# PRACTICA-1-Analizador.pdf

**CreatorBeastGD**

**Análisis y diseño de algoritmos**

**2º Grado en Ingeniería del Software**

**Escuela Técnica Superior de Ingeniería Informática**
**Universidad de Málaga**


Javier Molina Colmenero

Análisis y Diseño de Algoritmos – Práctica 1 (Analizador)

**Descripción del problema**

El programa Analizador.java consta de un algoritmo que es capaz de identificar la complejidad experimental de un algoritmo proveniente de una clase Algoritmo.class, a partir de hacer una media de los tiempos de ejecución de dicho algoritmo en base a unos valores definidos por el programa, y teniendo este que diferenciar entre 8 complejidades distintas (1, N, N^2, N^3, 2^N, n*log(N), log(N), y N!).

**Explicación del analizador**

Mi algoritmo para analizar complejidades experimentales está dividida en dos partes:

- Una función Analizar (long n), que se encarga de ejecutar la función suministrada por Algoritmo.class, y recopilará una media del tiempo empleado en realizar distintas pruebas del algoritmo para un valor.

```
public static double Analizar (long n) {
    double media = 0;
    for (int i = 0 ; i < 6 ; i++) {
        t.reiniciar();
        t.iniciar();
        Algoritmo.f(n);
        t.parar();
        if (i != 0) {
            media += t.tiempoPasado();
        }
    }
    return media / (double) 5;
}
```

- La función main (String arg[]), encargada de calcular el ratio formado por el análisis de dos valores distintos, y compararlos con una serie de rango de valores que igualará dicho algoritmo con una complejidad experimental.

```
public static void main(String arg[]) {
    long n1 = 10;
    long n2 = 20;
    double ratio = Analizar(n2)/Analizar(n1);
    if (ratio > 900) {
        if (ratio < 30000) {
            System.out.println("2N"); // Aprox 1024
        } else {
            System.out.println("NF"); // Factorial
        }
    } else {
        {
            if (6.0 <= ratio && ratio < 10.0) { // Aprox 8
                System.out.println("N3");
            } else if (3.0 <= ratio && ratio < 6.0) { // Aprox 4
                System.out.println("N2");
            } else if (1.95 < ratio && ratio < 3.0) { // Aprox 2.6
                System.out.println("NLOGN");
            } else {
                n1 = 1000;
                n2 = 100000;
                ratio = Analizar(n2)/Analizar(n1);
                if (ratio > 20) {
                    System.out.println("N");
                } else if (ratio < 1.05) {
                    System.out.println("1");
                } else {
                    System.out.println("LOGN");
                }
            }
        }
    }
}
```

**Explicación de la función Analizar (long n):**

Para hacer el análisis de un valor, se ha tomado una variable tipo double para hacer la media de los valores obtenidos, puesto que abarca una gran cantidad de valores, y hay menos riesgo para que el valor devuelto por la función se salga del límite permitido por la variable y sea negativo.

Por otra parte, se ha inicializado una variable tipo Temporizador, t, para poder contar el tiempo que tarda en hacer la función del algoritmo una vez.

```
private static Temporizador t = new Temporizador();
```

Para hacer el análisis, se ha decidido realizar el algoritmo 6 veces para un mismo valor, y para cada uno de ellos se contará el tiempo que se ha tardado en realizar el algoritmo. Se destaca el hecho de que el primer análisis de los 6 no se cuenta para la media, esto se debe a que el primer tiempo del primer valor siempre será mucho más alto que el resto, por ser el primer algoritmo que se ejecuta en todo el programa.

```
if (i != 0) {
    media += t.tiempoPasado();
}
```

Se devolverá una media de los tiempos obtenidos al ejecutar el algoritmo las 5 últimas veces.

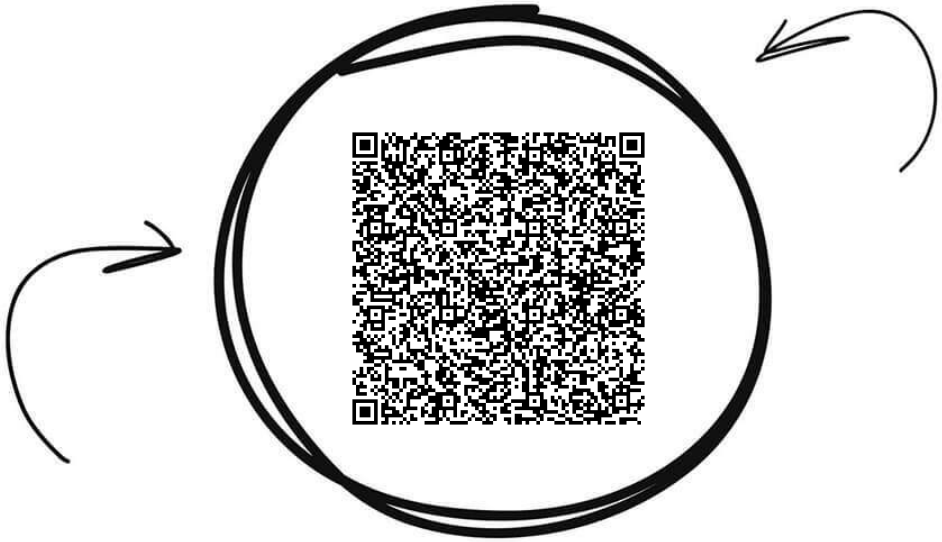
**Explicación de la función main (String arg[]):**

# 5€ DE BIENVENIDA

Con esta promo, te llevas **5€** por
tu cara bonita al subir **3 apuntes**
a Wuolah Wuolitah

WUOLAH

# Análisis y diseño de algoritmos



## Banco de apuntes de la

WUOLAH

**Comparte estos flyers en tu clase y consigue más dinero y recompensas**

1. Imprime esta hoja
2. Recorta por la mitad
3. Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes
4. Llévate dinero por cada descarga de los documentos descargados a través de tu QR

Para hacer el análisis del algoritmo, se ha decidido hacer una separación de tres grupos, para lograr una mayor exactitud en el análisis:

- La fase principal cuenta con el cálculo de la división (ratio) entre el análisis del algoritmo para 20, y el análisis del algoritmo para 10 En primer lugar comparará el algoritmo con las complejidades 2^N y N factorial, puesto que serán las complejidades que poseerán el mayor ratio. Compara si el ratio está cerca de 1024 (desde 900), y si es correcto, saldrá que la complejidad del algoritmo es 2^N. Si es mucho mayor será N factorial.

  En caso de que ninguno se cumpla, se pasa a la siguiente fase.

```
long n1 = 10;
long n2 = 20;
double ratio = Analizar(n2)/Analizar(n1);
if (ratio > 900) {
    if (ratio < 30000) {
        System.out.println("2N"); // Aprox 1024
    } else {
        System.out.println("NF"); // Factorial
    }
} else {
```

- En la segunda fase del analizador, se verá si el algoritmo, comprobado con los mismos valores que los anteriores, tiene una complejidad de N^3 (El ratio será entre 6 y 10, valores próximos a 8), N^2 (Entre 3 y 6, valores próximos a 4), y a N*Log(N) (ratio comprendido entre 1.95 y 3, valor cercano a 2.6, resultado de dividir 20log(20)/10log(10)).

  Si todavía no se ha cumplido el análisis, se pasa a la fase final.

```
} else {
    {
        if (6.0 <= ratio && ratio < 10.0) { // Aprox 8
            System.out.println("N3");
        } else if (3.0 <= ratio && ratio < 6.0) { // Aprox 4
            System.out.println("N2");
        } else if (1.95 < ratio && ratio < 3.0) { // Aprox 2.6
            System.out.println("NLOGN");
        } else {
```

- En la última fase del algoritmo, se vuelve a hacer un análisis con dos valores distintos, esta vez serán 100000 y 1000, que serán valores que permitirán reconocer fácilmente las complejidades de N, 1, y LogN.

  Puesto que los resultados de dividir 100000/1000 y log(100000)/log(1000) son muy distintos (100 y 1.6), si en el ratio obtenemos un valor muy alto (superior a 20, puesto que el valor obtenido por el análisis de LogN no puede ser tan alto), entonces la complejidad será de N. En el caso de que el ratio sea muy cercano a 1, su complejidad será 1. Para otro caso, se concluye que su complejidad corresponde a la de un logaritmo.


No sé en qué momento nos pareció buena idea lanzar nuestro podcast para estudiantes en verano.

escúchatelo o algo, así le digo a mi jefe que ha sido un éxito

```
} else {
    n1 = 1000;
    n2 = 100000;
    ratio = Analizar(n2)/Analizar(n1);
    if (ratio > 20) {
        System.out.println("N");
    } else if (ratio < 1.05) {
        System.out.println("1");
    } else {
        System.out.println("LOGN");
    }
}
```

```
public class Analizador {
	private static Temporizador t = new Temporizador();

	public static void main(String arg[]) {
		long n1 = 10;
		long n2 = 20;
		double ratio = Analizar(n2)/Analizar(n1);
		if (ratio > 900) {
			if (ratio < 30000) {
				System.out.println("2N"); // Aprox 1024
			} else {
				System.out.println("NF"); // Factorial
			}
		} else {
			{
				if (6.0 <= ratio && ratio < 10.0) { // Aprox 8
					System.out.println("N3");
				} else if (3.0 <= ratio && ratio < 6.0) { //
Aprox 4
					System.out.println("N2");
				} else if (1.95 < ratio && ratio < 3.0) { //
Aprox 2.6
					System.out.println("NLOGN");
				} else {
					n1 = 1000;
					n2 = 100000;
					ratio = Analizar(n2)/Analizar(n1);
					if (ratio > 20) {
						System.out.println("N");
					} else if (ratio < 1.05) {
						System.out.println("1");
					} else {
						System.out.println("LOGN");
					}
				}
			}
		}
	}

	public static double Analizar (long n) {
		double media = 0;
		for (int i = 0 ; i < 6 ; i++) {
			t.reiniciar();
			t.iniciar();
			Algoritmo.f(n);
			t.parar();
			if (i != 0) {
				media += t.tiempoPasado();
			}
		}
		return media / (double) 5;
	}
}
```