

# UNIVERSIDAD MAYOR, REAL Y PONTIFICIA DE SAN FRANCISCO XAVIER DE CHUQUISACA

Facultad de Ciencias de la Tecnología



**NOMBRE:** Lopez Chavez Pablo

**CARRERA:** Ingeniería en ciencias de la computación

**MATERIA:** Aplicaciones móviles

**CU:** 111-386

Sucre 2023

$$h = \sin\left(\frac{\pi}{3}\right) * l$$

h=86.6 pixeles

$l = \text{largo de lado}$

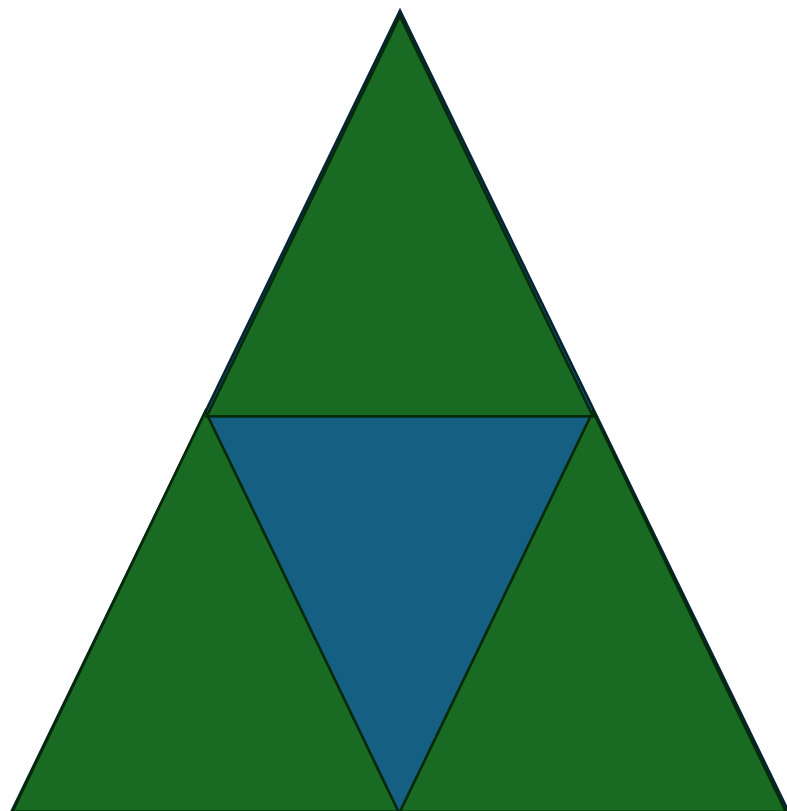
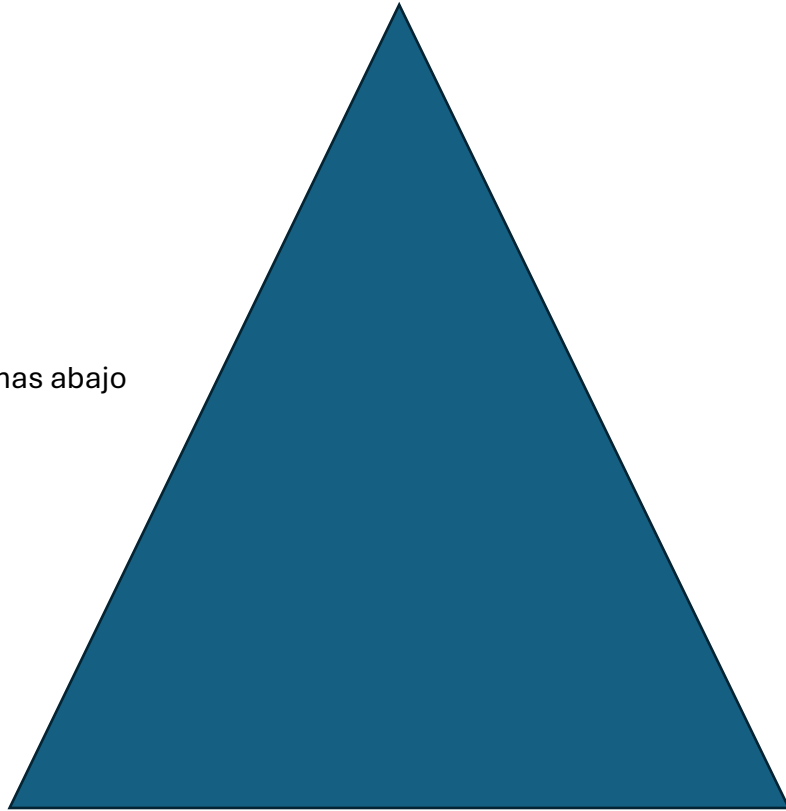
si  $l = 100$

Centro x,y

En y – es mas arriba y + es mas abajo

La creación de 4 triangulos

Y se elimina el del centro



$$halfL = \frac{l}{2}$$

halfL=50

Lado de los triangulos

$$thirdH = \frac{h}{2}$$

thirdH=43.3

Altura de los triangulos

Triangulo Izquierdo

Centro

Eje x se movera a la izquierda (halfL/2)

X= x(padre)-25

Y= y(padre)+ 43.3

halfL es el largo de los lados

Triangulo derecho

Centro

Eje x se movera a la derecha (halfL/2)

X= x(padre)+25

Y= y(padre)+ 43.3

halfL es el largo de los lados

Triangulo Superior

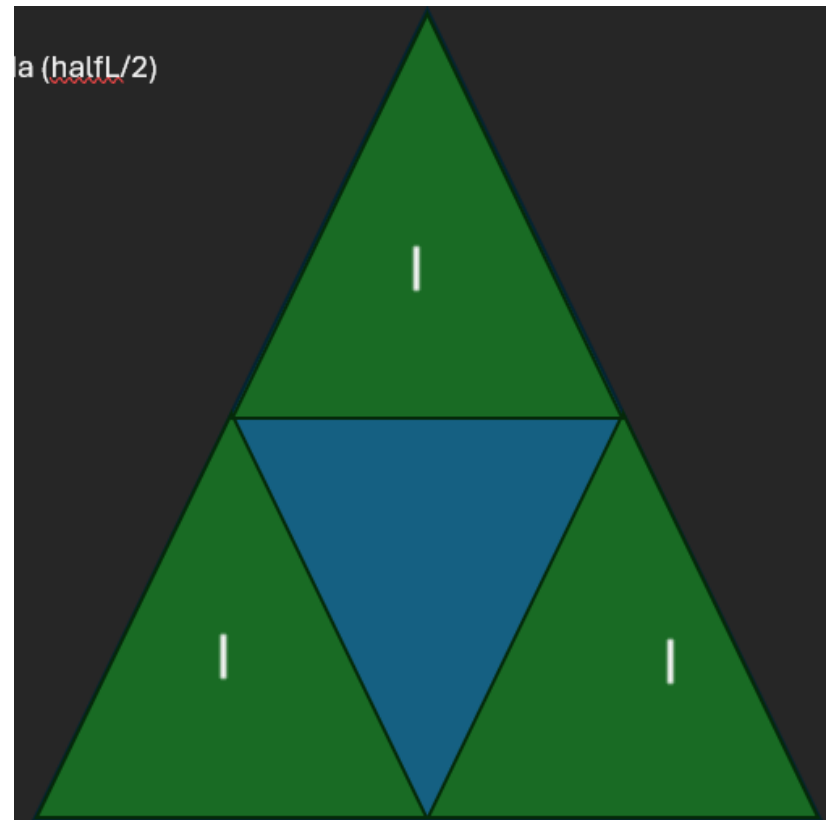
Centro

Eje x se queda en el mismo lugar

X=

Y= y(padre)- 43.3

halfL es el largo de los lados



```

public class Sierpinski extends View {
    @Override
    protected void onDraw(@NonNull Canvas canvas) {
        super.onDraw(canvas);
        canvas.drawColor(Color.BLACK);
        Paint paint = new Paint();
        paint.setColor(Color.WHITE);
        paint.setStrokeWidth(3);
        paint.setStyle(Paint.Style.STROKE);
        int ancho = canvas.getWidth();
        int alto = canvas.getHeight();
        float x = ancho / 2;
        float y = alto / 2;
        float l = Math.min(ancho, alto)*0.8f;
        float h = (float) (sin(PI / 3) * l);
        drawSierpinski(canvas, x, y - h / 2, l, 1, paint);
    }
    private void drawSierpinski(Canvas canvas, float x, float y, float l, int nivel, Paint paint) {
        if (nivel == 0) {
            drawTriangle(canvas, x, y, l, paint);
            return;
        } else {
            float h = (float) (sin(PI / 3) * l);
            float halfL = l / 2;
            float thirdH = h / 2;
            drawSierpinski(canvas, x - halfL / 2, y + thirdH, halfL, nivel - 1, paint);
            drawSierpinski(canvas, x + halfL / 2, y + thirdH, halfL, nivel - 1, paint);
            drawSierpinski(canvas, x, y - thirdH, halfL, nivel - 1, paint);
        }
    }
    private void drawTriangle(Canvas canvas, float x, float y, float l, Paint paint) {
        float h = (float) (sin(PI / 3) * l);
        canvas.drawLine(x - l / 2, y + h / 2, x + l / 2, y + h / 2, paint); // Línea base
        canvas.drawLine(x - l / 2, y + h / 2, x, y - h / 2, paint); // Lado izquierdo
        canvas.drawLine(x + l / 2, y + h / 2, x, y - h / 2, paint); // Lado derecho
    }
}

```

Comienzo desde  $x$  hasta  $x + \text{size}$

Desde  $y$  hasta  $y + \text{size}$ , se elimina el cuadrado del centro

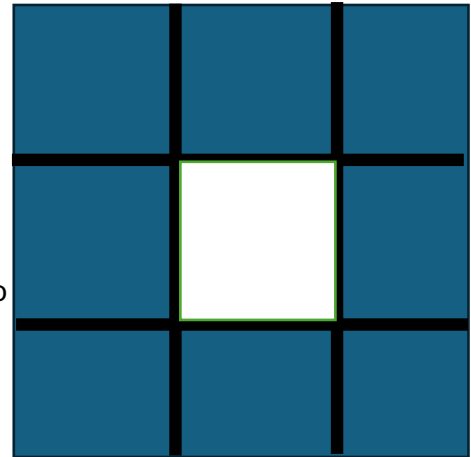
Para el siguiente nivel en cada cuadrado hijo

El tamaño de los cuadrados será  $\text{size}/3$ , va recorriendo

Por una matriz de lugar en lugar, cuando llega al cuadrado

En la posición 1,1 que es centro no lo dibuja

Para el siguiente nivel se divide el  $\text{size}$  en 9



Para calcular la grafica de en que lugar será cada

Cuadrado

la tercera parte del largo de un cuadrado

$\text{new size} = \text{size}/3$

para saber en que lugar comienza cada uno se usa la

matriz para ver en que posición va

ejemplo

para la posición 0,0

$\text{new x} = x + 0 * \text{new size}$

$\text{new y} = y + 0 * \text{new size}$

hasta

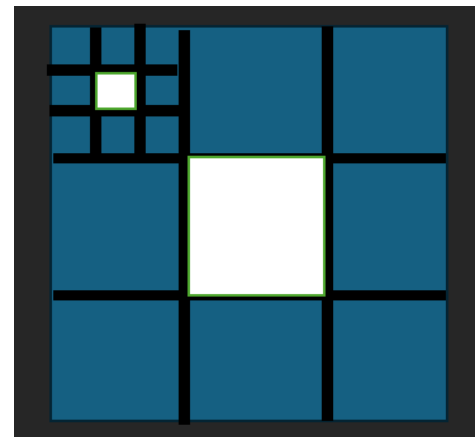
$\text{new x} + \text{new size}$

$\text{new y} + \text{new size}$

para la posición 1,0 que es la que esta centro superior

$\text{new x} = x + 1 * \text{new size}$

para  $y$  que sigue en el mismo lugar es  $\text{new y} = y + 0 * \text{new size}$  así sucesivamente



```

public class SierpinskiAlfombra extends View{
    public SierpinskiAlfombra(Context context) {
        super(context);
    }
    @Override
    protected void onDraw(@NonNull Canvas canvas) {
        super.onDraw(canvas);
        canvas.drawColor(Color.BLACK); // Fondo negro

        Paint paint = new Paint();
        paint.setColor(Color.WHITE); // Color blanco para los cuadrados
        paint.setStyle(Paint.Style.FILL); // Relleno del cuadrado
        int width = canvas.getWidth();
        int height = canvas.getHeight();
        float size = Math.min(width, height) * 0.8f;
        float x = (width - size) / 2;
        float y = (height - size) / 2;
        drawSierpinskiCarpet(canvas, x, y, size, 3, paint); // Nivel de recursión 5
    }
    private void drawSierpinskiCarpet(Canvas canvas, float x, float y, float size, int nivel, Paint
    paint) {
        if (nivel == 0) {
            canvas.drawRect(x, y, x + size, y + size, paint);
        } else {
            float newSize = size / 3;
            for (int i = 0; i < 3; i++) {
                for (int j = 0; j < 3; j++) {
                    if (i == 1 && j == 1) {
                        continue;
                    }
                    float newX = x + i * newSize;
                    float newY = y + j * newSize;
                    drawSierpinskiCarpet(canvas, newX, newY, newSize, nivel - 1, paint);
                }
            }
        }
    }
}

```

La formula seria que dentro de una matriz cada sub cuadrado tiene que tener el tamaño del cuadrado grande sobre 3 y para determinar la posición x y “y ” se usa matrices con i y j donde de va sumando entre 0 y 2 los valores de x original multiplicado por el size del cuadrado original /3 y así se saca, igual que el triangulo se elimina el del centro y se muestra el de los costados