

DESARROLLANDO EL PENSAMIENTO COMPUTACIONAL CLASE #5 - BÚSQUEDA Y ORDENAMIENTO DE DATOS¹

Al final de esta clase serás capaz de:

1. **Entender** cómo el computador busca y ordena datos
2. **Desarrollar** tu intuición sobre cómo buscar y ordenar datos de manera más rápida

Actividad #1 - Entendiendo cómo busca datos el computador

Imagina que tienes que buscar una palabra en un diccionario. ¿Cómo lo harías? Describe los pasos del algoritmo para resolver intuitivamente este problema:

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____

¿Cómo buscarías en el diccionario el significado de la palabra “árbol”?



¿Cómo buscarías en el diccionario el significado de la palabra “zapato”?



¿Cómo buscarías en el diccionario una palabra cualquiera?



¹ Parte de este material fue adaptado de: Computer Science Unplugged (Tim Bell, Ian H. Witten, Mike Fellows). Derivado con permiso bajo licencia *Creative Commons* “Reconocimiento-NoComercial-CompartirIgual 3.0”.

¿Por qué puedes buscar así en un diccionario? ¿Qué pasaría si las palabras estuvieran desordenadas?

¿Si te hacen adivinar un número entre 1 y 100? ¿Cómo lo harías? ¿Con qué número intentarías primero? ¿Por qué?

¿Qué hace el siguiente programa en Scratch?

```
al presionar bandera verde clicada
  fijar intentos a 0
  fijar numero a número al azar entre 1 y 100
  repetir 10
    preguntar numero? y esperar
    si numero = respuesta
      decir Ganaste!! por 2 segundos
      detener todo
    si no
      si numero < respuesta
        pensar Es menor... por 2 segundos
        cambiar intentos por 1
      si no
        pensar Es mayor... por 2 segundos
        cambiar intentos por 1
  decir :-| por 2 segundos
  decir unir El número era: numero por 2 segundos
```

Abre el archivo Adivinador.sb y comprueba tu respuesta anterior. Modifica el programa para adivinar un número entre 1 y 10, entre 1 y 100, y entre 1 y 1000. Intenta adivinar el número con cada una de las versiones del juego y completa la siguiente tabla:

Adivinar un número entre 1 y ...	Número de intentos
10	
100	
1000	

Comenta tus resultados con un compañero. ¿Se les ocurre alguna estrategia para adivinar el número secreto en el menor número de intentos?

Actividad #2 - Entendiendo cómo ordena datos el computador

Los computadores se usan muy frecuentemente para ordenar datos (información) en algún orden. Por ejemplo, nombres en orden alfabético o correos electrónicos ordenados por la fecha. Cuando la información está ordenada, es muy rápido buscar un dato específico en ella: así si contamos con datos numéricos, cuando están ordenados es directo encontrar el más pequeño y el más grande.

Existen distintas estrategias para ordenar y, afortunadamente, existen algunas muy rápidas. En esta actividad queremos que descubras diferentes métodos para ordenar y veas cómo un método inteligente es mucho más rápido que uno simple.

Objetivo: Encontrar un método para ordenar en orden ascendente un grupo de tarjetas marcadas con números.

En esta actividad vamos a intentar replicar cómo el computador trabaja para ordenar un conjunto de datos. Para ello, forma un grupo con seis otros compañeros. Tu profesor les entregará a cada uno una tarjeta con un número y **no se lo debes mostrar a nadie. Este es tu número secreto.** La idea es que descubran una estrategia para que ustedes se ordenen, haciendo una comparación a la vez, de acuerdo al número de su carta. Como el computador puede sólo comparar dos números a la vez, tú puedes sólo comparar, en cada instante, tu carta con la de un/a compañero/a y tomar una decisión.

Piensen en conjunto por unos minutos qué estrategia se les ocurre para ordenar las tarjetas, de menor a mayor, en el menor número de comparaciones. Planteen su estrategia en forma de algoritmo:

¡Ahora es momento de poner en práctica su estrategia! Por turnos, cada miembro del grupo escogerá con quién comparará su tarjeta, ambos le mostrarán sus números a su profesor, quien les dirá cuál de los dos tiene el número mayor.

¿Pudieron ordenar correctamente **todas** las tarjetas? ¿Cuántas comparaciones tuvieron que hacer para poder cumplir con la tarea?

--

Ahora que pusieron en práctica su estrategia como grupo, cambien sus tarjetas y repitan la actividad. ¿Cuántas comparaciones hicieron ahora? ¿Se les ocurre una mejor manera de mejorar su algoritmo?

Comenten su estrategia con los otros grupos. Después de discutir, ¿cuál crees que es la que mejor funciona? Plántela en forma de algoritmo:

Ahora vamos a ver formas en que tradicionalmente los computadores pueden ordenar conjuntos de datos (como por ejemplo, ustedes cuando ordenaban sus tarjetas). Estos algoritmos fueron desarrollados a lo largo del tiempo por distintos científicos, ingenieros y mentes brillantes de la computación.

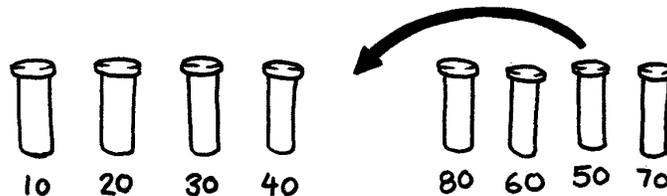
1. Ordenamiento por Selección.

<https://www.youtube.com/watch?v=Ns4TPTC8whw>

Inicialmente el grupo ordenado está vacío y el grupo desordenado está formado por todos los/las alumnos/as.

Este método funciona como sigue:

1. Encontrar el/la alumno/a que tiene el número más pequeño del grupo que aún no está ordenado. Comenzando de izquierda a derecha, cada dos alumnos/as van comparando su número hasta encontrar el que contiene el menor
2. El/La alumno/a con el número más pequeño pasa al grupo de los ordenados
3. Si era el último, terminar
4. Ir al paso 1 nuevamente



Cuenta el número de comparaciones que haces.

¿Cuándo es mayor el número de comparaciones?

2. Ordenamiento rápido: Quicksort

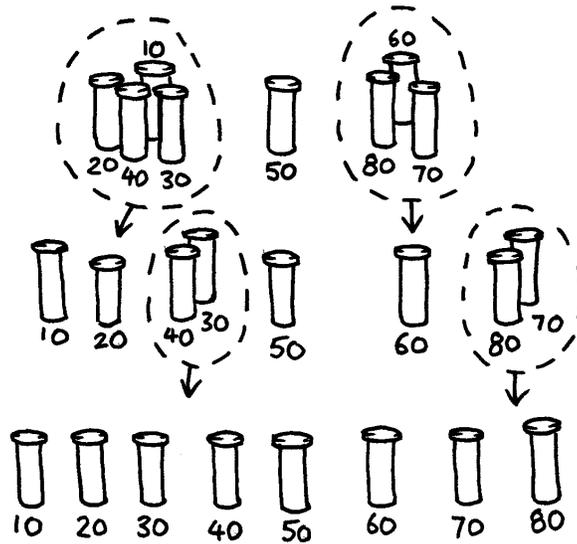
<https://www.youtube.com/watch?v=ywWBy6J5gz8>

Quicksort es mucho más rápido que el algoritmo de selección, en particular para conjuntos grandes. Este algoritmo usa una estrategia muy usada en computación que se llama **dividir y conquistar**.

El algoritmo funciona así:

1. Escoge un/a alumno/a al azar y pídele que se ponga a un lado
2. Comparar el resto de los alumnos con el/la alumno/ha seleccionado/a. Pon los/las que tienen un número más grande a la derecha del/la seleccionado/a y los/las que tienen un número más pequeño a la izquierda del/la seleccionado/a.
3. Repetir este proceso con cada uno de los grupos hasta que cada grupo quede con un

alumno/a. Cuando se logra esto último los alumnos/as están ordenados



¿Cuántas comparaciones toma este proceso?

Tú encontrarás que quicksort es más rápido a menos que siempre has seleccionado para comparar el más pequeño o el más grande de cada grupo. Si tú tienes suerte, y siempre el seleccionado es el valor que está en la mitad, sólo necesitaras 14 comparaciones mientras que para el mismo ejemplo, el algoritmo anterior toma (ordenamiento por selección) habrá tomado 28 comparaciones. En cualquier caso, quicksort no será nunca peor que el algoritmo de selección.

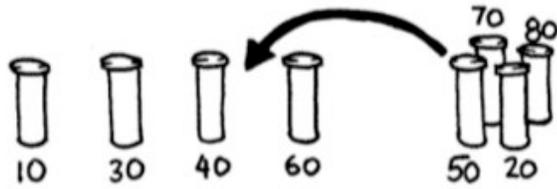
¿En qué situaciones aplicarías este algoritmo?

Consideremos de aquí en adelante que cualquier objeto que se pueda ordenar de acuerdo a un valor.

3. Algoritmo por Inserción

<https://www.youtube.com/watch?v=ROaLU379l3U>

Este algoritmo funciona sacando un objeto de los que no están ordenados y poniéndolo en la posición correcta en el que tiene al grupo ya ordenado. Con cada inserción, el grupo desordenado disminuye en uno y el ordenado aumenta en uno. Los jugadores de cartas usan este método para ordenar una mano.

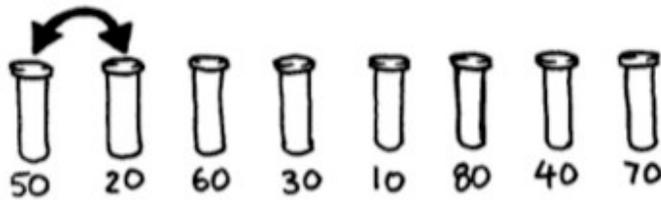


¿Piensas que este algoritmo es rápido o lento? ¿Por qué?

4. Método de la Burbuja

<https://www.youtube.com/watch?v=lyZQPjUT5B4>

Este método recorre la lista de números, comparando de dos en dos e intercambiando los que no están en orden. La lista está ordenada cuando no es necesario intercambiar ninguno. Este método es fácil de entender pero no es rápido.



¿Por qué este algoritmo no es rápido (como quicksort)?

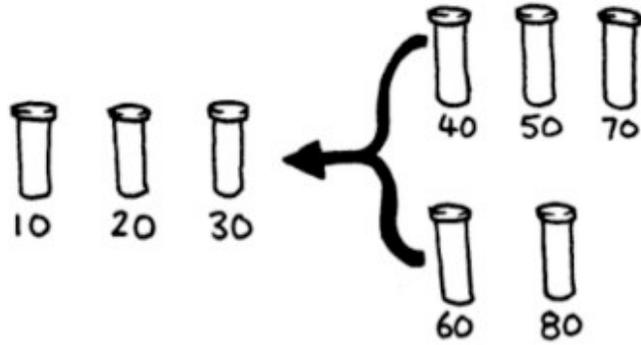
5. Mergesort

https://www.youtube.com/watch?v=XaqR3G_NVoo

Este es otro método basado en la estrategia dividir y conquistar. Primero divide la lista de objetos en dos partes de igual tamaño (o casi igual). Cada lista es ordenada y luego las dos listas ordenadas son mezcladas para ordenar la lista entera.

Mezclar dos listas ordenadas es fácil: repetidamente quitas el más pequeño de las dos listas y lo agregas a la lista en donde los estás ordenando. ¿Cómo ordenas las listas más

pequeñas? Aplicando el mismo procedimiento. En algún momento todas las listas quedan de tamaño 1 y están ordenadas.



De todos los algoritmos que viste anteriormente, ¿cuál crees que es el que mejor funciona? ¿Por qué?

De todos los algoritmos que viste anteriormente, ¿cuál crees que es el más fácil de aplicar? ¿y el más difícil? ¿Por qué?

¿Cuál ocuparías para ordenar un conjunto de datos (por ejemplo, alfabéticamente los nombres de todos tus compañeros de curso en el colegio)?
