

# ANÁLISIS Y DISEÑO DE ALGORITMOS

---

## COMPLEJIDAD TEMPORAL: ENFOQUE EMPÍRICO (I)

### Práctica 1 de laboratorio

Fecha de realización y entrega: del 5 al 10 de febrero de 2015  
(en la sesión de laboratorio que corresponde a cada alumno)

---

A partir de las funciones `burbuja()` y `tiempo()`, disponibles en la sección de materiales de prácticas del *Campus Virtual*, realiza un análisis empírico del tiempo promedio que el algoritmo de ordenación *Burbuja* requiere para realizar su cometido.

Para ello, hay que escribir una función `main()` que genere vectores con contenido aleatorio<sup>1</sup> y de distintas tallas para que el algoritmo *Burbuja* los ordene. El tamaño de los vectores debe ser suficientemente grande para que el análisis no se desvirtúe por la influencia de factores ajenos a la propia ordenación, en este sentido, las potencias exactas de 2 desde 8 hasta 16 son tamaños apropiados.

La salida que debe proporcionar dicha función `main()` es una tabla que recoge los tiempos que el algoritmo de ordenación ha empleado para cada una de las tallas analizadas, como la que se muestra a continuación.

Talla	Tiempo medio (seg.)
256	0.0002
512	0.0014001
1024	0.0065604
2048	0.0264817
4096	0.105847
8192	0.422826
16384	1.69155
32768	6.76698
65536	27.0463

---

<sup>1</sup>La función `int rand(void)`, en `stdlib.h`, devuelve un entero pseudo-aleatorio entre 0 y `RAND_MAX`. Las series generadas son siempre las mismas aún en distintas ejecuciones, puesto que siempre parten del mismo valor (semilla). Para evitarlo se debe suministrar en cada ejecución un nuevo valor para la semilla, mediante `void srand(unsigned int semilla)`.