

# Estructuras de Datos 2022-1

## Proyecto 3

Pedro Ulises Cervantes González `confundeme@ciencias.unam.mx`  
Yessica Janeth Pablo Martínez `yessica_j_pablo@ciencias.unam.mx`  
América Montserrat García Coronado `ame_coronado@ciencias.unam.mx`  
Emmanuel Cruz Hernández `emmanuel.cruzh@ciencias.unam.mx`  
Adrián Felipe Vélez Rivera `adrianf.velez@ciencias.unam.mx`

## Introducción

Un grupo de amigos estaba discutiendo sobre quién tenía mayor vocabulario, así que decidieron competir de la siguiente manera: a partir de una secuencia aleatoria de 9 letras, tenían un minuto para escribir la mayor cantidad de palabras. Mientras más larga la palabra, más puntos valía, de tal forma que al final del minuto el que tenga la puntuación más alta sería el ganador.

Tu objetivo es crear un programa con el que ellos puedan competir.

## Sobre el proyecto

El alumno tendrá que implementar un programa que permita jugar contrarreloj para escribir la mayor cantidad de palabras usando cualquier estructura de datos vista durante el curso.

## Secuencia inicial

Para iniciar la carrera contrarreloj para formar palabras se necesita de una **secuencia inicial de 9 letras**. Esta secuencia no debe llevar signos de puntuación y puede tener letras repetidas.

Deben existir las siguientes dos implementaciones y será el usuario quien elija cuál de ellas utilizar al momento de la ejecución:

- **Secuencia creada por la computadora:** La computadora generará una secuencia aleatoria de 9 letras sin signos de puntuación, donde puede haber letras repetidas. Si el alumno lo desea, puede darle menor probabilidad de aparecer en la secuencia a letras poco comunes en palabras como *W*, *X*, *Q*, etc.
- **Secuencia brindada por el usuario:** El usuario deberá introducir al programa una secuencia de 9 letras. El programa deberá eliminar los signos de puntuación de dicha secuencia y confirmar que son letras y no otro tipo de caracteres.

## Carrera contrarreloj

Una vez establecida la secuencia de letras, el usuario tendrá **un minuto** para brindar la mayor cantidad de palabras que se puedan formar a partir de dicha secuencia.

El alumno decidirá si implementar algún tipo de cronómetro para que el usuario sepa cuánto tiempo le queda, o bien, no mostrar ningún tipo de cronómetro y finalizado el tiempo simplemente se le avisa al usuario que se acabó.

## Validación de palabras

Aunque el usuario puede escribir cualquier palabra durante la carrera contrarreloj, solo se tomarán en cuenta para puntuación las que cumplan los siguientes criterios:

- **No se distinguen mayúsculas de minúsculas:** No debe haber distinción entre mayúsculas y minúsculas, de tal forma que si el usuario ingresa *hola*, *HoLa* y *holA*, entonces para el programa son la misma palabra.
- **Los signos de puntuación sí cuentan:** Palabras con las mismas letras pero distinta puntuación son diferentes, de tal forma que si el usuario ingresa *opera* y *ópera*, entonces para el programa son dos palabras diferentes.
- **Debe usar letras de la secuencia inicial:** Las palabras solo pueden contener letras de la secuencia inicial. Podrán repetir cierta letra  $x$  tantas veces como dicha letra  $x$  esté repetida en la secuencia inicial. Pueden agregar puntuación a las letras de la secuencia inicial.

Ejemplo: Si la secuencia inicial tiene las letras *OFIARASEJ*, entonces son palabras válidas *fresa*, *fría*, *rajó* y *orejas*, y no serían válidas palabras como *ceja*, *fresas* o *fajara*.

- **Deben pertenecer al diccionario:** Para verificar que las palabras sean válidas, deben estar contenidas en el siguiente banco de palabras: **Diccionario español**.

Su programa deberá cargar este banco de palabras al iniciar la ejecución y no cada vez que se quiera validar una palabra.

Recordando que el programa no debe hacer distinción entre mayúsculas y minúsculas, si en el banco de palabras está la palabra *éxito* y el jugador ingresó la palabra *Éxito*, entonces es una palabra válida.

**Nota:** Aunque el banco de palabras que les pedimos usar tiene muchas palabras, no podrá validar algunas palabras como conjugaciones de verbos o nombres propios. Esto no se considera una falla del programa, sino, una limitación del mismo.

## Puntuación de palabras

La puntuación del jugador será la suma de las puntuaciones de cada palabra válida que haya ingresado durante un minuto. Las palabras se puntúan como sigue:

- **No puntúan palabras repetidas:** Si el usuario ya ingresó cierta palabra  $p$  y puntuó, no puede volver a puntuar con esa misma palabra  $p$ .
- **Debe ser una palabra válida:** Si la palabra  $p$  usa letras que no están en la secuencia inicial o no está contenida en el banco de palabras, entonces la puntuación de  $p$  es 0.
- **Puntuación según su longitud:** Si la palabra  $p$  es válida y su longitud es  $k$ , entonces la puntuación de  $p$  es  $k^2$ .

El proyecto deberá ser **robusto**.

## Datos de entrada

- El banco de palabras con el que el programa hará la validación.
- La secuencia inicial de letras en el caso de que el usuario haya decidido otorgarla.
- Las palabras que el usuario escribe durante el minuto que dura el juego.
- Un documento con las estadísticas del juego, esto es, las 3 mejores puntuaciones por cada secuencia de letras. El programa puede usar siempre el mismo documento, por lo que no es necesario preguntarle al usuario su ruta.

## Datos de salida

- La secuencia inicial de letras en el caso de que el usuario haya decidido que la computadora debe otorgarla.
- La puntuación de cada palabra.
- Una vez concluido el tiempo del juego:
  - Una notificación de que el tiempo terminó.
  - Su puntuación total.
  - Las estadísticas con las 3 mejores puntuaciones de la secuencia con la que ha jugado el usuario.
- A solicitud del usuario, se deben devolver todas las estadísticas, esto es, las 3 mejores puntuaciones por cada secuencia de letras.

- El archivo donde se guardan las estadísticas debe actualizarse al finalizar la ejecución del programa. El programa puede usar siempre el mismo documento, por lo que no es necesario preguntarle al usuario su ruta. Se sugiere el uso de *Serializable*.

## Sobre la implementación

Pueden usar cualquier estructura de datos que consideren necesaria. En caso de usar alguna implementación de estructuras de datos existente en las bibliotecas de java, exceptuando arreglos, tendrán penalización de calificación.

Es muy importante que escojan una estructura de datos apropiada en cada parte del proyecto, así como buenos algoritmos, para que el tiempo de ejecución no afecte la experiencia de usuario. Si la ejecución de su programa tiene tiempos de ejecución excesivos, tendrán penalización de calificación.

Para este proyecto no es necesario usar interfaz gráfica, pero si la agregan habrá calificación adicional.

## Entrega

- El proyecto sigue los mismos lineamientos de entrega que las prácticas.
- La elaboración del proyecto puede ser en parejas.