

# Tarea 2 de Computación 1 (2021)

Ingeniería Forestal, CENUR Noreste

## Descripción del problema

Utilizando las funciones **distHamming** y **encontrarMasParecido** implementadas en la Tarea 1, se pretende que los grupos de estudiantes programen un buscador de palabras dentro de textos.

Ejemplo, dado el siguiente texto<sup>1</sup>:

*“La manera usual con la cual se planifica hoy para el mañana es en el vocabulario de ayer. Lo hacemos porque tratamos de avanzar con los conceptos que nos son familiares, los cuales adquieren un significado en nuestra experiencia pasada. Por supuesto, las palabras y los conceptos no encajan precisamente porque nuestro futuro difiere de nuestro pasado, pero las estiramos un poco. Los lingüistas están bastante familiarizados con el fenómeno en el cual los significados de las palabras evolucionan en el tiempo, pero también saben que éste es un “proceso lento y progresivo.”*

Si se quiere buscar la palabra *lingüistas*, el programa debería indicar que esta se encuentra a partir de la posición 421 (siendo la posición 1 la del primer carácter del texto)

## Se pide

Entregar 2 archivos .m separados cada uno con los siguientes nombres:

- **buscador.m** (contiene la función que buscará palabras en un texto)
- **texto\_ejemplo.m** (contiene un vector con el texto de prueba)

La ejecución para el ejemplo anterior sería la siguiente:

```
» texto_ejemplo
»
»buscador(texto, 'lingüistas')
palabra ubicada en la posición 421
```

El archivo **texto\_ejemplo.m** contendrá un vector texto:

```
texto = 'La manera usual con la cual se planifica hoy para el mañana es en el vocabulario de ayer. Lo hacemos porque tratamos de avanzar con los conceptos que nos son familiares, los cuales adquieren un significado en nuestra experiencia pasada. Por supuesto, las palabras y los conceptos no encajan precisamente porque nuestro futuro difiere de nuestro pasado, pero las estiramos un poco. Los lingüistas están bastante familiarizados con el fenómeno en el cual los significados de las palabras evolucionan en el tiempo, pero también saben que éste es un proceso lento y progresivo.'
```

## Importante

- No se debe entregar un archivo comprimido sino que los 2 archivos separados.
- Para la corrección, las tareas se ejecutarán con la versión 6+ de Octave.
- La ejecución se realizará desde la línea de comandos (sin interfaz gráfica).
- No está permitido utilizar facilidades de Octave que permitan resolver los problemas de forma trivial (funciones max, min, sort, etc.). SOLO SE PERMITIRÁ LAS UTILIZACIÓN DE LAS FUNCIONES **distHamming** y **encontrarMasParecido**.
- En esta tarea, como en todos los problemas de este curso, se valorará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios impartidos en el curso. De esta manera, se hará énfasis en buenas prácticas de programación que lleven a un código legible y bien documentado, tales como:
  - indentación adecuada
  - utilización correcta y apropiada de las estructuras de control
  - código claro y legible
  - algoritmos razonablemente eficientes
  - utilización de comentarios que documenten y complementen el código
  - nombres mnemotécnicos para variables, constantes, etc.

---

<sup>1</sup> extracto de “Cerca de la crueldad de realmente enseñar ciencias de la computación” de Edsger Dijkstra