



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

APPLICATION DEVELOPMENT FOR MOBILE
DEVICES

“DESBLOQUEO DE PANTALLA”

GRUPO: 3CV16

BOBADILLA SEGUNDO DAN ISRAEL

PROFESOR: CIFUENTES ALVAREZ ALEJANDRO SIGFRIDO

FECHA DE ENTREGA: 01/11/2021



INTRODUCCIÓN

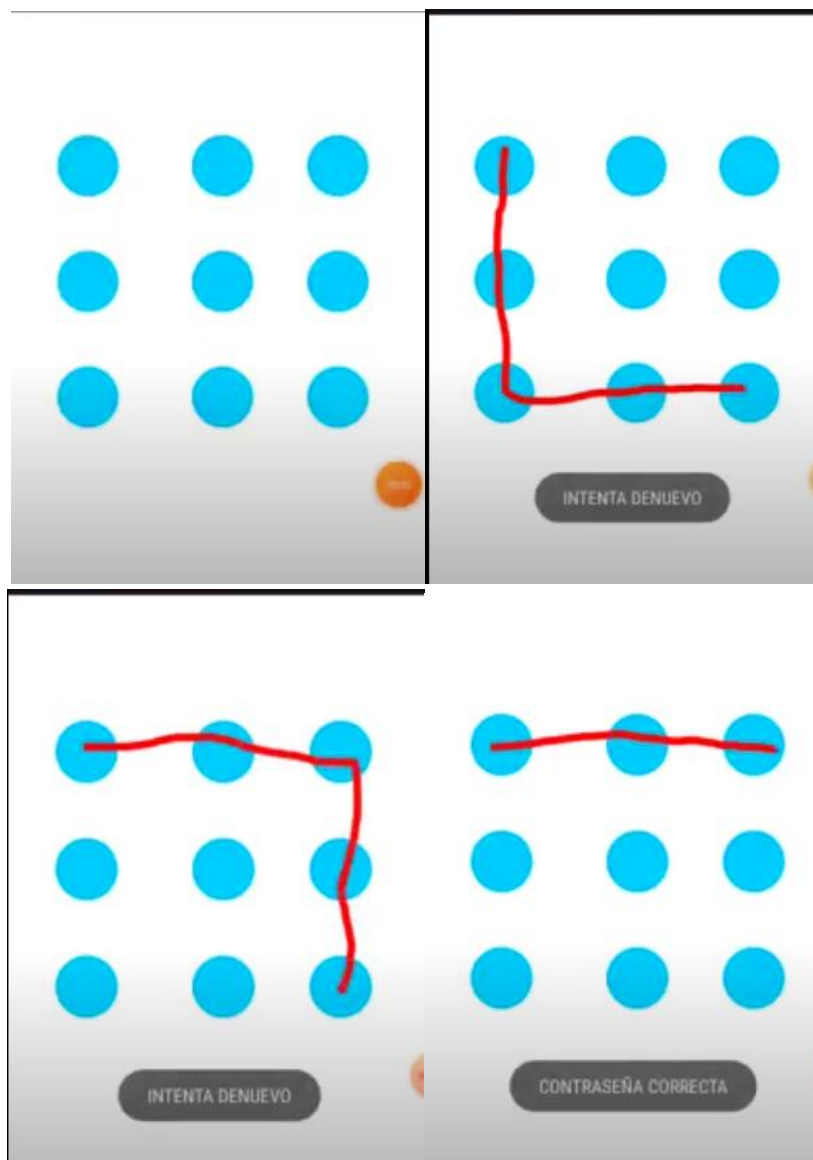
Como sabemos el tema de seguridad en un dispositivo móvil es bastante importante, por lo que es importante que un programador sepa como crear un patrón de desbloqueo utilizando las herramientas que tenga a su mano, como es el caso de canvas, sin embargo, en los últimos años estos patrones han sido descontinuado sin embargo no dejan de ser importantes.

DESARROLLO

Se explicará de manera sencilla el funcionamiento y se mostrará el código principal. Para los círculos creamos una clase circulo la cual recibe un canvas y un Paint, después creamos un contenedor dentro de cada circulo, este contenedor se implementa con un método que determina si usuario pasa el cursor encima del círculo.

Después sobreescribimos el método onTouchEvent para capturar cuando el usuario da click en la pantalla, de este método creamos dos listas para guardar los valores y que no se pierda, por ultimo llamamos al método invalidate() que se encarga de repintar las figuras del canvas.

```
public boolean onTouchEvent(MotionEvent event) {
    posX = (int) event.getX();
    posY = (int) event.getY();
    switch (event.getAction()){
        case MotionEvent.ACTION_DOWN:
            paint2 = new Paint();
            paint2.setStrokeWidth(20);
            paint2.setARGB( a: 255, r: 255, g: 0, b: 0);
            paint2.setStyle(Paint.Style.STROKE);
            paints.add(paint2);
            path = new Path();
            path.moveTo(posX,posY);
            paths.add(path);
            invalidate();
            break;
        case MotionEvent.ACTION_MOVE:
        case MotionEvent.ACTION_UP:
            int puntosHistoricos = event.getHistorySize();
            for (int i = 0; i< puntosHistoricos;i++){
                path.lineTo(event.getHistoricalX(i),event.getHistoricalY(i));
            }
            for (Circulo cir: contenido) {
                if(cir.inside(posX,posY) && cir.getBandera()){
                    cir.bandera = false;
                    resultado+= cir.numero();
                }
            }
            break;
    }
}
```



CONCLUSIÓN

Me parece interesante la práctica, sin embargo estos tipos de patrones son muy vulnerables, por lo que los nuevos patrones que son integrados ya tienen una codificación como lo es en Java puro, o utilizando la huella dactilar que es algo bastante interesante.