



Práctica 4

[Problema 1](#)

[Problema 2](#)

[Problema 3](#)

[Apartado A](#)

[Apartado B](#)

Problema 1

```
# Parámetros
simulaciones = 10000
x = runif(simulaciones, 0, 1)
y = runif(simulaciones, 0, 1)
simulacion = data.frame(cbind(x, y))
simulacion$dist = sqrt(x**2 + y**2)
simulacion$ok = as.factor((simulacion$dist >= 1)*1)

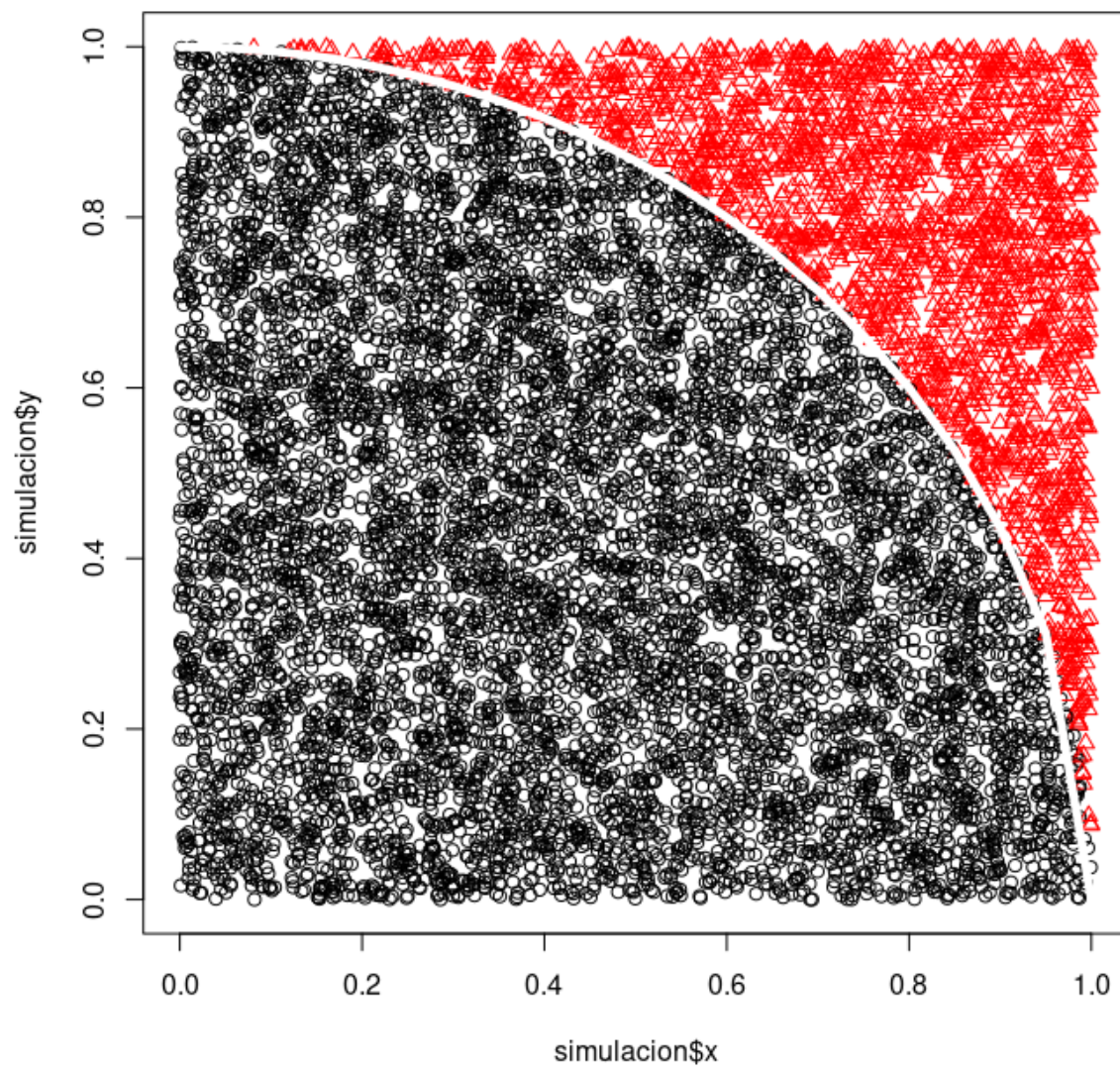
# Gráfico
plot(simulacion$x, simulacion$y, pch=c(1, 2)[simulacion$ok], col=c(1, 2)[simulacion$ok])
x = seq(0, 1, by = 0.05)
y = sqrt(1-x**2)
circunferencia = data.frame(cbind(x, y))
lines(circunferencia, col="white", lwd=5)

# Contar puntos de un solo cuadrante
A = nrow(subset(simulacion, ok=="0")) / nrow(simulacion)

# Multiplicar por 4 para obtener el total
A = 4*A
```

```
> A  
[1] 3.1409
```

Cambiando el valor de la variable `simulaciones` el valor `A` se aproximará al de π .



Gráfica que representa la simulación

Esos puntos corresponden a los que caen dentro de la circunferencia con los que poder calcular el valor de `A`, lo más cercano posible a π .

Problema 2

```
import java.util.Scanner;

public class Problema2 {

    public static void main(String[] args) {
        int cantidad, semilla;

        Scanner sc = new Scanner(System.in);

        System.out.print("Semilla (int): ");
        semilla = sc.nextInt();

        System.out.print("Números: ");
        cantidad = sc.nextInt();

        sc.close();

        for (int i = 0; i < cantidad; i++) {
            semilla = seleccion((int) Math.pow(semilla, 2), String.valueOf(semilla).length());

            System.out.println(semilla);
        }
    }

    private static int seleccion(int semilla, int tam) {
        String cadenaNueva = "", cadena;
        int tamano, posicion, c = 0;

        cadena = String.valueOf(semilla);
        tamano = cadena.length();
        posicion = tamano % tam;

        if (posicion == 0) {
            posicion = tamano / tam;

            for (int i = posicion; i < tamano; i++) {
                if (c != tam) {
                    cadenaNueva += cadena.charAt(i);
                    c++;
                } else {
                    break;
                }
            }
        } else {
            for (int i = posicion-1; i < tamano; i++) {
                if (c != tam) {
                    cadenaNueva += cadena.charAt(i);
                    c++;
                } else {
                    break;
                }
            }
        }

        return Integer.parseInt(cadenaNueva);
    }
}
```

```
}  
}
```

```
Semilla (int): 5555  
Números: 15  
8580  
6164  
9948  
9627  
6791  
1176  
8297  
8402  
5936  
2360  
6960  
4416  
5010  
1001  
200
```

Problema 3

Apartado A

```
import java.util.Scanner;  
  
public class Problema3 {  
  
    public static void main(String[] args) {  
        int x, a, c, m;  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Semilla (x): ");  
        x = sc.nextInt();  
  
        System.out.print("Constante multiplicativa (a): ");  
        a = sc.nextInt();  
  
        System.out.print("Constante aditiva (c): ");  
        c = sc.nextInt();  
  
        System.out.print("Módulo (m): ");  
        m = sc.nextInt();  
  
        sc.close();  
  
        congruenciaMixto(x, a, c, m);  
    }  
  
    private static void congruenciaMixto(int x, int a, int c, int m) {  
        int periodo = 0, seguir = 0;
```

```

        while (seguir != x) {
            if (periodo == 0) {
                seguir = x;
            }

            x = (a*c + x) % m;

            System.out.println(x);

            periodo++;
        }

        if (periodo == m) {
            System.out.println("Período completo: " + periodo);
        } else {
            System.out.println("Período incompleto: " + periodo);
        }
    }
}

```

```

Semilla                (x): 4
Constante multiplicativa (a): 5
Constante aditiva      (c): 7
Módulo                 (m): 8
7
2
5
0
3
6
1
4
Período completo: 8

```

Apartado B

```

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Problema3 {

    public static void main(String[] args) {
        int x, a, c, m;

        Scanner sc = new Scanner(System.in);

        System.out.print("Semilla                (x): ");
        x = sc.nextInt();

        System.out.print("Constante multiplicativa (a): ");
        a = sc.nextInt();

        System.out.print("Constante aditiva      (c): ");
        c = sc.nextInt();
    }
}

```

```

        System.out.print("Módulo (m): ");
        m = sc.nextInt();

        sc.close();

        congruenciaMixto(x, a, c, m);
    }

    private static void congruenciaMixto(int x, int a, int c, int m) {
        List<Integer> valores = new ArrayList<>();
        int periodo = 0, seguir = 0;

        while (seguir != x) {
            if (periodo == 0) {
                seguir = x;
            }

            x = (a*c + x) % m;

            System.out.println(x);

            valores.add(x);
            periodo++;
        }

        if (periodo == m) {
            System.out.println("Período completo: " + periodo);
        } else {
            System.out.println("Período incompleto: " + periodo);
        }

        System.out.println("Ciclos: " + ciclos(valores));
    }

    public static List<Integer> ciclos(List<Integer> lista) {
        List<Integer> ciclo = new ArrayList<>();
        boolean terminar = false;
        int contador = 0;

        while (contador < lista.size() && !terminar) {
            for (int i = 0; i < lista.size(); i++) {
                if (contador == lista.get(i)) {
                    ciclo.add(i);
                    terminar = true;
                }
            }

            contador++;
        }

        return ciclo;
    }
}

```