

Universidad Tecnológica Nacional

Facultad Regional Avellaneda



Técnico Superior en Programación

Materia: Laboratorio de Computación I

Apellido:		Fecha:	
Nombre:		Docente:	
División:		Nota:	
Legajo:		Firma:	
Instancia:	PP	RPP	SP
		RSP	FIN
			X

Desarrollar en ANSI C:

Los logs son archivos en los cuales se registra información inherente a los procesos ejecutados en un servidor. Se requiere desarrollar un programa el cual permita separar y filtrar los mensajes de fallos reportados por un servidor web. Todos los mensajes generados por el servidor, se escribirán en un archivo **log.txt**. Cada línea del archivo tendrá información separada por “;” y el siguiente formato:

Fecha	Hora	ID del servicio que falló	Gravedad del error (0 a 9)	Mensaje de error (64 caracteres máximo)
-------	------	---------------------------	----------------------------	---

Ejemplo:

09/07/2017;09:33;33;5;Disk failure

09/07/2017;09:38;44;9;Out of memory

Por otro lado existirá el archivo services.txt el cual tendrá los nombres de los servicios que se ejecutan en el servidor con el ID de los mismos separados por “;” así como también el e-mail del empleado que brinda el soporte de dicho servicio. Cada línea tendrá el siguiente formato:

ID	Nombre del servicio (32 caracteres max)	Email soporte (64 caracteres max)
----	---	-----------------------------------

Ejemplo:

33;ServicioClientes.exe;juan@soporte.com.ar

44;ScriptCache.exe;pedro@soporte.com.ar

56;ServiceCleanDisk.exe;carlos@soporte.com.ar

El programa tendrá un menú con las opciones:

1. Leer Log
2. Procesar información
3. Mostrar estadísticas

1. Leer Log

Deberá parsear el archivo log.txt y services.txt cargando un array dinámico con las entidades "LogEntry" y otro array con las entidades "Service" descriptas a continuación.

<pre>struct S_LogEntry { char date[11]; char time[6]; int servid; int gravedad; char msg[65]; }; typedef struct S_LogEntry LogEntry;</pre>	<pre>struct S_Service { int id; char name[33]; char email[65]; }; typedef struct S_Service Service;</pre>
--	---

2. Procesar información

Deberá evaluar el campo "Gravedad" con el siguiente criterio:

- Si la gravedad es menor a 3, se descartará el error.
- Si la gravedad es 3, se deberán copiar los mensajes en el archivo warnings.txt, para cualquier servicio menos el de ID 45.
- Si la gravedad tiene un valor entre 4 y 7 (inclusive) se imprimirán por pantalla con el siguiente formato :

Fecha	Hora	Nombre servicio	Mensaje de error	Gravedad
-------	------	-----------------	------------------	----------

- Si la gravedad es mayor a 7, se copiarán los mensajes en el archivo errors.txt.

Cada línea de los archivos **warnings.txt** y **errors.txt** deberán tener el siguiente formato:

Fecha	Hora	Nombre servicio	Mensaje de error	E-mail soporte
-------	------	-----------------	------------------	----------------

Los campos estarán separados por “,”

3. Mostrar estadísticas

Deberá indicar el servicio que más fallos tiene y la cantidad de fallos para los siguientes valores de gravedad:

- Menores a 3
- 3
- 4 y 7 (inclusive)
- Mayores a 7

Nota 0: El código deberá tener comentarios con la documentación de cada una de las funciones y respetar las reglas de estilo de la cátedra.

Nota 1: Se deberá utilizar la biblioteca arrayList.

Nota 2: Cada una de las entidades identificadas deberá contar con su archivo *.h y *.c