

# Guía de ejercicios: Análisis del tiempo de ejecución

## Consignas

### Ejercicio 1

Calcular el tiempo de ejecución de cada ítem:

- a) 

```
j=k-2;
i=j*12;
printf("%d\n",i);
```
- b) 

```
for (i=0;i<n;i++)
printf("%d\n",i);
```
- c) 

```
for (i=0;i<10;i++)
for (j=0;j<n;j++)
printf("%d\n",i+j);
```
- d) 

```
for (i=0;i<100;i++)
q=i+j;
```
- e) 

```
for (i=0;i<n;i++)
for (j=0;j<n;j++)
for (k=0;k<n;k++) {
    q=i+j*k;
    printf("%d\n",q);
}
```
- f) 

```
for (i=1;i<=n;i++)
{
    printf("%d\n",i);
    for (j=n;j>0;j--)
        for (k=0;k<n-10;k++)
            q=i+2j*k;
}
```

```
g) for (i=0;i<n;i++)
    for (j=0;j<n;j++)
        acum+=A[i][j];
k=0;
while (k<n) {
    acum-=k;
    k++;
}
```

```
h) i=1;
while (i*i<n)
    i++;
```

```
i) i=n;
while (i>1) {
    printf("%d",i);
    i/=2;
}
```

## Ejercicio 2

Desarrollar un algoritmo que calcule la sumatoria de los primeros n números naturales.  
Calcular su tiempo de ejecución.

## Ejercicio 3

Desarrollar un algoritmo iterativo que calcule la división entre dos números enteros por restas sucesivas. Calcular su tiempo de ejecución.

## Ejercicio 4

Desarrollar un algoritmo para calcular la media de un vector de n elementos. Luego, calcular su tiempo de ejecución.

## Ejercicio 5

Desarrollar un algoritmo para calcular el elemento máximo de un vector, y luego calcular su tiempo de ejecución.