Capítulo 2

|  |  |  |
| --- | --- | --- |
| Jonathan Zinzan Salisbury Vega  *Universitat de les Illes Balears* Palma, España jonathan.salisbury1@estudiant.uib.cat | Joan Sansó Pericás  *Universitat de les Illes Balears* Palma, España joan.sanso4@estudiant.uib.cat | Joan Vilella Candia  *Universitat de les Illes Balears* Palma, España joan.vilella1@estudiant.uib.cat |

*El coste asintótico de los algoritmos ha sido uno de los principales focos de estudio (y de problemas) a lo largo de la historia de la informática. Las soluciones más sencillas suelen tener costes asintóticos no deseables dependiendo del problema a resolver. Además, encontrar soluciones óptimas suele traer verdaderos quebraderos de cabeza a los distintos programadores. En este proyecto se ha implementado una aplicación capaz de visualizar los distintos tiempos que tienen estos algoritmos. Para ello se ha utilizado el Modelo Vista Controlador. Utilizar esta arquitectura permite un mayor control de errores del programa y facilidad de reutilización como de escalado. Todo esto a coste de una mayor complejidad de código. Con esta práctica se busca poner en práctica la arquitectura MVC y ofrecer una herramienta al programador para poder discernir de una manera gráfica con la que estudiar la viabilidad de los distintos algoritmos.*

Keywords—MVC, rendimiento, coste asintótico, algoritmo.

# Introducción

Una de las máximas que ha de tener un programador a la hora de diseñar sus algoritmos, es el coste asintótico de los mismos. Muchas veces, por un simple descuido o por desconocimiento sobre la materia, se acaban implementando soluciones muy alejadas de lo que podríamos considerar como aceptable.

La herramienta que se ha desarrollado permite visualizar de una manera más fácil el coste asintótico de los distintos órdenes de complejidad. Esto permite al usuario hacerse una idea de la viabilidad de sus soluciones, no necesariamente para que sean las óptimas, pero sí para que se puedan realizar en un tiempo aceptable.

El segundo objetivo de la práctica ha sido la puesta en práctica del MVC (Modelo Vista Controlador) visto en clase. La estructura del documento será la siguiente, en primer lugar, se describe la implementación del modelo MVC y después se comentan los resultados de los distintos costes asintóticos.

# Modelo vista controlador

## Contexto y entorno del estudio

Uno de los requisitos a la hora de realizar la práctica es el uso del lenguaje de programación Java. Además, se ha dado la opción de elegir entre dos IDE (*Integrated Development Environment).*

* NetBeans
* IntelliJ

En este caso se ha escogido el IDE de NetBeans por familiaridad de uso. Además, se utiliza una herramienta de control de versiones (Git). Más específicamente su versión de escritorio *Github Desktop* por su facilidad de uso mediante interfaz gráfica.

## Diseño de la solución

Como ya se ha mencionado en la introducción, para realizar esta práctica se ha utilizado el patrón de arquitectura de software MVC.

# Bibliografía

|  |  |
| --- | --- |
| [1] | Oracle, «UISwing Concurrency - Swing Worker, » [En línea]. Available: https://docs.oracle.com/javase/tutorial/uiswing/concurrency/worker.html. |
| [2] | Wikipedia, «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Divide-and-conquer\_algorithm. |
| [3] | «Oracle Java Documentation - SwingWorker,» [En línea]. Available: https://docs.oracle.com/javase/7/docs/api/javax/swing/SwingWorker.html. |
| [4] | J. M. Alarcón, «Rendimiento de Algoritmos y Big O - Campus MVP,» [En línea]. Available: https://www.campusmvp.es/recursos/post/Rendimiento-de-algoritmos-y-notacion-Big-O.aspx. |
| [5] | P. M. H. Thomas V. Perneger, «Writing a research article: advice to beginners - Oxford Academic,» [En línea]. Available: https://academic.oup.com/intqhc/article/16/3/191/1814554. |