

# Práctica 1: Recursividad

José Luis Quiroz Fabián

Restricciones;

1. Celulares frente a ustedes pantalla con abajo de la mesa.
2. No tener audífonos.
3. Abrir su navegador en la página del API de Java.
4. No tener abierta otra página sin pre-autorización.

Usando **nano**, **vi** o **vim** para crear una clase llamada **Recursividad**. La Figura 1 muestra un ejemplo del editor. Observe que puede escribir directamente su código y para guardarlo tiene que presionar **Ctrl+O** y para salir **Ctrl+X**.



```
UW PICO 5.09 File: Recursividad.java
public class Recuesividad{

    public static void main(String[] args){

        System.out.println("Hola Mundo");

    }

}
```

Get Help WriteOut Read File Prev Pg Cut Text Cur Pos  
Exit Justify Where is Next Pg UnCut Text To Spell

Figure 1: Editor nano

La Figura 2 muestra el proceso de ejecución y compilación con Java. El compilador de Java es **javac**, el cual genera un archivo **\*.class** que es el binario para la máquina virtual **java**.

```

jluisqf@MacBook-Pro-de-UAM ~ % nano Recursividad.java
jluisqf@MacBook-Pro-de-UAM ~ % javac Recursividad.java
jluisqf@MacBook-Pro-de-UAM ~ % ls
AndroidStudioProjects  Downloads      Music          Recursividad.class
Desktop               Library        Pictures       Recursividad.java
Documents             Movies         Public
jluisqf@MacBook-Pro-de-UAM ~ % java Recursividad
Hola Mundo
jluisqf@MacBook-Pro-de-UAM ~ % █

```

Figure 2: Compilación y ejecución en Java

En la clase **Recursividad** para cada problema implemente al menos un método.

1. ~~Calcular el producto de dos números positivos por medio de sumas.~~
2. Calcular el  $n$ -ésimo número de Xiomi. Los primeros  $n$  números de Xiomi
 
$$x_1 = 1$$

$$x_2 = 1$$

$$x_3 = 1$$

$$x_n = x_{n-2} + x_{n-3} \text{ con } n \geq 4$$
 En su implementación no calcule más de una vez un número de Xiomi que ya calculo, utilice un arreglo para ir almacenando los números de Xiomi.
3. Indicar si todos los símbolos de una cadena se encuentran en otra cadena.
4. Invertir una cadena, por ejemplo, la cadena "El gato esta corriendo" se debe invertir como "odneirrocatse otagiE". En este ejercicio su método debe recibir un String.
5. StdDraw.java es una biblioteca para dibujar. Genera un panel en el cual se puede dibujar. La puede compilar por medio de **javac -Xlint:none StdDraw.java** y ejecutar con **java StdDraw.java**. Por medio de StdDraw.java implementar el fractal de la Figura 3 para un valor de N ingresado por parámetro. En este ejemplo el valor N=3.

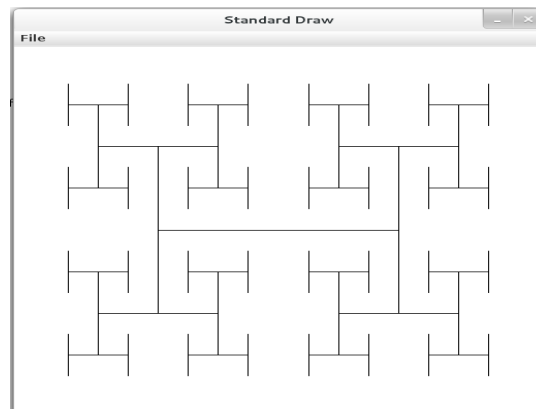


Figure 3: Fractal H

Tome como base el siguiente código.

---

```

/*****
 * Compilacion: javac StdDraw.java Htree.java
 * Ejecucion: java Htree N
 *
 *****/

public class Htree {

    public static void dibujar() {

        StdDraw.line(0.0, 0.0, 0.5, 0.5);
        StdDraw.line(0.0, 1.0, 1.0, 0.0);
    }

    public static void main(String[] args) {
        dibujar();
    }
}

```

---

6. Encontrar cuántos caminos hay entre una coordenada entera del cuadrante positivo al origen. Cualquier camino que pase por los negativos no es válido.