



Proyecto Semestral Programación II

(Pizarra para hacer clases en línea)

Nicolas Ricciardi
Matías Barriga

(Grupo 15) (Tema 4)

Diagrama UML

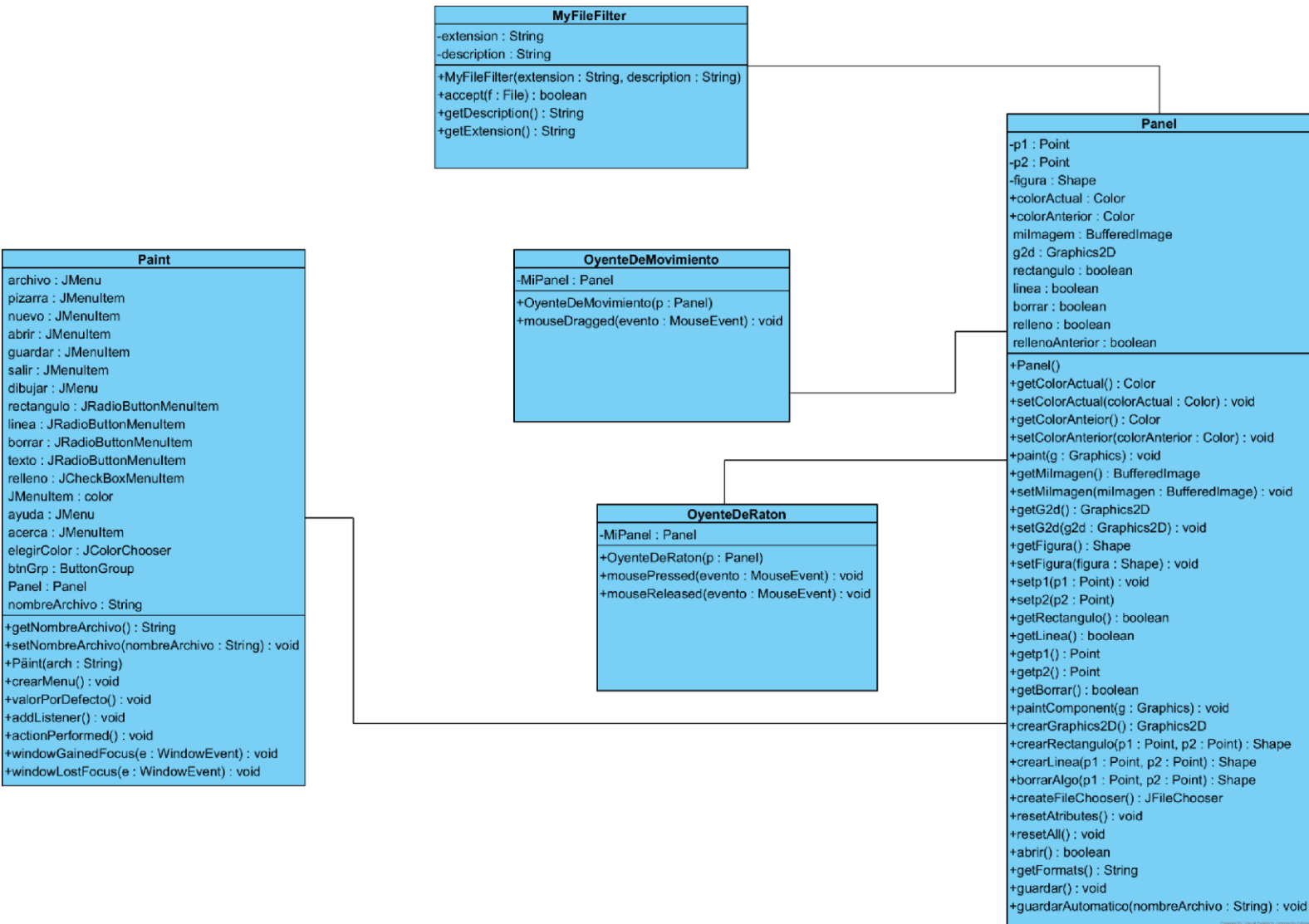
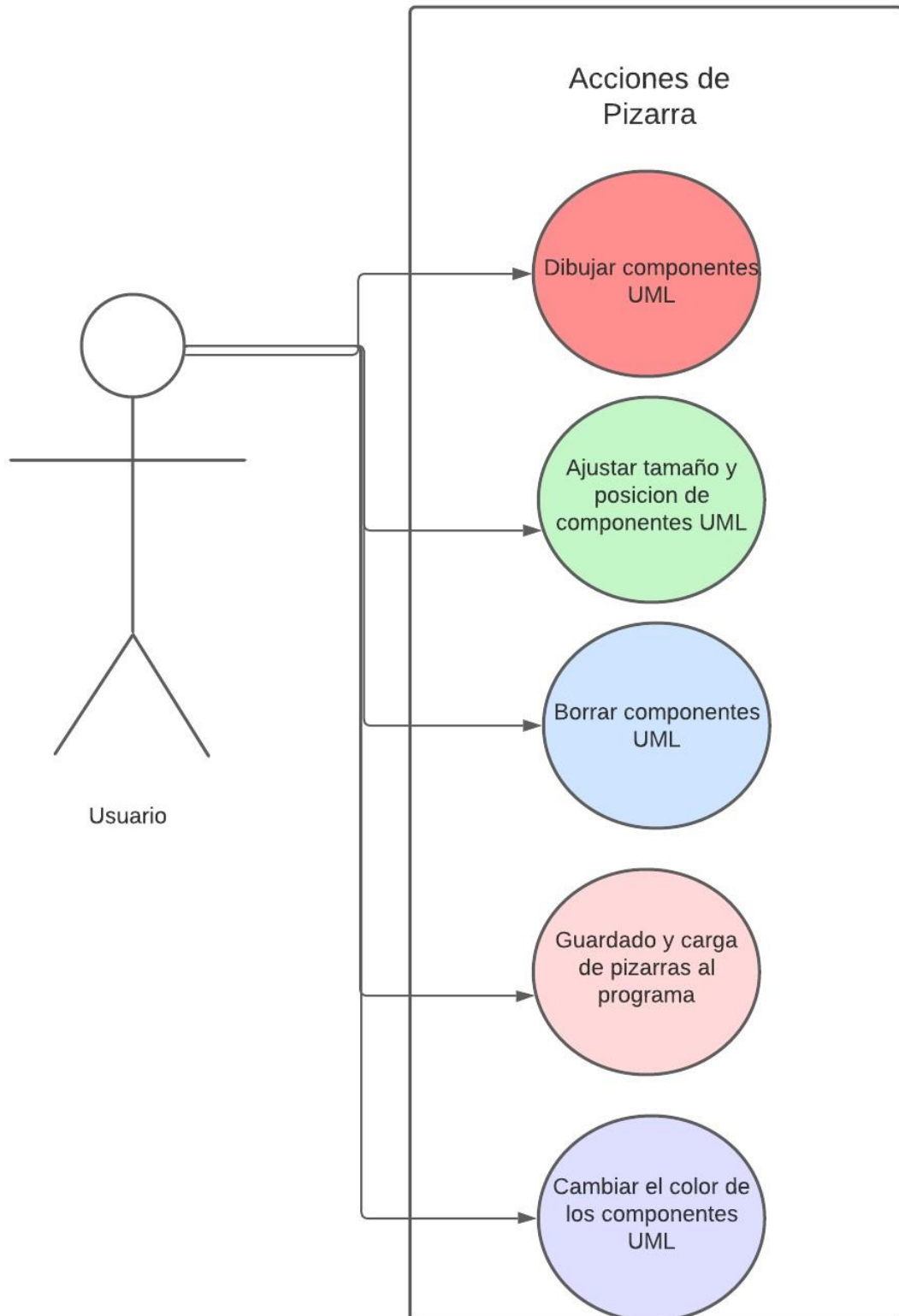


Diagrama de casos de uso

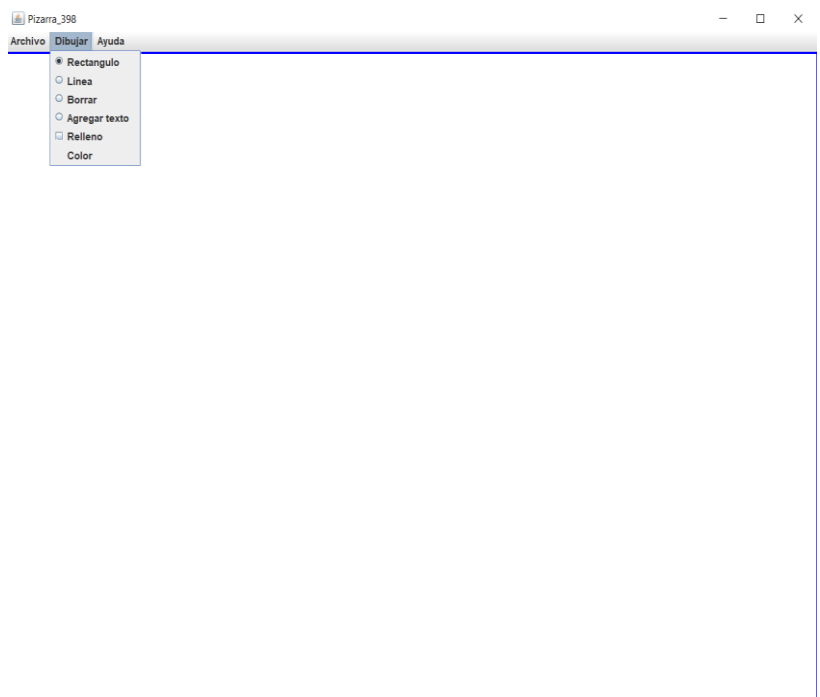
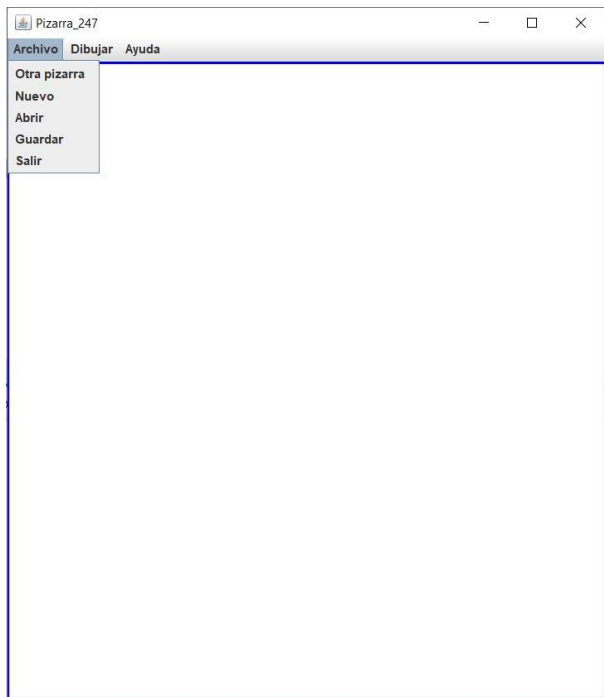


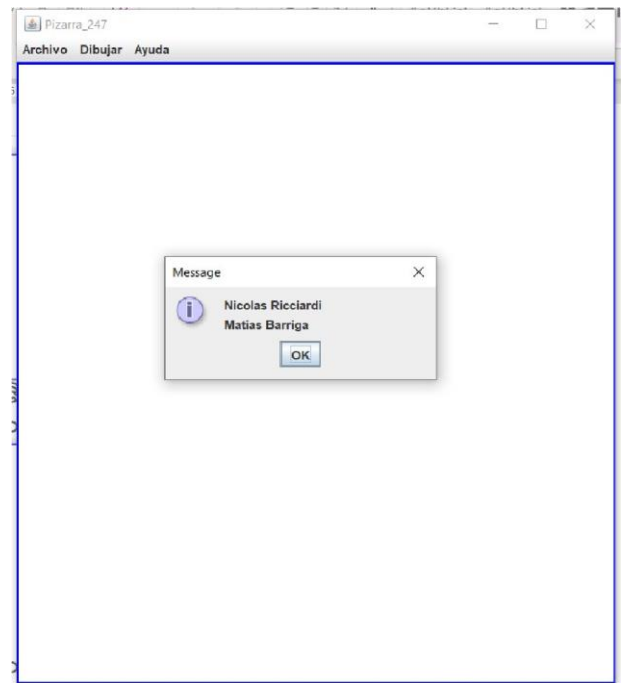
Lista y justificación de los patrones utilizados

En la imagen a continuación se aprecia el uso de Patrón Holder puesto que en primera instancia obtener las coordenadas (p1,p2) de donde va a dibujar la persona que posteriormente las recibe el método crearRectangulo y que a su vez recibe el método anterior en el setFigura para obtener que figura se va a crear, también utilizamos patron Observer donde en la clase Paint en el método actionPerformed observa a la clase MiPanel cada vez que se hace una acción en la pizarra para actualizar los componentes de la pizarra como el color y la forma del dibujo que se quiere hacer.

```
}
MiPanel.setp2(evento.getPoint());
MiPanel.setFigura(MiPanel.crearRectangulo(MiPanel.getp1(), MiPanel.getp2()));
g2d = (Graphics2D) MiPanel.getGraphics();
g2d.setXORMode(MiPanel.getBackground());
g2d.setColor(MiPanel.getColorActual());
g2d.draw(MiPanel.getFigura());
else {
    if (MiPanel.getLinea()) {
        if (MiPanel.getFigura() != null) {
            g2d = (Graphics2D) MiPanel.getGraphics();
            g2d.setXORMode(MiPanel.getBackground());
            g2d.setColor(MiPanel.getColorActual());
            g2d.draw(MiPanel.getFigura());
        }
        MiPanel.setp2(evento.getPoint());
        MiPanel.setFigura(MiPanel.crearLinea(MiPanel.getp1(), MiPanel.getp2()));
    }
}
```

Captura de pantalla de la interfaz





Una breve sección en la que explique las decisiones que han tenido que tomar durante el proyecto

-Crear la ventana principal llamada 'Paint' la cual será la pizarra de nuestro proyecto, posteriormente crear el Panel y Panel1 (Panel1 se termino quitando después por errores), creamos los métodos con la librería Graphics2d la cual nos facilito en gran manera el dibujar las figuras, además utilizamos el método JColorChooser de la librería javax.swing.JColorChooser para seleccionar el color de las figuras a dibujar, también utilizamos varios métodos de las librería javax.swing.JMenu y sus derivados para tener un menú limpio y funcional.

Decidimos crear varias carpetas para poder distribuir mejor las clases del programa, además para simplificar el numero de métodos en el panel principal y así tener un código mas ordenado y que se pueda ver mejor, también decidimos crear una opción en el menú para crear el texto, al apretar la opción de crear texto aparece una ventana donde escribes el texto que quieres escribir, para luego colocarlo donde quieres, de la misma manera en como se dibuja los rectángulos y líneas, para crear el texto creamos métodos en la Clase MiPanel.

Estas adiciones se hicieron para que el proyecto cumpla su función como pizarra creadora de graficos UML.

Una breve sección en la que se describen los problemas encontrados y que sirve de autocrítica

-El primer problema importante encontrado fue con el Panel1 puesto que se buscaba tener dos paneles (uno más grande que el otro), para hacer una interfaz más asemejada a una pizarra de verdad, pero a la hora de correr nuestro programa no se mostraba nada de nuestro menú por lo que no podíamos dibujar ni crear ningún objeto, se soluciono simplemente eliminando la clase Panel1

-El segundo problema importante fue con el procedimiento para guardar los archivos puesto que o no se creaba o no se podía asignar una ubicación correcta para el archivo, se corrigió utilizando métodos de la librería javax.swing.JFileChooser y sus excepciones correspondientes en caso de error , y en el error de la ruta del archivo, se eligió ".\\" (Para una carpeta más arriba de donde se encuentra el proyecto (arriba de la carpeta 'src')).

-Tambien hubo una falta de coordinación, la cual intentamos mejorar en la segunda revisión del proyecto.