclas tales como: Ctrl+g, Ctrl+n, y las ejecutara.
- Se implementara la opción “Zoom” para que el usuario pueda ampliar una pequeña área dentro de su trabajo y aumente su precisión al dibujar. Mientras esté usando la opción se podrá mover a otra área con ayuda de slidebar.
- El programa enviara los informes de errores al servidor, a través de internet, para ser corregidos. Los informes constan de una traza del error lanzado y una descripción del problema.
- El programa ventanas de aviso el usuario cuando se ejecute un error o cuando no logre reconocer una orden.
- El programa incluirá la opción “imprimir” con la que podrá enviar su trabajo directamente a una impresora.
- [opcional] el programa usará librerías externas para mejorar o extender las funcionalidades ya establecidas.

Anexo: Segunda Entrega

Diagrama de clases

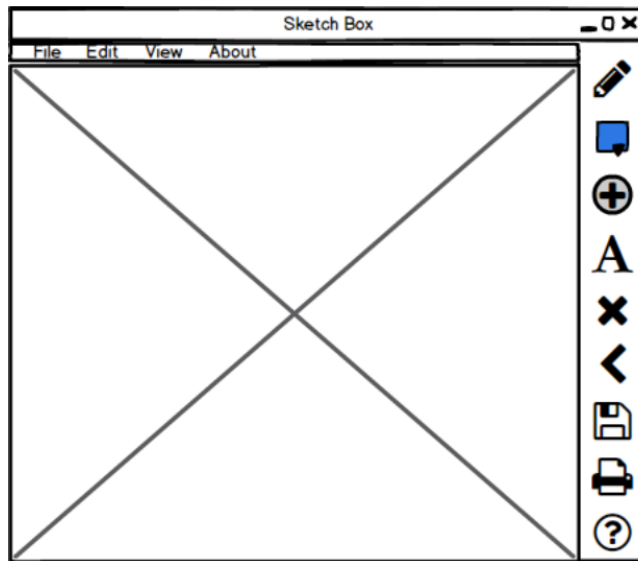


* Se suprimieron varios de los métodos llevados a cabo por los botones mediante `ActionListeners`.

Mock-ups

Inicialmente se enfocará el desarrollo del programa en las funcionalidades por lo que el diseño quedará en un segundo plano. Esto significa que las primeras versiones usaran el aspecto por defecto de las aplicaciones Java; una vez la funcionalidad del programa sea estable se pasará a modificar el diseño gráfico de la aplicación para hacerla más atractiva. Actualmente se tienen en cuenta dos alternativas: Usar el creador de interfaces graficas que hace parte de la IDE NetBeans, lo que le da un diseño un poco metálico pero moderno; la otra opción es usar una librería externa para aplicar un diseño minimalista conocido como “Material Design”.

Los lineamientos de diseño del programa es enfocarse en que el usuario sienta que el programa es amigable, dada su estructura el programa busca ser minimalista y presentar únicamente las herramientas necesarias sin restar funcionalidad. En ese mismo sentido, se apuesta por el uso de imágenes o iconos en vez de botones con textos, éstas son lo suficientemente expresivas para hacer que la experiencia sea intuitiva.

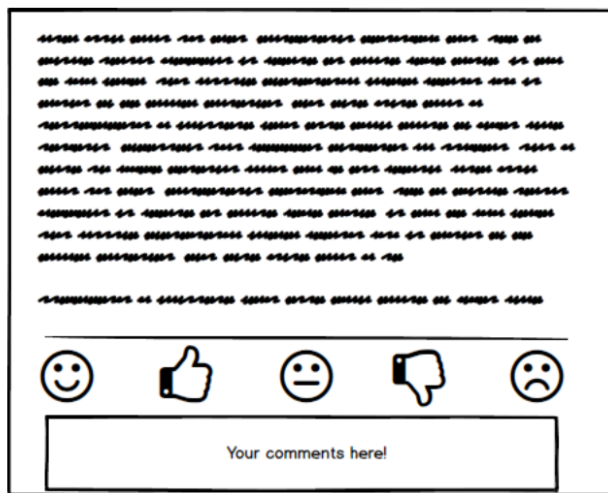


Ventana principal

- ← Permite seleccionar el color
- ← Permite agregar figuras preestablecidas
- ← Permite "limpiar" la imagen
- ← Abre una ventana del sistema para guardar e imprimir
- ← Abre el panel de ayuda

- La ventana principal consta de un panel de dibujo (El cual ocupa la mayor parte de la aplicación), un panel lateral (Ubicado en la parte derecha del recuadro, por comodidad y rápido acceso) y una barra de menú en la parte superior (En donde se encuentran algunas de las opciones).
- Por defecto, el primer botón del panel lateral (El icono de lápiz) aparece seleccionado. Esta es la herramienta de dibujo: Permite realizar trazos a mano alzada que se representan a modo de líneas.
- El segundo botón en el panel lateral (De arriba abajo) es una paleta de colores: Permite al usuario elegir el color con el que pintará.
- El tercer botón, como se indica en el diagrama, permite agregar figuras geométricas, por ejemplo: Cuadrados, círculos, rectángulos, etc...
- El botón con el icono de "A" representa una opción para agregar texto a la imagen. Por defecto se tiene seleccionada una fuente.
- El botón con el icono de "X" permite borrar todo el contenido del panel de dibujo, dejando en blanco; el botón que le sigue tiene una funcionalidad similar: Permite al usuario deshacer la última acción que llevó a cabo.
- Los iconos de "disquete" e "impresora", abren una ventana del sistema que permite guardar e imprimir respectivamente la imagen diseñada en el panel.
- Tanto el botón "About us" en la barra superior como el que se encuentra ubicado en la esquina inferior derecha (Identificado con un símbolo: ?), despliegan una ventana en la que se encuentran:
 - Información de ayuda para el usuario
 - Acerca de los creadores
 - Un panel de retroalimentación que consta de una caja de texto para opiniones y una serie de botones que permiten calificar la experiencia
 - Un mensaje de texto que muestra el promedio de calificaciones por parte de los usuarios

- Igualmente desde el botón “About us”, se puede acceder a la opción “Changelog” la cual muestra en forma de texto plano, las versiones del programa y sus cambios más importantes.
- En cuanto al panel superior: En la pestaña “File” se encuentran las opciones de “Guardar”, “Imprimir”, “Cerrar” y “Cambiar de idioma”; en la pestaña “Edit” se encuentra la opción de limpiar el panel de dibujo y deshacer la última acción; la pestaña “View” tiene como propósito proporcionar zoom y poder modificar el tamaño de la ventana, sin embargo dado el poco tiempo, aún no está decidido si hará parte del diseño final.
- La ventana funciona como cualquier otra: En la esquina superior derecha se encuentran los botones de cerrar, cambiar tamaño y minimizar. Así mismo, en la parte superior centro se encuentra el nombre de la aplicación y en la esquina superior izquierda se encuentra el icono que identifica al software.



← Ayuda al usuario y acerca de los autores

Panel de Ayuda

← Serie de iconos que permiten al usuario calificar su experiencia

↑ Un campo de texto, para las opiniones

- Adicional a las ventanas presentadas, se incluyen también cuadros de dialogo generalizados como por ejemplo ventanas del sistema o confirmación de acciones.

** NOTA: Para la creación de los “mock-ups” se usó el programa: Balsamiq Mockups 3

** NOTA: Para la creación del diagrama de clases se usó el programa StarUML v2.7.0