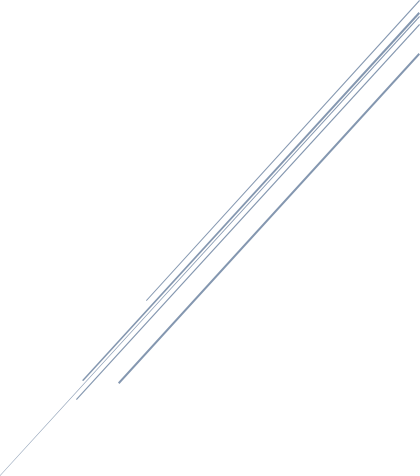
*TRABAJO PRÁCTICO 1 – CURSO: K2006*

*GRUPAL*

*INTEGRANTES:*

* *Lucas Daniel Caceres*
* *Walter Manuel Gomez Barrios*
* *Martina Roldán Pérez*



***ÍNDICE***

| *1.* [*Historia y presentación*](#_rrf10yaoi9p2) *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | *2* |
| --- | --- |
| *1.1* [*JavaScript*](#_nybe72gju6i9) *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | *2* |
| *1.2* [*GoLang*](#_xhgsl1jq7r3l) *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | *3* |
| *2.* [*Benchmarks*](#_4igvh6qlj1dm) *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | *4* |
| *2.1* [*Sort*](#_neh61wljvi8s) *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | *4* |
| *2.2* [*Búsqueda en listas*](#_az0hg29pr1t7) *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | *7* |
| *3.* [*Bibliografía*](#_96iayjklmof7) *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | *10* |

# HISTORIA Y PRESENTACIÓN

### JAVASCRIPT

JavaScript es un lenguaje de programación que se utiliza para añadir características interactivas a las páginas web, como mapas interactivos, animación, reproductores de video, juegos, eventos que ocurren cuando los botones son presionados o los datos son introducidos en los formularios, efectos de estilo dinámicos y más. Algunas de sus características son:

● Es multiplataforma.

● Es imperativo y estructurado.

● Es prototipado.

● Es interpretado: ejecuta las instrucciones directamente, sin una previa compilación del programa a instrucciones en lenguaje máquina.

● Está orientado a objetos y eventos.

Antes de hablar sobre el nacimiento de JavaScript, es importante mencionar que Netscape Communications Corporation, liderada por Marc Andreessen, desarrolló el navegador Netscape Navigator en 1994, el primer navegador masivo y popular. Debido a esto, Microsoft, liderada por Bill Gates, crea Internet Explorer; comenzando un enfrentamiento entre estas dos empresas.

En 1995, Netscape se da cuenta de que las páginas web eran demasiado simples, solamente se podía consumir información, leer texto y ver imágenes; el usuario no tenía forma de interactuar sin tener que recargar toda la página. Para cambiar esto y mejorarlas, Netscape contrató a Brendan Eich para desarrollar un lenguaje de programación que funcione en el navegador, un lenguaje de scripting que permitiera hacer algunas cosas básicas. En tan solo una semana, se crea LiveScript, posteriormente llamado JavaScript aprovechando la fama que tenía Java en ese momento (aunque no tenían nada que ver uno con el otro). Frente a esto, nuevamente Microsoft no quiere quedarse atrás, por lo que crean su versión de JavaScript: JScript.

Como estos dos lenguajes no eran del todo compatibles, surge la idea de necesitar un estándar para su desarrollo. Netscape le envía el código de JavaScript a ECMA (European Computer Manufacturers Association), una organización internacional cuyo objetivo es desarrollar estándares de tecnologías, entonces:

* En 1997 sale ECMAScript 1, pero a pesar de esto, Microsoft siguió utilizando JScript
* En 1998 sale ECMAScript 2
* En 1999 sale ECMAScript 3
* Hubo un intento de lanzar una versión 4, pero se fue retrasando hasta abandonar esta idea. Recién en 2009 se aprobó ECMAScript 5.

Durante el abandono que JavaScript sufrió por una década, se lo tomaba como un lenguaje simple y el cual sólo servía para hacer cosas muy básicas. Se pensaba que no tenía las características que tendría que tener un lenguaje serio como conectarse a bases de datos o instalarse en un servidor. Su popularidad subió gracias a que se desarrolló una implementación del mismo para servidores, lo que le abrió al lenguaje las puertas a otros usos más allá de las simples páginas web para lo cual era utilizado en un principio. En 2005 surgió Ayax, el que permitió a las páginas web actualizar contenido sin volver a cargar toda la página. Esto llevó a una mayor demanda de bibliotecas de JavaScript, como jQuery, React, Zepto; sumado a la construcción de frameworks, como AngularJs, Ember.js, Meteor y Node.js, que simplificaron la escritura de código y aportaron muchísimas mejoras al lenguaje.

Debido al aumento de su popularidad, desde el 2015 se mantiene constantemente actualizado.

### GO

Go, también conocido como Golang, es un lenguaje de programación creado por Google. Es conocido por su sencillez y eficiencia, y se emplea en una amplia gama de aplicaciones, desde programas de línea de comandos hasta sistemas distribuidos y aplicaciones web de gran envergadura.

Dado su enfoque en la eficiencia, Go resulta apropiado para la creación de aplicaciones que deben gestionar múltiples solicitudes simultáneas. Además, su diseño se enfoca en la seguridad, recolección de basura y la prevención de errores.

En los últimos años, ha ganado mucha fuerza. Grandes empresas confían en él, entre ellas: Dropbox, Docker, Facebook, Netflix, Uber y Twitter.

Características de Go

* Simplicidad: sintaxis clara, limpia y organizada.
* Facilidad de uso: está pensado para facilitar la vida al máximo a los desarrolladores.
* Compatibilidad: al igual que otros muchos lenguajes, es compatible con sistemas Windows, Mac OS X, Linux y FreeBSD.
* Código abierto: permite que usuarios externos desarrollen nuevos programas y librerías utilizándolo.
* Lenguaje compilado: programa que traduce el código fuente escrito en un lenguaje de programación a uno que la computadora “entiende”.

Go es un lenguaje de Programación moderno, lanzado en el año 2009. Con influencias de C y Pascal.

La historia del desarrollo de Go es muy reciente: puede remontarse hasta el año 2007, fue anunciado en Noviembre de 2009 y su versión 1.0 tomó forma recién en 2012. Luego de esta versión, Goland se fue actualizando anualmente (y llegaron a realizar más versiones en un mismo año). Las últimas versiones fueron:

* GO1.19: Agosto de 2022
* GO1.20: Febrero de 2023
* GO1.21.0: Agosto de 2023

# BENCHMARKS

### SORT

**En que consiste el Benchmark 1:**

En este benchmark estamos comparando un algoritmo de ordenamiento de una lista con 10.000 elementos siendo estos del tipo int (valores aleatorios).

Como se puede ver en el código de **JS** utilizamos una función **generateRandomArray()** la cual lee de un archivo y le inserta los valores utilizando el paquete “**fs”**, retornando el array. Para poder medir el tiempo de ejecución se utiliza el método **time** y **timeEnd** de **console** el cual te permite imprimir por consola el tiempo transcurrido, por último se utilizó el método **sort()** que la cual recibe como parámetro una función que define el método de ordenamiento.

En el caso de **GoLang** se utilizo un slice, en lugar de un array, ya que son una forma flexible y eficiente de representar arreglos, y a menudo se usan en lugar de arreglos debido a su tamaño dinámico y características adicionales.En este caso también se utilizo una funcion para generar el slice **generateRandomSlice()** el cual devuelve un Slice de 10.000 elementos obtenidos de un archivo utilizando los paquetes “bufio”, “os” y “strconv”, para finalmente usar la función **sort.Ints()**, para medir el tiempo de ejecución se utilizó el paquete “time” para crear una variable que captura el tiempo en que se inicializó la función y otra cuando termina, para finalmente utilizar la función Printf del paquete fmt para imprimir el tiempo total de ejecución.

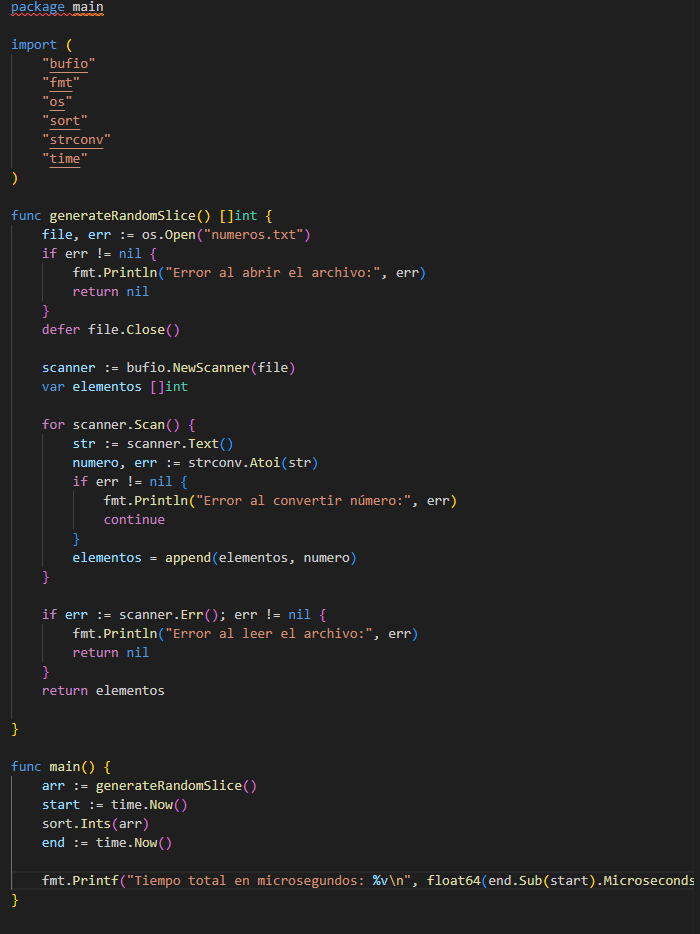
**JavaScript:**

****

**Tiempo en ejecutar:**



**GoLang:**



**Tiempo en ejecutar:**

****

**Conclusión Benchmark 1:**

Como podemos observar los resultados, la ejecución de de **GoLang** es 3 veces más rápida debido a la utilización de Slice la cual además de ser más eficiente que un array, tiene una función específica de ordenamiento para cada tipo de dato (algoritmo quickSort en este caso).

### BÚSQUEDA EN LISTAS:

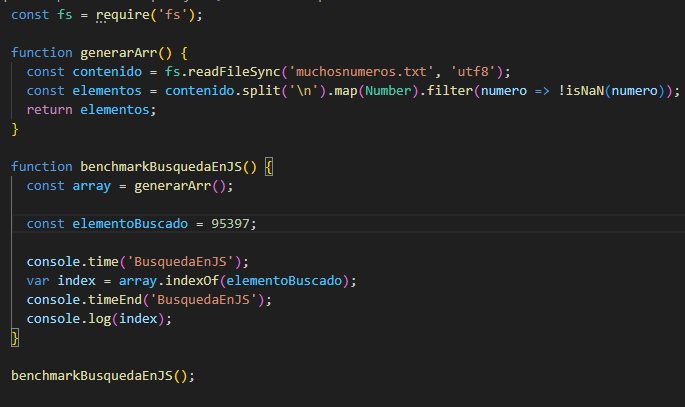
**En qué consiste Benchmark 2:**

En este segundo Benchmark se está buscando comparar el tiempo de ejecución de la búsqueda de un elemento en una lista de 1 millón de elementos.

En el caso de **JS** se utiliza una función la cual crea un array y luego le agrega los elementos a este utilizando un archivo, para realizar la búsqueda se utilizó el método de array **indexOf()**, el cual devuelve el índice de la primera ocurrencia del elemento pasado por parámetro, o es su defecto -1 en el caso que no lo encuentre.

En el caso de **GoLang** se utilizo la función **make()** para inicializar el Slice la cual tiene como parámetros el tipo de dato y la cantidad de elementos que tiene, para la búsqueda se utilizó **SearchInts()** del paquete “sort”en la cual se le pasa como parámetro el slice y el elemento a buscar en ese slice y retornar su índice, por último se volvió a utilizar el paquete time para calcular el tiempo.

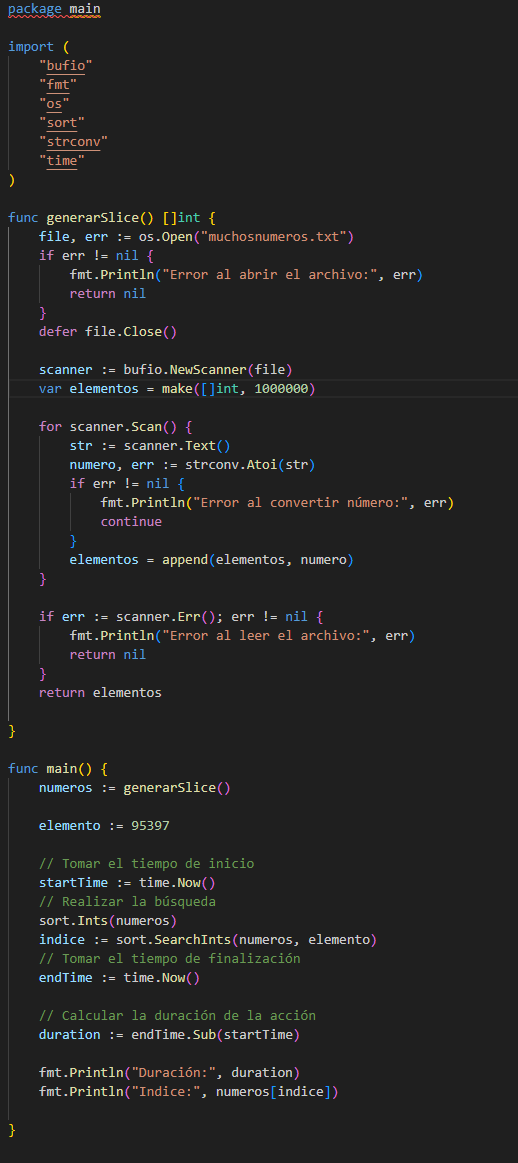
**JavaScript:**

****

**Tiempo en ejecutar:**



**GoLang:**

****

**Tiempo en ejecutar:**

****

**Conclusión Benchmark 2:**

Se puede ver que para un algoritmo de búsqueda de un elemento en una lista (de 1 millón de elementos) resultó ser más performante **JavaScript**, esto se debe a que para poder utilizar la función **SearchInts()** del paquete “sort” primero se debe tener el Slice ordenado, ya que esta función utiliza el algoritmo de búsqueda binaria, como resultado tardando proporcionalmente a la cantidad de elementos.

# Bibliografía:

<https://www.wikiwand.com/es/JavaScript>

<https://openwebinars.net/blog/que-es-javascript/#:~:text=JavaScript%20se%20introdujo%20en%201995,saber%20para%20agrupar%20p%C3%A1ginas%20web>.

<https://recluit.com/una-breve-revision-a-la-historia-de-javascript/>

<https://www.youtube.com/watch?v=FuCiNVP88pc>

<https://www.youtube.com/watch?v=ZGB_hQiI4ac>

<https://tomcopeland.blogs.com/EcmaScript.html>

<https://damiandeluca.com.ar/go-el-lenguaje-de-programacion-creado-por-google>

<https://www.linkedin.com/pulse/lenguaje-de-programaci%C3%B3n-go-el-c-del-siglo-xxi-palacio-gaviria/?originalSubdomain=es>

<https://go.dev/doc/>

<https://www.mytaskpanel.com/lenguaje-programacion-go/#:~:text=Go%2C%20tambi%C3%A9n%20conocido%20como%20Golang,aplicaciones%20web%20de%20alta%20escala>.

<https://www.digitaltoo.com/2023/06/28/estas-son-las-nuevas-funciones-integradas-de-golang-1-21/>

<https://go.dev/doc/devel/release>

<https://go.dev/ref/spec>

<https://go.dev/talks/2009/go_talk-20091030.pdf>

<https://github.com/go-lang-plugin-org/go-lang-idea-plugin/blob/master/grammars/go.bnf>