

# MEDICIÓN DE TIEMPOS DE EJECUCIÓN

Algoritmos y Estructuras de Datos I

- ▶ ¿Cómo medimos el tiempo que tarda en ejecutar un algoritmo?

- ▶ ¿Cómo medimos el tiempo que tarda en ejecutar un algoritmo?
- ▶ Vamos a utilizar:
  - ▶ **clock()**: tiempo aproximado de CPU que transcurrió desde que nuestro programa fue iniciado, expresado en ticks de reloj.

- ▶ ¿Cómo medimos el tiempo que tarda en ejecutar un algoritmo?
- ▶ Vamos a utilizar:
  - ▶ **clock()**: tiempo aproximado de CPU que transcurrió desde que nuestro programa fue iniciado, expresado en ticks de reloj.
  - ▶ **CLOCKS\_PER\_SEC**: representa el número de ticks de reloj por segundo.

## EJEMPLO

Queremos saber cuanto tiempo tarda en ejecutar la siguiente función

---

```
1 int indicePrimeraAparicion(vector<int>& v, int elem){
2     int res = -1;
3     for(int i = 0; i < v.size(); i++){
4         if(v[i] == elem){
5             res = i;
6         }
7     }
8     return res;
9 }
```

---

¿Cómo podemos hacer?

# MEDICIÓN DE TIEMPO CON CLOCK

---

```
1
2 vector<int> v = {1, 2, 3, 4, 5, 6}
3
4 double t0 = clock();
5 int indice = indicePrimeraAparicion(v, 1);
6 double t1 = clock();
7
8 double tiempo = (double(t1-t0)/CLOCKS_PER_SEC);
```

---

¿Cómo guardamos en un archivo cuanto tiempo tarda nuestro programa para diferentes tamaños de vectores?

¿Cómo guardamos en un archivo cuanto tiempo tarda nuestro programa para diferentes tamaños de vectores?

## FORMATO

n	tiempo
0	0.001
1000	0.006
2000	0.011
3000	0.016
4000	0.021
5000	0.026
6000	0.032
7000	0.037
8000	0.041
9000	0.047



## GUARDAR TIEMPOS CONTINUACIÓN

```
1
2 int n = 0; int hasta = 10000; int paso = 1000;
3 ofstream fout;
4 fout.open("datos.csv");
5
6 fout << "n\t" << "tiempo" <<endl;
7 fout << "n\t" << "tiempo" <<endl;
8
9 while(n < hasta){
10     vector<int> v = construir_vector(n, "asc");
11
12     double t0=clock();
13     int indice = indicePrimeraAparicion(v, 1);
14     double t1 = clock();
15
16     double tiempo = (double(t1-t0)/CLOCKS_PER_SEC);
17
18     fout << n << "\t" << tiempo << endl;
19
20     n +=paso;
21 }
22 fout.close()
```

¿Cómo graficamos los tiempos en función del tamaño de la entrada?

¿Cómo graficamos los tiempos en función del tamaño de la entrada?

```
$python graficar.py --help
```

¿Cómo graficamos los tiempos en función del tamaño de la entrada?

```
$python graficar.py --help
```

```
Usage: graficar.py [OPTIONS]
```

Options:

<code>-i, --input PATH</code>	<code>[required]</code>
<code>-o, --salida TEXT</code>	
<code>--guia [sqrt logn n n2 n3 nlogn]</code>	
<code>--help</code>	Show this message and exit.

## VOLVEMOS AL EJEMPLO

```
$python graficar.py -i datos.csv -o lineal.png --guia n
```

# GRÁFICO

