

## UNIDAD TEMÁTICA 5 – ARBOLES GENÉRICOS y TRIES

### Trabajo de Aplicación 3

#### ESCENARIO

Una de las aplicaciones en las que el uso de un trie puede ser muy conveniente es en el desarrollo de sistemas de texto predictivo o “autocompletar”. Estamos muy acostumbrados al uso de estas funcionalidades, en nuestros teléfonos, en múltiples sitios web, incluso en muchas herramientas potentes de desarrollo de software.

#### EJERCICIO 0

Desarrolla un algoritmo en pseudocódigo abstracto, que, dado un trie que ha sido cargado con un conjunto de palabras (strings), permita obtener todas aquellas palabras contenidas que tengan como prefijo una string pasada como parámetro. Podemos utilizar una lista para almacenar estas palabras

**Del Tipo ArbolTrie**

**predecir (string unPrefijo)** //devuelve una LinkedList con las palabras

**del tipo NodoTrie**

**predecir (string unPrefijo, LinkedList unaLista)** // al terminar, en “unaLista” están todas las palabras que tienen como  
//prefijo la string “unPrefijo”

## EJERCICIO 1

Se desea, utilizando las estructuras de datos de TRIE que hemos visto, el pseudocódigo abstracto para “autocompletar” o “predecir”, la interfaz de usuario y un archivo de palabras provistos, desarrollar una funcionalidad del trie que permita que, a medida que digitamos caracteres en el campo de entrada, se vayan mostrando en el campo memo las palabras contenidas en la colección que tengan como prefijo la cadena actual del campo de entrada.

El equipo ha de realizar las siguientes tareas y productos:

1. Agregar a las estructuras de Trie existentes una funcionalidad “**predecir**”, que, dada una cadena de caracteres, devuelva el conjunto de palabras contenidas en el trie tales que la cadena de entrada sea un prefijo de ellas.
2. Utilizando las clases de interfaz de usuario provistas, programar un comportamiento que haga que a medida que se digitan cada nuevo carácter en el campo de entrada, se muestre en el campo memo las palabras sugeridas.
3. Crear el o los casos de prueba que sean convenientes y apropiados, cargando el trie con un pequeño conjunto de palabras de prueba
4. Cargar el trie con el conjunto de palabras suministrado “**palabras.txt**”.
5. Probar para diferentes combinaciones de caracteres de entrada.

**NOTA:** OBSERVA EL METODO `buscarNodoTrie(String s)`. ¿Qué hace?