

Requerimientos

Alumnos Gallegos Mota Victor Hugo y Carlos Cruz Rangel

LENGUAJES DE PROGRAMACIÓN USADOS:

JAVA (Programación imperativa)

- Probado en las siguientes versiones
 - openjdk version "11.0.11" 2021-04-20
 - openjdk version "1.8.0_292"

ELIXIR (Programación declarativa)

- Probado en las siguiente versión
 - Elixir 1.9.1 (compiled with Erlang/OTP 22)


¿CÓMO DESCARGAR ELIXIR?

Documentación para instalar elixir en diferentes sistemas operativos.

<https://elixir-lang.org/install.html>

Probado en Ubuntu 20.04.3 LTS

NOTA:

Ambos programas fueron desarrollados con el editor de código [Visual Studio Code](#)  y ejecutados en el mismo, sin embargo en las siguientes instrucciones se especifica cómo ejecutar y probar ambos programas desde una terminal en ubuntu ...

¿CÓMO EJECUTAR LOS PROGRAMAS?

- **EN JAVA** De la forma tradicional podemos compilar nuestro programa desde la terminal, para esto basta con abrirla y situarnos en la carpeta contenedora del programa ej: **Tarea 1/src/JAVA** posteriormente debemos ingresar el siguiente comando **javac ProgramacionImperativa.java** esto generara un archivo ProgramacionImperativa.class para lograr compilar nuestro programa y por último ingresar el siguiente comando para ejecutarlo **java ProgramacionImperativa.java**

```
Tarea 1/src/JAVA on 🌱 rama_de_victor [👤] via = v11.0.11
> javac ProgramacionImperativa.java
Tarea 1/src/JAVA on 🌱 rama_de_victor [👤] via = v11.0.11 took 3s
> java ProgramacionImperativa.java
Conjunto de datos dado:
6 5 3 1 8 7 2 4
Conjunto de datos ordenado por MERGESORT:
1 2 3 4 5 6 7 8
Conjunto de datos ordenado por HEAPSORT:
1 2 3 4 5 6 7 8

Busqueda Binaria de un Elemento sobre el arreglo recién ordenado:
1 2 3 4 5 6 7 8
Elemento 1 encontrado en el índice 0
```

COMPLEJIDAD EN TIEMPO DE JAVA PROGRAMACIÓN DECLARATIVA

Algoritmo	Descripción	Complejidad en tiempo
Merge Sort	Dividir la matriz en submatrices más pequeñas hasta lograr pares de elementos, y luego combinarlos de tal manera que estén en orden	$O(n \log (n))$
Heap Sort	Crear un heap máximo (o mínimo) y extraer el primer elemento del montón (o raíz), y luego enviarlo al final del heap. Disminuir el tamaño del montón y repetir hasta que el heap tenga un solo nodo.	$O(n \log (n))$
Búsqueda Binaria	Dividir repetidamente una matriz a la mitad. Comience con un intervalo que cubra toda la matriz. Si el valor de la clave de búsqueda es menor que el elemento en el medio del intervalo, reduzca el intervalo a la mitad inferior. De lo contrario, acoplarlo a la mitad superior. Verifique repetidamente hasta que se encuentre el valor o el intervalo esté vacío.	$O(\log n)$ cuya relación recursiva es $T(n / 2) + O(1)$

- **EN ELIXIR**, de la misma forma debemos ejecutar desde la terminal, para esto basta con abrirla y situarnos en la carpeta contenedora del programa ej: **Tarea 1/src/ELIXIR** posteriormente debemos ingresar el siguiente comando **> iex** esto abrirá el intérprete de nuestro programa, posteriormente debemos ingresar el siguiente comando **> iex(1)> c("ProgramacionDeclarativa.ex")** para cargar el módulo de nuestro programa si todo sale bien lo mostrará como **[ProgramacionDeclarativa]** y en las siguientes líneas ponemos el comando de la función que queramos probar En el archivo "Programación Declarativa.ex" está documentado también como probarlo.

```
Tarea 1/src/ELIXIR on 🌱 rama_de_victor [🔒👤]
> iex
Erlang/OTP 22 [erts-10.6.4] [source] [64-bit] [smp:4:4] [ds:4:4:10] [async-threads:1]

Interactive Elixir (1.9.1) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> c("ProgramacionDeclarativa.ex")
[ProgramacionDeclarativa]
iex(2)> ProgramacionDeclarativa.mcd(10, 5)
5
iex(3)> ProgramacionDeclarativa.diferenciasimetrica()
#MapSet<["Superman", "WonderWoman"]>
iex(4)> ProgramacionDeclarativa.conjuntopotencia([1,2,3,4])
[
  [],
  [1],
  [2],
  [1, 2],
  [3],
  [1, 3],
  [2, 3],
  [1, 2, 3],
  [4],
  [1, 4],
  [2, 4],
  [1, 2, 4],
  [3, 4],
  [1, 3, 4],
  [2, 3, 4],
  [1, 2, 3, 4]
]
iex(5)> ProgramacionDeclarativa.quicksort_doble_pivote([6,4,2])
[2, 4, 6]
iex(6)> 
```