**Complejidad Computacional**

La complejidad computacional considera globalmente todos los posibles algoritmos para resolver un problema dado. Estamos interesados en la distinción que existe entre los problemas que pueden ser resueltos por un algoritmo en tiempo polinómico y los problemas para los cuales no conocemos ningún algoritmo polinómico, es decir, el mejor es no-polinómico. La teoría de la *NP*-Completitud no proporciona un método para obtener algoritmos de tiempo polinómico, ni dice que que estos algoritmos no existan. Lo que muestra es que muchos de los problemas para los cuales no conocemos algoritmos polinómicos están computacionalmente relacionados. De esta forma se presentarán definiciones que pretenden distinguir entre los **problemas tratables** (aquellos que no son tan duros) y los **problemas** **intratables** (duros o que consumen mucho tiempo). La mayoría de estos problemas ocurren como problemas de optimización combinatoria.

* **Complejidad del mejor caso:**Se entiende que es el menor número de operaciones necesarias para resolver un problema.
* **Complejidad del caso promedio:** Se entiende que es el número promedio de operaciones realizadas para solucionar un problema.

* **Complejidad del peor caso:** Se entiende que es el mayor número de operaciones necesarias para resolver un problema.
* **¿La complejidad está asociada a un problema o a una solución del problema?**

La complejidad está asociada a un problema debe ser porque están clasificados por la decisión en que se denota, ya sean de clase P, NP, NP-completos

* **¿Cómo se mide la complejidad?**

La complejidad se mide por la notación asintótica O (log n) y esta a su vez determina el grado de complejidad del algoritmo, el cual se mide por tiempo de procesamiento, espacio de memoria, etc.

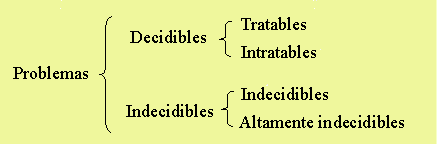
**CLASIFICACIÓN DE LOS PROBLEMAS**

Los problemas matemáticos se pueden dividir en primera instancia en dos grupos:

* **Problemas indecidibles**: aquellos que no se pueden resolver mediante un algoritmo.
* **Problemas decidibles**: aquellos que cuentan al menos con un algoritmo para su cómputo.

Que un problema sea decididle no implica que se pueda encontrar su solución. Esto permite separar los problemas decidibles en dos:

* **Intratables:** aquellos para los que no es factible obtener su solución.
* **Tratables:** aquellos para los que existe al menos un algoritmo capaz de resolverlo en un tiempo razonable.



**EJEMPLOS DE LOS PROBLEMAS**

**Intratables:** Aquellos para los que no es factible obtener su solución. A los problemas intratables se le conoce también como problemas NP (de orden no determinístico polinomial)

Ej: Problema Intratable

Determine las variables de espacio – tiempo que influyeron en la generación del Big Bang que dio origen al sistema solar.

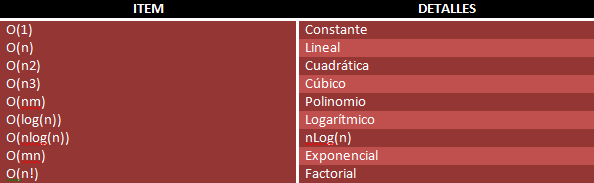
**Tratables:** Aquellos para los que existe al menos un algoritmo capaz de resolverlo en un tiempo razonable. A los problemas tratables se les conoce también como problemas de complejidad P (de orden polinomial)



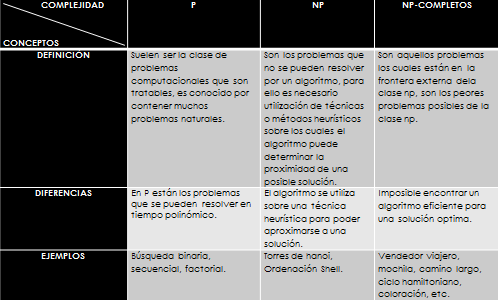
Ej: Problema Tratable

Calcule la rentabilidad económica de las acciones de Codelco en la bolsa internacional de valores del último semestre.

**TIPOS DE COMPLEJIDAD DE ALGORITMOS**

[](https://jorgecontrerasp.files.wordpress.com/2012/06/sin-tc3adtulo1.png)

**CLASIFICACIÓN DE PROBLEMAS SEGÚN COMPLEJIDAD**

[](https://jorgecontrerasp.files.wordpress.com/2012/06/sin-tc3adtulo2.png)