

PARTE 3: Ejercicio de programación JAVA

Duración 60 minutos

Este ejercicio comprende **2** pasos:

1. Desarrollo de la funcionalidad especificada más abajo
2. Desarrollo de los casos de prueba ("test cases") para verificar la corrección de la funcionalidad implementada.

Escenario:

Escenario

Se han recibido una serie de mensajes encriptados en forma de Código binario (los caracteres son ceros y unos), y se desea comenzar a descifrarlos. Para ello se ha de comenzar identificando la ocurrencia de un cierto patrón en estos mensajes.

Se desea entonces utilizar un árbol de sufijos para **reconocer patrones en una secuencia de caracteres que representan el mensaje**.

En este caso, el **"alfabeto"** está integrado por los caracteres **"0"** y **"1"**.

Dado un mensaje (en el archivo **"mensaje.txt"**), se desea hallar todas las ocurrencias de un cierto patrón en el mismo.

PASO 1: Funcionalidad a desarrollar:

Descargar de la webasignatura el archivo "Parcial1-2021.zip" que contiene el Proyecto VSC a ser completado.

1. Crear la clase **TArbolTrie** que implementa la interfaz provista **IArbolTrie**, y crear la clase **TNodoTrie**.
2. Implementar los métodos necesarios para:
 - **Insertar** las cadenas en el árbol de sufijos
 - Dado un cierto patrón, **buscar** en el árbol de sufijos todas las ocurrencias del mismo, indicando las posiciones del mensaje en que comienzan.

Del tipo TArbolTrie:

public LinkedList<Integer> encontrarPatron(String patron) // devuelve una lista con las posiciones en que comienza el patrón en la secuencia

3. En la clase **Main**,
 - a. Generar todos los sufijos del mensaje (archivo **"mensaje.txt"**).
 - b. Instanciar un Trie e insertar todos los sufijos generados, teniendo cuidado de indicar al final de cada uno las posiciones de inicio en el mensaje.
 - c. Invocar al método **"encontrarPatron"**, y emitir por consola
 - i. Cantidad de veces que el patrón aparece en el mensaje
 - ii. Posiciones del mensaje en que comienza el patrón

Para los siguientes **2 patrones**:

- **"110101"**
- **"1101011"**

PASO 2: TEST CASES.

Implementa el o los **Casos de Prueba** necesarios para verificar el correcto funcionamiento de los métodos desarrollados.

RUBRICA DE CALIFICACIÓN: se utilizarán los siguientes criterios en la evaluación del trabajo remitido:

1. EJECUCIÓN: 50%

- No ejecuta o no termina / ejecuta incorrectamente / ejecuta 100% de acuerdo a lo solicitado

2. DESARROLLO 30%

- Estructura de datos utilizada (pertinencia, eficiencia)
- Cumplimiento de la interfaz publicada
- Código de inserción
- Código para “**encontrarPatron**”, a nivel de trie y de nodoTrie
- Calidad del código (nombres de variables y métodos, orientación a objetos, encapsulación, modularidad, etc.).

3. PRUEBAS DE UNIDAD 20%.

- Calidad de los tests desarrollados:
 - Sin tests / tests apenas básicos / tests completos (prueban todas las condiciones normales y de borde)