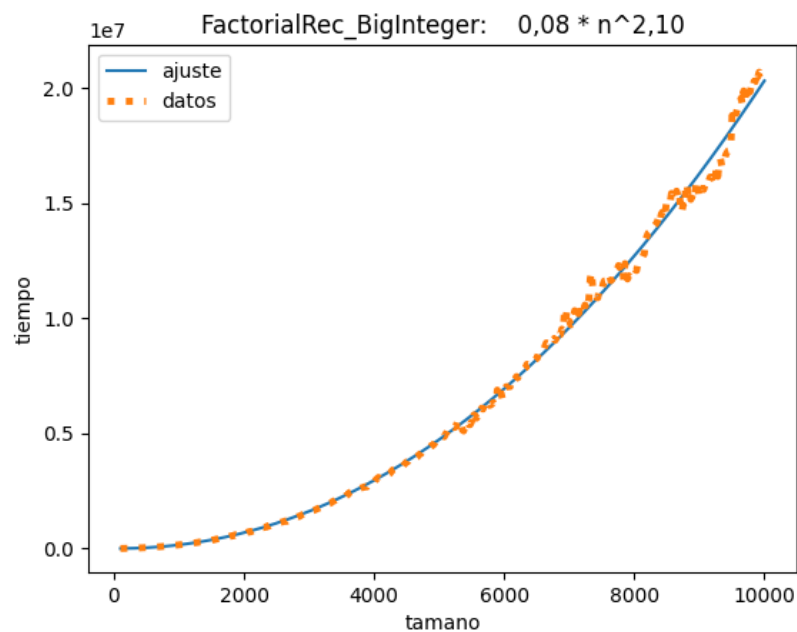
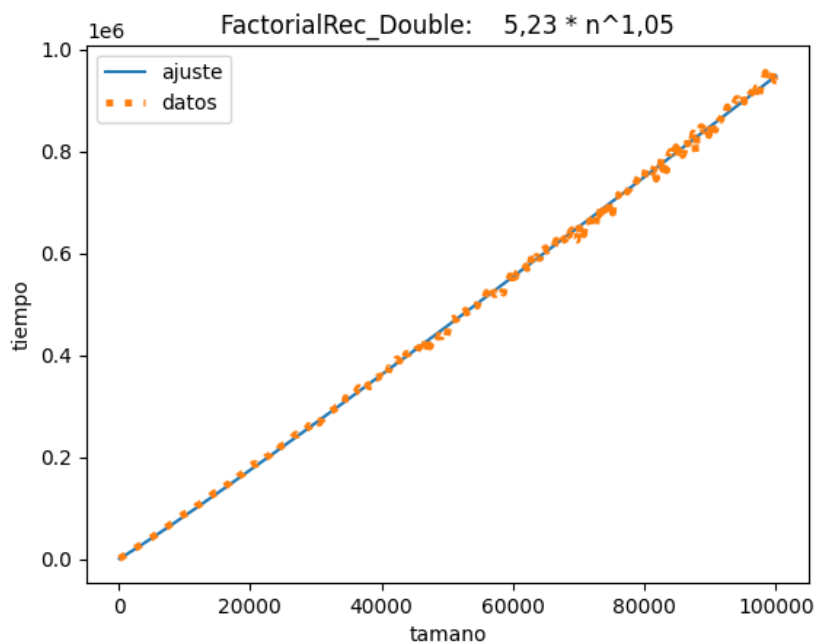
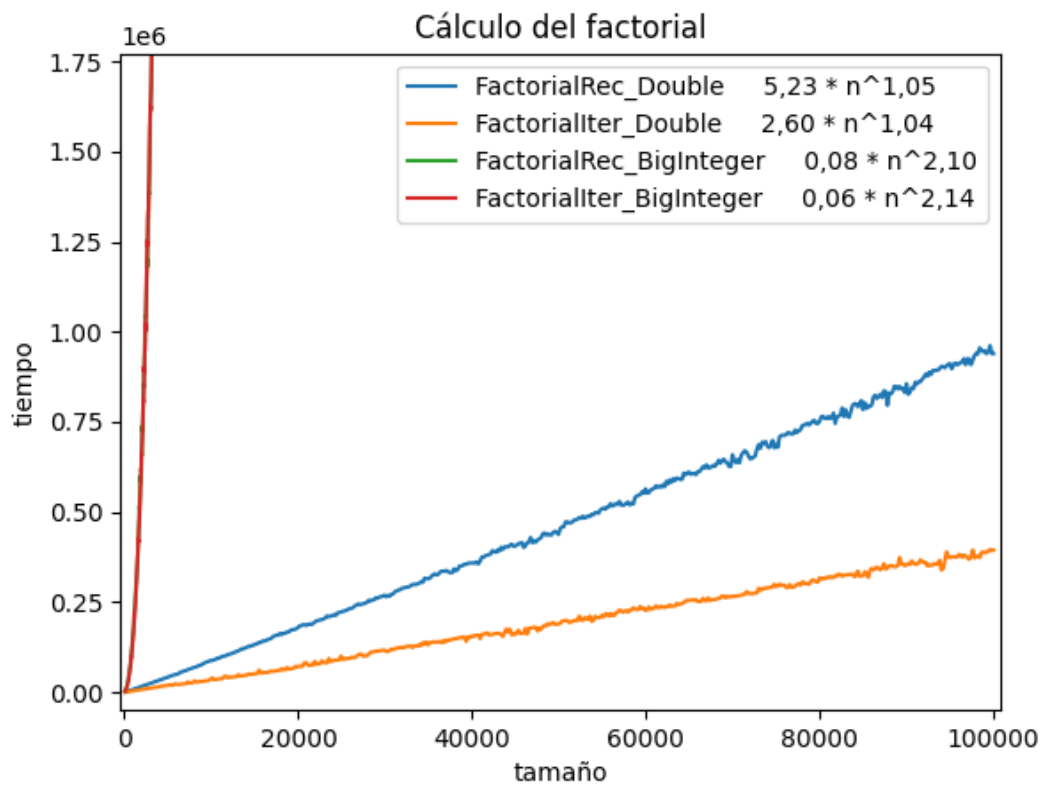


## Ejercicio 1.

En este ejercicio se debe analizar la complejidad del cálculo del factorial de un número en versiones recursiva e iterativa, usando los tipos Double y Big Integer.

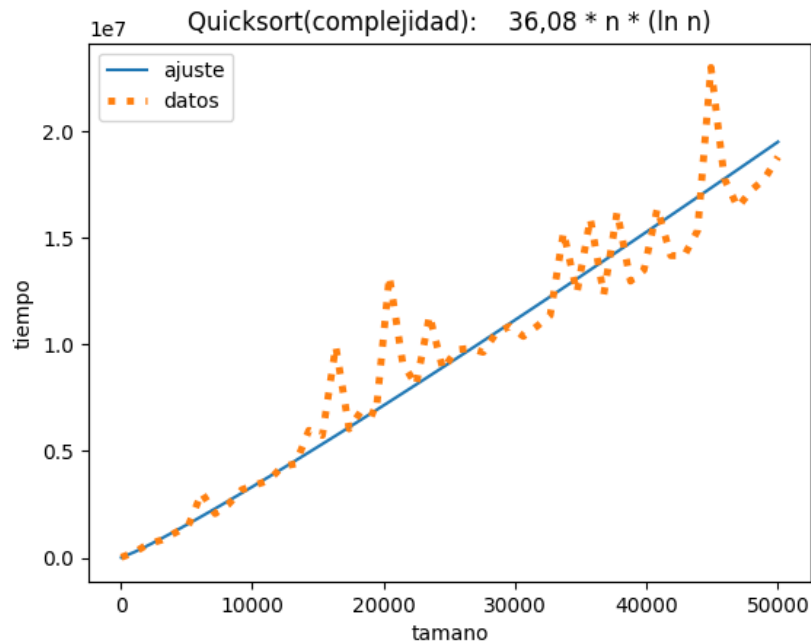
Los resultados deben mostrar que la complejidad de la función es de orden lineal al usar tipo Double, y cuadrático al usar BigInteger. Se muestra un ejemplo de la versión recursiva para Double y BigInteger. La comparativa entre los distintos tipos mostrará la mayor complejidad al usar BigInteger.



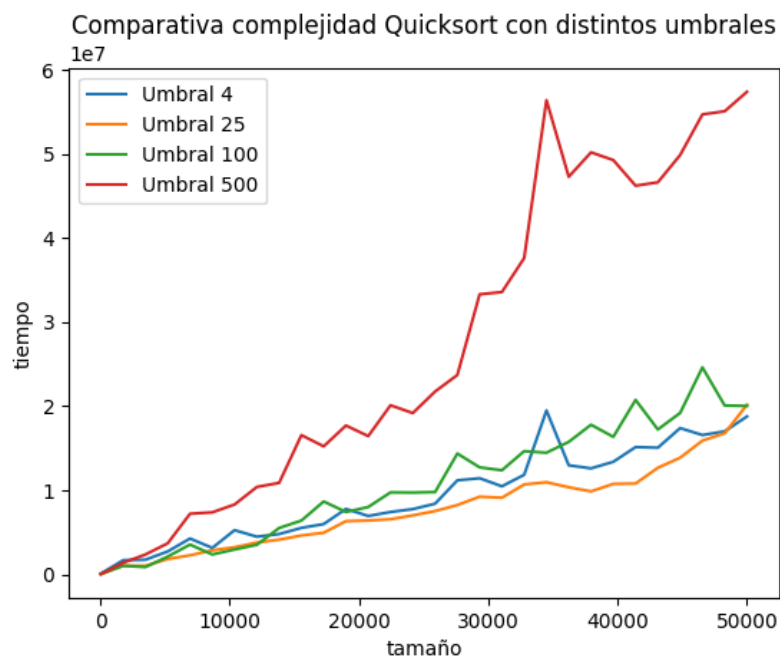


## Ejercicio 2.

El algoritmo Quicksort tiene complejidad  $n \log(n)$  en el caso medio. Se muestra un ejemplo del resultado esperado usando distintos tamaños de listas con un umbral de caso base fijo (4):



Por otro lado, se deben analizar las curvas de complejidad con algunos umbrales para distintos tamaños de listas. Se muestra un ejemplo de Quicksort con cuatro umbrales distintos:



### Ejercicio 3.

Utilizando los datos de entrada de los ficheros Ejercicio3DatosEntradaBinario.txt y Ejercicio3DatosEntradaNario.txt, los resultados de los diferentes tests deben ser:

#### Árboles binarios

Arbol: A(B,C)	Caracter: D	[[AB, AC]]
Arbol: A(B,C)	Caracter: C	[[AB]]
Arbol: A(B,C)	Caracter: A	[[[]]]
Arbol: A(B(C,D),E(F,_))	Caracter: H	[[ABC, ABD, AEF]]
Arbol: A(B(C,D),E(F,_))	Caracter: D	[[ABC, AEF]]
Arbol: A(B(C,D(E,F(G,H))),I(J,K))	Caracter: H	[[ABC, ABDE, ABDFG, AIJ, AIK]]
Arbol: A(B(C,D(E,F(G,H))),I(J,K))	Caracter: C	[[ABDE, ABDFG, ABDFH, AIJ, AIK]]

#### Árboles n-arios

Arbol: A(B,C,D)	Caracter: A	[[[]]]
Arbol: A(B,C,D)	Caracter: C	[[AB, AD]]
Arbol: A(B,C,D)	Caracter: D	[[AB, AC]]
Arbol: A(B(C,D,E),F(G,H,I),J(K,L))	Caracter: F	[[ABC, ABD, ABE, AJK, AJL]]
Arbol: A(B(C,D,E),F(G,H,I),J(K,L))	Caracter: K	[[ABC, ABD, ABE, AFG, AFH, AFI, AJL]]
Arbol: A(B(C,D(E,F(G,H,I),J),K))	Caracter: D	[[ABC, ABK]]
Arbol: A(B(C,D(E,F(G,H,I),J),K))	Caracter: I	[[ABC, ABDE, ABDFG, ABDFH, ABDJ, ABK]]

---

### Ejercicio 4.

Utilizando los datos de entrada de los ficheros Ejercicio4DatosEntradaBinario.txt y Ejercicio4DatosEntradaNario.txt, los resultados de los diferentes tests deben ser:

#### Árboles binarios

```
pepe(pepa,pepe): true
pepe(pepa,pep): false
ada(eda(ola,ale),eda(ele,ale)): true
ada(eda(ola,ale),eda(ele,al)): false
cafe(taza(bote,bolsa),perro(gato,leon)): true
cafe(taza(bote,bolsa),perro(gato,_)): false
cafe(taza(bote,bolsa),perro(gato,tortuga)): false
```

**Árboles n-arios**

```
pepe(pepa,pepe,pepo): true
pepe(pepa,pepe,pep): false
ada(eda(ola,ale,elo),eda(ele,ale,alo)): true
ada(eda(ola,ale,elo),eda(ele,ale,al)): false
cafe(taza(bote,bolsa,vaso),perro(gato,leon,tigre)): true
cafe(taza(bote,bolsa,vaso),perro(gato,leon)): false
cafe(taza(bote,bolsa,vaso),perro(gato,tortuga)): false
```

---