

**Autores:** Francisco Javier Marqués Gaona, Germán Ruano García y David Subires Parra

## **Ejercicio 3 Tema 3**

### **Ejercicio 3.**

3. Desarrollar con más detalle el problema de asignación de parejas.  
¿Qué ocurre si se le da un cierto valor a la similitud de afinidades?  
En todos los casos se ha de:
- Presentar el esquema algorítmico detallado (pseudocódigo).
  - Seleccionar las estructuras de datos adecuadas para almacenar los datos.
  - Implementar el correspondiente código en Java.
  - Evaluar experimentalmente la eficiencia utilizando diversos juegos de prueba.
  - Almacenar juegos de prueba, resultados (indicando quiénes juegan) y tiempos de ejecución.
  - Analizar la eficiencia obtenida empíricamente frente a la teórica.
  - Hacer una crítica a la solución, buscando ejemplo problemáticos (problema de la desigualdad de afinidades).

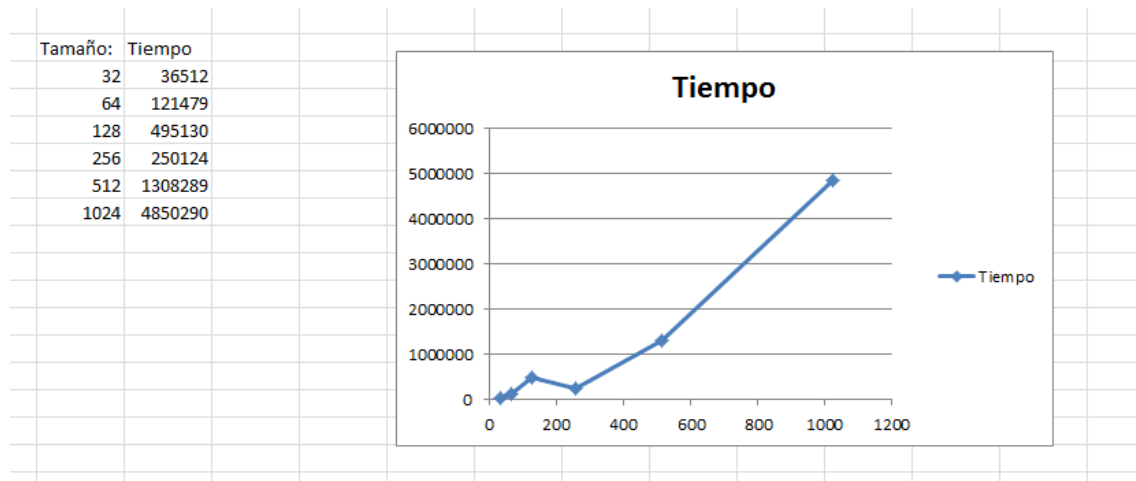
---

#### **Pseudocódigo generaAsignacionParejas**

```
generaAsignacionParejas ()  
  para cada e ∈ Empleados hacer  
    si e no tiene equipo hacer  
      para cada f ∈ Empleados hacer  
        si f no tiene equipo y e es distinto de f hacer  
          sumaAfinidad = afinidad(e,f) + afinidad(f,e)  
          si sumaAfinidad > mejorAfinidad hacer  
            mejorAfinidad = sumaAfinidad  
          almacenarParejaMejorAfinidad(e,f)  
  Devolver parejasMejorAfinidad  
}
```

### Tiempos de ejecución del algoritmo

Tiempo resultado de ejecutar el programa para distintos tamaños, empezando por una lista de 32 empleados para consecutivamente ir duplicando su tamaño:



**Eficiencia del algoritmo analizado teóricamente**

- Algoritmo AsignacionParejas ( $\Theta(n^2)$ ):

El algoritmo consta de dos bucles iterativos, por lo que su tiempo de ejecución será  $\Theta(n^2)$ . Aunque en la mitad de los casos, entrará en el bucle y comprobará que el empleado  $i$  ya está asignado, por lo tanto no realizará ninguna operación (salvo la comprobación de si está asignado o no), no obstante está ejecutando cada bucle  $n$  veces.