

Evaluación II.

Indicaciones

- **Resolver el problema de manera individual y en silencio.**
- **Puede auxiliarse en sus códigos o internet(no recomendado).**
- **Crear un proyecto nuevo en c++ bajo el nombre Diagnostico.**
- **Al terminar comprimir el proyecto en un zip y ponerle su nombre-apellido.**

Diagnostico de enfermedades.

Existen diferentes tipos de enfermedades cada una de las cuales posee síntomas característicos, por lo que al ingresar una serie de síntomas podemos obtener la enfermedad que posea el mayor porcentaje de coincidencias para estos síntomas.

Clase Enfermedad 40%:

Una vez descrito el algoritmo se plantea el UML de la clase Enfermedad:

Enfermedad
- nombre: string - sintomas: string[] - contador: int
+ Enfermedad() + Enfermedad(string) + agregarSintoma(string): void + coincidencia(string[]): double + getNombre(): string + getSintomas(): string* + getContador(): int

Para definir la longitud del arreglo de síntomas es necesario definir una variable global a enfermedad de tamaño diez, Ej: #define TAM 10.

Se describen los métodos más importantes:

- **Enfermedad(string):** se recibe una cadena que es el nombre de la enfermedad, además se inicializa contador a cero.
- **agregarSintoma(string) :** recibe una cadena de texto y se agrega a los síntomas en la posición contador y luego contador incrementa en uno, todo esto si contador es menor a TAM.

Programación II

- Coincidencia(string[]) : recibe un arreglo de string de tamaño TAM que se serán los síntomas a evaluar, se procede a comparar cada uno de los elementos del vector recibido con los de los síntomas guardados, para asegurarse recorrer cada uno de ellos es necesario dos bucles finitos

(for), en la comparación de los elementos, estos no deben estar vacíos. Se guarda en un acumulador las coincidencias encontradas y se retorna la siguiente fórmula : $(100 * \text{coincidencias}) / (\text{contador} * 1.0)$; que es una regla de tres simple, que devuelve en términos de porcentaje el factor de coincidencia.

Implementación en main.cpp :

Una vez terminada la codificación de la clase Enfermedad codificar los siguientes en el archivo main.cpp:

```
main.cpp x Enfermedad.h x
1  #include <iostream>
2  using namespace std;
3  #include "Enfermedad.h"
4
5  Enfermedad enfermedades[TAM];
6  int c=0;
7  double coincidencia_minima = 25.0;
8  void inicializarEnfermedades(){
9      enfermedades[c] = Enfermedad("gripe");
10     enfermedades[c].agregarSintoma("fiebre");
11     enfermedades[c].agregarSintoma("congestion_nasal");
12     enfermedades[c].agregarSintoma("dolor_muscular");
13     enfermedades[c].agregarSintoma("nauseas");
14     enfermedades[c].agregarSintoma("tos");
15     c++;
16     enfermedades[c] = Enfermedad("anemia");
17     enfermedades[c].agregarSintoma("dificultad_para_respirar");
18     enfermedades[c].agregarSintoma("mareo");
19     enfermedades[c].agregarSintoma("dolor_de_cabeza");
20     enfermedades[c].agregarSintoma("frio");
21     enfermedades[c].agregarSintoma("palidez");
22     c++;
23     enfermedades[c] = Enfermedad("asma");
24     enfermedades[c].agregarSintoma("dificultad_para_respirar");
25     enfermedades[c].agregarSintoma("silbidos_en_el_pecho");
26     enfermedades[c].agregarSintoma("flemas");
27     enfermedades[c].agregarSintoma("fatiga");
28     c++;
29 }
30 void mostrarEnfermedades(){
39 void buscarDiagnostico(){
67 void agregarEnfermedad(){
70 int main(){
```

Programación II

Se describen los metodos:

- void inicializarEnfermedades() **5%**: se guardan 3 enfermedades con sus sintomas para posteriores pruebas.
- void mostrarEnfermedades() **10%**: muestra TODAS las enfermades con sus respectivos sintomas.

```

C:\ "C:\Documents and Settings\Jonathan\Escritorio\Diagnostico\bin\Debug\Diagnostico.exe"
Diagnostico
Nombre : gripe
Sintomas :
    fiebre
    congestion_nasal
    dolor_muscular
    nauseas
    tos
Nombre : anemia
Sintomas :
    dificultad_para_respirar
    mareo
    dolor_de_cabeza
    frio
    palidez
Nombre : asma
Sintomas :
    dificultad_para_respirar
    silbidos_en_el_pecho
    flemas
    fatiga

Process returned 0 (0x0)   execution time : 0.020 s
Press any key to continue.
    
```

- void buscarDiagnostico() **30%**: se crea un arreglo de sintomas de tamaño TAM y se piden dentro de un bucle (do-while) de manera que se le pregunte al final si desea agregar otro sintoma, cuando sea 'n' o el contador supere a TAM se detendra.
Cuando se deje de pedir se procede a comparar cada una de las n enfermedades guardadas en el arreglo con el vector de sintomas utilizando la funcion coincidencia, se guarda ese resultado y se evalua con la variable coincidencia_maxima, si es mayor o igual se procede a mostrar esa enfermedad y sus sintomas.

```

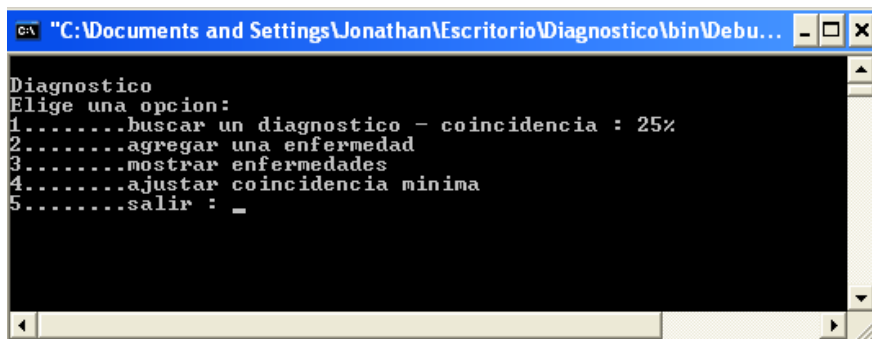
C:\ "C:\Documents and Settings\Jonathan\Escritorio\Diagnostico\bin\Debug\Diagnostico.exe"
Diagnostico
Ingresa el sintoma : flemas
Desea agregar otro sintoma (s/n) : s
Ingresa el sintoma : dificultad_para_respirar
Desea agregar otro sintoma (s/n) : s
Ingresa el sintoma : palidez
Desea agregar otro sintoma (s/n) : s
Ingresa el sintoma : fatiga
Desea agregar otro sintoma (s/n) : n

Coincidencia del : 40%
Nombre : anemia
Sintomas :
    dificultad_para_respirar
    mareo
    dolor_de_cabeza
    frio
    palidez
Coincidencia del : 75%
Nombre : asma
Sintomas :
    dificultad_para_respirar
    silbidos_en_el_pecho
    flemas
    fatiga

Process returned 0 (0x0)   execution time : 80.143 s
    
```

Programación II

- void agregarEnfermedad() **10%**: pide el nombre de la enfermedad, luego los sintomas, estos dentro de un bucle (do-while) de manera que se detenga cuando no desee agregar otro sintoma o supere el tamaño TAM.
- int main() **5%**: primero se llama a la funcion inicializar, luego se construira un menú con estas opciones:



```
G:\ "C:\Documents and Settings\Jonathan\Escritorio\Diagnostico\bin\Debu...
Diagnostico
Elige una opcion:
1.....buscar un diagnostico - coincidencia : 25%
2.....agregar una enfermedad
3.....mostrar enfermedades
4.....ajustar coincidencia minima
5.....salir : _
```