

<b>Nombre:</b> Andrés Alejandro Cárdenas de la Cruz		<b>Matricula:</b> 2896424
<b>Nombre del curso:</b> Computación En Java	<b>Nombre del profesor:</b> Manuel Cruz Serrano	
<b>Tema 7.</b> Instrucciones de control de flujo	<b>Actividad 7.</b> Algoritmos	
<b>Fecha:</b> 04/10/2020		

## Solución Implementada

```
package codigos;
//librerias
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
/*
Nombre: Andres Cardenas
Materia: Computacion en Java
fecha: 04/10/2020
*/
public class Main extends JFrame implements ActionListener {
    //variables de los elementos graficos
    private JTextField Iterar;
    private TextArea result;
    private JButton Bprimos,Bfibonacci,reset;

    public static void main(String[] args) {
        //declara el tamaño y título de la ventana por medio de la recursividad
        Main frame = new Main();
        frame.setSize(500,350);
        frame.setTitle("Actividad 7");
        //llama al metodo de interfaz
        frame.interfaz();
        //hace visible los elementos dentro de la ventana
        frame.setVisible(true);
    }
    //metodo que establece y define los elementos graficos
    private void interfaz() {
        //establece cuando cerrar la app
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        //establece un panel y el tipo de panel
        Container window = getContentPane();
        window.setLayout(new FlowLayout());
        //Establece los elementos graficos y sus asignaciones
        JLabel titulo = new JLabel("Cuantos Numeros desea calcular ? ");
        result = new TextArea("");
        Iterar = new JTextField(30);
        Bprimos = new JButton("Sucesion de numeros Primos");
        Bfibonacci = new JButton("Sucesion de Fibonacci");
        reset = new JButton("Reset");
        //añade a los botones la accion al ser presionadas
        Bprimos.addActionListener(this);
        Bfibonacci.addActionListener(this);
        reset.addActionListener(this);
        //añade los elementos al panel
        window.add(titulo);
        window.add(Iterar);
        window.add(Bprimos);
        window.add(Bfibonacci);
        window.add(reset);
        window.add(result);
        //establece la dimension de los botones
        Bprimos.setPreferredSize(new Dimension(230,25));
    }
}
```

```

        Bfibonacci.setPreferredSize(new
Dimension(230,25));
        reset.setPreferredSize(new Dimension(230,25));
    }
    //metodo que establece la accion que se tienen que tomar con cada boton o
con los botones en general
    @Override
    public void actionPerformed(ActionEvent event) {
        funciones op = new funciones();
        int atributo = Integer.parseInt(Iterar.getText());
        if (event.getSource().equals(Bprimos)){
            result.setText(op.primos(atributo));
        }
        else if (event.getSource().equals(Bfibonacci)){
            result.setText(op.fibonacci(atributo));
        }
        else if (event.getSource().equals(reset))
            result.setText("");
    }
}

```

```

package codigos;

public class funciones {

    //metodo para sacar n cantidad de numeros primos devuelva la cadena de los
numeros concatenada
    public String primos(int serie) {
        int [] num_primos = new int[serie];
        int primo,cant_primos=0;
        for (int x = 1;cant_primos!=serie;x++){
            primo=0;
            for (int cont = 1; cont <=x; cont++) {
                if(x%cont == 0)
                    primo++;
            }
            if(primo ==2) {
                num_primos[cant_primos] = x;
                cant_primos++;
            }
        }
        return resultado(num_primos);
    }

    //metodo para sacar la secuencia de n cantidad de fibonacci devuelva la
cadena de los numeros concatenada
    public String fibonacci(int x){
        int [] sucesion_fibonacci = new int[x];
        if (x > 0)
            sucesion_fibonacci[0]=0;
        if (x > 1)
            sucesion_fibonacci[1]=1;
        if (x > 2){
            for (int pos = 2;pos<x;pos++)
                sucesion_fibonacci[pos]=sucesion_fibonacci[pos-
1]+sucesion_fibonacci[pos-2];
        }
        return resultado(sucesion_fibonacci);
    }

    //metodo que concatena el arreglo llenado por los metodos anteriores
separandolos con , y saltos de linea cada 15 iteraciones

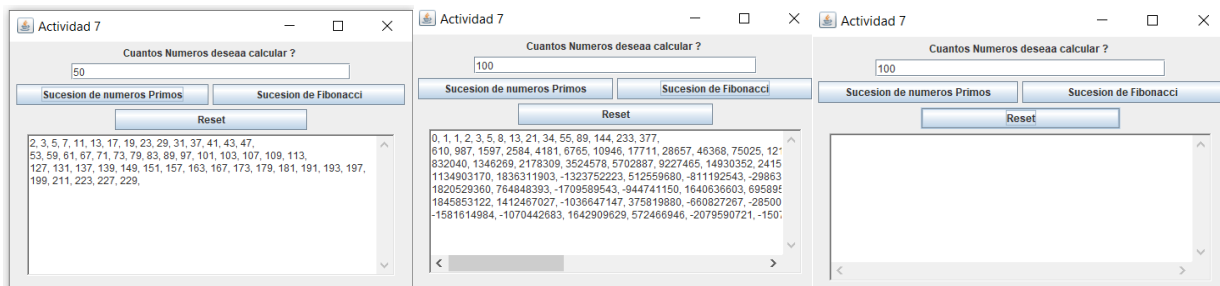
```

```

        public String resultado(int [] serie) {
            StringBuilder cadena = new StringBuilder();
            int salto = 0;
            for (int cont : serie) {
                cadena.append(cont).append(", ");
                salto++;
                if (salto == 15) {
                    cadena.append("\n");
                    salto = 0;
                }
            }
            return cadena.toString();
        }
    }

```

## Capturas



## Definiciones y Algoritmos

**Números Primos:** Los números primos son aquellos que solo son divisibles exactamente entre 2 números, que son el "1" y entre sí mismos, es utilizado para la velocidad del procesamiento en las computadoras permitiendo una mayor velocidad del procesamiento de datos complejos.

**Algoritmo Números Primos:** Se puede realizar la siguiente función para identificar si un número es primo o no:

- Se establece el número del cual se desea saber si es primo o no en este caso "x"
- Ese numero se le va a realizar la operación módulo con cada número, desde el 1 hasta en sí mismo
- Si el resultado de esta operación es igual a 0 en exactamente 2 ocasiones el numero "x" es un número primo, en dado caso que sea diferente a 2 no es un número primo

```

for (int cont = 1; cont <= x; cont++) {
    if(x%cont == 0)
        primo++;
}
if(primo == 2) {
    numeros_primos[medidor] = x; //es primo
}

```

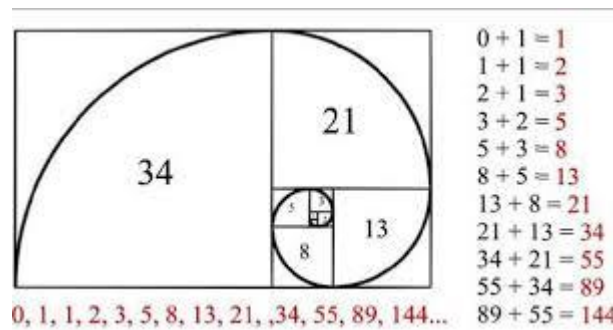
0									
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

**Sucesión de Fibonacci:** La sucesión de Fibonacci comienza con el 0 y el 1 y los siguientes números es la suma de los 2 anteriores, es utilizado en la ciencia de la computación

**Algoritmo de la sucesión de Fibonacci:** Se puede realizar la siguiente función para sacar la sucesión de Fibonacci:

- El primer número es 0
- El Segundo número es 1
- La sucesión del tercer número en Adelante se cálcula con la suma de los 2 números anteriores

```
int [] sucesion_fibonacci = new int[x];  
if (x > 0)  
    sucesion_fibonacci[0]=0;  
if (x > 1)  
    sucesion_fibonacci[1]=1;  
if (x > 2){  
    for (int pos = 2;pos<x;pos++)  
        sucesion_fibonacci[pos]=sucesion_fibonacci[pos-1]+sucesion_fibonacci[pos-2];  
}
```



<https://github.com/alex-crdnz/actividades.git>