

Apellido y nombre: _____ Legajo: _____ Cursó con Prof: _____

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

WhatsApp

Temas evaluados: Abstracción, estructuras de datos enlazadas.

Contexto

Usted es parte de un equipo que desarrolla aplicaciones móviles para de mensajería, tipo WhatsApp. La aplicación actualiza las conversaciones según los eventos remotos que se producen: llegadas de mensajes nuevos, recepción y lectura de mensajes enviados, y conversaciones nuevas. Usted es el encargado desarrollar el analizador de mensajes en que detecta palabras “de riesgo”.

Componentes de la solución ya disponibles

El sistema tiene la función **bool ObtenerProximo(Mensaje& m)** que recibe como parámetro una estructura **Mensaje**. En caso de haber mensajes para procesar, lo devuelve en **m**, y retorna **true**, si no retorna **false** y no modifica **m**. Un mensaje está formado por:

idMensaje	idConversacion	idEmisor	idReceptor	Texto
-----------	----------------	----------	------------	-------

Los mensajes sin palabras “de riesgo” deberán ser enviados al receptor usando la función **void Enviar(Mensaje m, int idReceptor)**. En caso contrario el mensaje se deberá alertar a quien lo emite con la función **void Alertar(int idEmisor)**.

Está disponible la función **NodoPalabra* CrearPilaDePalabras(string texto)** que recibe un texto y retorna una pila las del texto.

Está disponible la variable global **in**, que es una conexión abierta in a un archivo binario, de mil palabras ordenadas alfabéticamente y ascendentemente, la longitud máxima de la palabra es **N**. Con esas palabras se determinar si los mensajes son riesgosos.

Problema

Desarrollar el procesamiento de los mensajes pendientes y enviar o alterar según el caso.

Se pide

1. Declarar TODAS las estructuras de datos requeridas para la resolución del problema.
2. Justifique con precisión la razón de la selección de cada una de las estructuras seleccionadas (persistencia, velocidad de procesamiento para este problema, utilización de recursos, o todo lo que estime adecuado).
3. Codifique o diagrame la función **EsRiesgoso** que dado un mensaje y retorna **true** si tres o más palabras del texto son inapropiadas, si no, **false**. Invoque a **CrearPilaDePalabras**.
4. Codifique o diagrame la función **ProcesarPendientes** que invoque a **ObtenerProximo** hasta que no queden mensajes pendientes para procesar. Para cada uno de ellos haga uso de la función **EsRiesgoso** y envíe o alerte, según corresponda.

Desarrolle cualquier otra función que invoque.

Funciones Disponibles

bool **ObtenerProximo** (Mensaje& m)
 NodoPalabra* **CrearPilaDePalabras**(string texto)
 void **Enviar**(Mensaje m, int idReceptor)
 void **Alertar**(int idEmisor)

Funciones a desarrollar (debe determinar valor de retorno y listas de parámetros)

EsRisegoso: Debe invocar a **CrearPilaDePalabras**. Debe determinar si el mensaje es de riesgo según el protocolo.

ProcesarPendiente: Debe invocar a **EsRiesgoso**, **ObtenerProximo**, **Enviar**, **Alertar**

Cualquier otra función que se requiera e invoque para dar solución al problema planteado

1.

```
struct Mensaje{
```

palabras:
 Arreglo de 26 posiciones, una por cada letra.
 Cada posición tiene una el principio de una lista de palabras ordenadas.
 La estructura en la que se va a cargar el archivo en caso de decidir no hacer búsqueda binaria en el mismo

2.

Permite un acceso rápido para la búsqueda al ser arreglo, no acota la cantidad de palabras riesgosas a ser arreglo de listas.

Se lee el file sólo una vez

La determinación de una palabra se hace desde la primera letra, luego, al estar ordenada se puede cortar la búsqueda rápidamente.

La inserción es un poco lenta porque hay que recorrer las listas hasta el final. Se podría solucionar con un puntero al final.

[O las razones que estimen correspondan según la seleccion](#)

3.

```
bool EsRiesgoso(m){
    s=CrearPilaDePalabras(m.texto)//función dada
    while not isEmpty(s){
        w=Pop(s)
        if(EsPalabraRiesgosa(s))
            ++nw
    }
    return nw>2
}
```

// función auxiliar para buscar la palabra. Aquí pueden ocurrir varias cosas: se puede decidir hacer búsqueda binaria en el archivo, Se puede optar por cargar los datos a memoria para agilizar la búsqueda, aquí puede ser en un vector o en una lista, queda la decisión para la justificación requerida en el punto 2.//

```
bool EsPalabraRiesgosa(string w){
    lista=palabras[PrimeraLetra(w)] // palabras ya está inicializado.
    while haya nodos
        if es igual
            return true
        if es mayor
            return false // para que seguir, no?
}
```

```
CargarPalabrasRiesgosas(){ // flujo in y arreglo palabras son variables externas
    while leer registro desde in con palabra
        InsertarAlFinal(palabras[PrimeraLetra(palabra)])
}
```

4.

```
ProcesarPendientes(){
    while(ObtenerProximoMensaje(m))
        if(EsRiesgoso(m))
            Alertar()
        else
            Enviar()
}
```