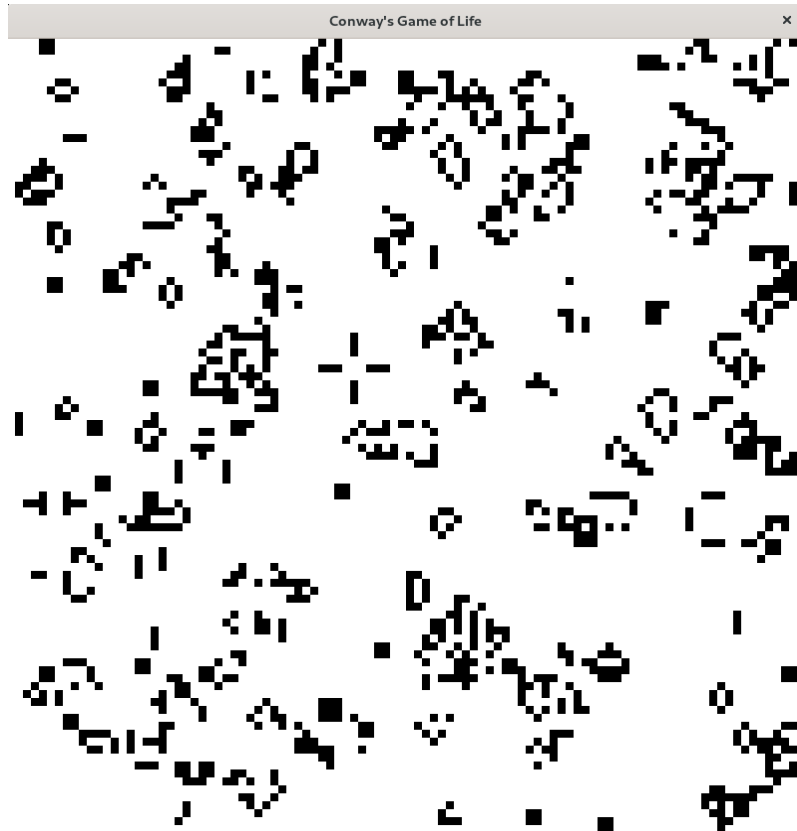


# Memoria de la práctica

18/04/2021



José María González Abad y Antonio Pablo García Sanabria

Universidad de Huelva

Fundamentos de Análisis de Algoritmos

Universidad de Huelva

# Índice

1. Introducción. Algoritmo de Búsqueda secuencial
2. Cálculo del tiempo teórico:
  - 2.1. Pseudocódigo y análisis de coste
  - 2.2. Tablas y Gráficas de coste
  - 2.3. Conclusiones
3. Cálculo del tiempo experimental:
  - 3.1. Tablas y Gráficas de coste
  - 3.2. Conclusiones
4. Comparación de los resultados teórico y experimental
5. Diseño de la aplicación
6. Conclusiones y valoraciones personales de la práctica

## 1. Introducción. Algoritmo de Búsqueda secuencial

Un algoritmo de búsqueda secuencial es aquel que reconoce cada valor de un vector individualmente y pasa al siguiente. Es uno de los algoritmos más básicos que se usan a la hora de encontrar valores en un arreglo.

El objetivo de este proyecto es comprender la importancia de analizar la eficiencia de los algoritmos y familiarizarse con las formas de llevarlos a cabo. La aplicación programada para analizar la eficiencia, muestra como realizar un análisis correcto de un algoritmo, mediante un estudio teórico y empírico.

## 2. Cálculo del tiempo teórico

A partir de la expresión del algoritmo, habrá que obtener la expresión de la función complejidad temporal ( $T(n)$ ) correspondiente al algoritmo. Se aplicarán las reglas conocidas para contar el número de operaciones que realiza un algoritmo. Este valor será expresado como una función de  $T(n)$  que dará el número de operaciones requeridas para un caso concreto del problema caracterizado por tener un tamaño  $n$ . El análisis lo realizaremos para los casos mejor, peor y medio.

### 2.1. Pseudocódigo y análisis de coste

Un pingo

### 2.2. Tablas y Gráficas de coste

Un pingo x2

### 2.3. Conclusiones

Podemos afirmar sin equivocarnos que el algoritmo sigue un modelo de crecimiento lineal. El coste temporal del algoritmo depende del tamaño del problema, pero incluso búsquedas del mismo tamaño pueden tener diferentes costes dependiendo de la ubicación del elemento a buscar. Aun así, al ser la función complejidad lineal, el algoritmo siempre sigue su modelo de crecimiento lineal, independientemente del caso en el que nos encontremos.

### 3. Cálculo del tiempo experimental

#### 3.1. Tablas y Gráficas de coste

aaaaa

#### 3.2. Conclusiones

Su **caso mejor** sucede cuando el elemento se encuentra en la primera posición del vector.

El **caso peor** ocurre cuando el elemento no se encuentra en el vector. Se usa -1.

Para el **caso medio** usamos la función *generarKey()* con la cual conseguimos valores aleatorios diferentes.

### 4. Comparación de los resultados teórico y experimental

aaaa

### 5. Diseño de la aplicación

aaaaa

### 6. Conclusiones y valoraciones personales de la práctica

aaa