

## PROGRAMACIÓN

### (GRADOS EN INGENIERO MECÁNICO, ELÉCTRICO, ELECTRÓNICO INDUSTRIAL y QUÍMICO INDUSTRIAL)

<b>Sesión</b>	14 (Archivos con organización secuencial)	
<b>Temporización</b>	1 hora (no presencial)	
<b>Objetivos formativos</b>	<ul style="list-style-type: none"> <li>• Conocer la sintaxis de C para la manipulación de estructuras de datos en memoria secundaria (archivos): apertura (para lectura y para escritura), lectura y escritura de componentes, cierre. Diferenciar entre archivos de texto y archivos binarios.</li> <li>• Implementar programas modulares en lenguaje de programación C. Identificar y corregir errores sintácticos que surgen durante la codificación.</li> <li>• Resolver problemas sencillos con archivos de texto/binarios, aplicando las operaciones básicas sobre los mismos (añadir componentes detrás del último y lectura secuencial de todos los componentes). Intercambiar información entre diferentes aplicaciones a través de archivos de texto/binarios.</li> <li>• Conocer los algoritmos básicos de clasificación y fusión externas. Aplicarlos a la resolución (diseño e implementación) de sub-problemas de clasificación por diferentes criterios de datos almacenados en archivos con organización secuencial y a la comparación cruzada de los componentes de diferentes archivos con diferentes objetivos.</li> <li>• Diseñar e implementar programas que resuelven problemas de ingeniería usando operaciones abstractas sobre tipos abstractos de datos: representar el modelo de información mediante una combinación de estructuras de datos y construir operaciones complejas mediante técnicas de diseño modular y programación estructurada.</li> <li>• Probar con datos operacionales la correctitud de los módulos y programas desarrollados e identificar y corregir los errores lógicos que surjan.</li> </ul>	
<b>Competencias a desarrollar</b>	<ul style="list-style-type: none"> <li>• RD1: Poseer y comprender conocimientos</li> <li>• RD2: Aplicación de conocimientos</li> <li>• UAL1: Conocimientos básicos de la profesión</li> <li>• UAL3: Capacidad para resolver problemas</li> <li>• UAL6: Trabajo en equipo</li> <li>• FB3: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en la ingeniería.</li> </ul>	X X X X  X
<b>Materiales</b>	Sesiones de teoría 13 y 14 + Bibliografía tema 5 + Internet Algoritmos de clasificación externa (implementados en C) Archivos con datos de prueba IDEs: Dev-C++/Code::Blocks (freeware)	
<b>Tarea</b>	Desarrollar los dos programas propuestos en esta ficha de trabajo y presentar un informe según modelo que se adjunta.	
<b>Fecha de entrega</b>	Siguiendo sesión del Grupo de Trabajo.	

<b>Criterios de éxito</b>	<ul style="list-style-type: none"> <li>• Terminar en el tiempo previsto la tarea.</li> <li>• Demostrar, en una prueba escrita u oral, mediante las respuestas a las preguntas del profesor que ha alcanzado los objetivos formativos.</li> </ul>	
<b>Plan de trabajo</b>	<b>Actividad</b>	<b>Temporización</b>
	Diseño de los sub-algoritmos correspondientes a cada uno de los ejercicios propuestos. Nota: puede simultanear esta actividad con las dos siguientes (para cada ejercicio).	30 mn
	Implementación en lenguaje C de los sub-programas correspondientes a los algoritmos diseñados.	20 mn
	Pruebas: los programas desarrollados serán validados utilizando como mínimo los datos de prueba suministrados. Nota: en caso de detectar errores en esta fase de pruebas, estos deberán ser corregidos modificando el código fuente y/o el algoritmo correspondiente.	5 mn
	Elaboración de la documentación a presentar según modelo adjunto, así como de la respuesta a las cuestiones planteadas en el mismo.	5 mn

## Ejercicios: desarrollo de programas

<b>Ejercicio 1</b>	Una estación meteorológica está monitorizando el tiempo atmosférico mediante un conjunto de sensores que miden la temperatura (en grados centígrados), la presión barométrica (mm de mercurio), la velocidad del viento (m/sg) y la humedad relativa (en %). Construir un programa que muestre en pantalla la información suministrada por los sensores, actualizándose cada 5 segundos. Además presentará los valores máximos y mínimos de la temperatura, así como la media aritmética de las últimas 10 medidas registradas. También presentará un mensaje indicando si la presión atmosférica está aumentando, disminuyendo o se mantiene igual tras las 3 últimas lecturas. Nota: la lectura de los sensores se simulará mediante un archivo binario que contiene el registro meteorológico de un intervalo de tiempo dado.								
<b>Datos de prueba</b>	<p>Archivo de datos (para simular las medidas): <b>meteo.dat</b></p> <p>Estructura del archivo :</p> <table> <tr> <td>Temperatura (°C):</td> <td>real (simple precisión)</td> </tr> <tr> <td>Presión atmosférica (mm de Hg):</td> <td>real (simple precisión)</td> </tr> <tr> <td>Velocidad del viento (m/sg):</td> <td>real (simple precisión)</td> </tr> <tr> <td>Humedad relativa (%):</td> <td>real (simple precisión)</td> </tr> </table> <p>El archivo de prueba contiene 30 registros, permitiendo simular una secuencia de tiempo de 150 sg (2mn 30 sg). Los datos que contiene son:</p> <pre> 1.0 2.0 3.0 4.0 1.1 2.1 3.1 4.1 1.2 2.2 3.2 4.2 1.3 2.3 3.3 4.3 1.4 2.4 3.4 4.4 1.5 2.5 3.5 4.5 1.6 2.6 3.6 4.6 1.7 2.7 3.7 4.7 </pre>	Temperatura (°C):	real (simple precisión)	Presión atmosférica (mm de Hg):	real (simple precisión)	Velocidad del viento (m/sg):	real (simple precisión)	Humedad relativa (%):	real (simple precisión)
Temperatura (°C):	real (simple precisión)								
Presión atmosférica (mm de Hg):	real (simple precisión)								
Velocidad del viento (m/sg):	real (simple precisión)								
Humedad relativa (%):	real (simple precisión)								

```

1.8 2.8 3.8 4.8
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.9 2.9 3.9 4.9
1.8 2.8 3.8 4.8
1.7 2.7 3.7 4.7
1.6 2.6 3.6 4.6
1.5 2.5 3.5 4.5
1.4 2.4 3.4 4.4
1.3 2.3 3.3 4.3
1.2 2.2 3.2 4.2
1.1 2.1 3.1 4.1
1.0 2.0 3.0 4.0

```

#### Ejemplos en diferentes instantes de tiempo:

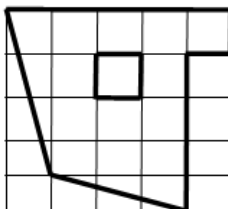
```

t=0 sg          ESTACION METEOROLOGICA
=====
Wed May 31 19:39:50 2006
=====
Temperatura (°c):          1.00
Presion atmosferica (mm de Hg):  2.00
Velocidad del viento (m/sg):    3.00
Humedad relativa (%):        4.00
=====
Evolucion de la temperatura:
    Temperatura maxima:      1.00
    Temperatura minima:     1.00
    Temperatura media (10 u.l.): 1.00
=====
Evolucion de la presion atmosferica:
    Datos de presion insuficientes

t=25 sg          ESTACION METEOROLOGICA
=====
Wed May 31 19:40:15 2006
=====
Temperatura (°c):          1.50
Presion atmosferica (mm de Hg):  2.50
Velocidad del viento (m/sg):    3.50
Humedad relativa (%):        4.50
=====
Evolucion de la temperatura:
    Temperatura maxima:      1.50
    Temperatura minima:     1.00
    Temperatura media (10 u.l.): 1.25
=====
Evolucion de la presion atmosferica:

```

	<p>La presion esta aumentando</p>
t=50 sg	<p>ESTACION METEOROLOGICA</p> <p>=====</p> <p>Wed May 31 19:40:40 2006</p> <p>=====</p> <p>Temperatura (°c): 1.90</p> <p>Presion atmosferica (mm de Hg): 2.90</p> <p>Velocidad del viento (m/sg): 3.90</p> <p>Humedad relativa (%): 4.90</p> <p>=====</p> <p>Evolucion de la temperatura:</p> <p>Temperatura maxima: 1.90</p> <p>Temperatura minima: 1.00</p> <p>Temperatura media (10 u.l.): 1.54</p> <p>=====</p> <p>Evolucion de la presion atmosferica:</p> <p>La presion esta aumentando</p>
t=75 sg	<p>ESTACION METEOROLOGICA</p> <p>=====</p> <p>Wed May 31 19:41:05 2006</p> <p>=====</p> <p>Temperatura (°c): 1.90</p> <p>Presion atmosferica (mm de Hg): 2.90</p> <p>Velocidad del viento (m/sg): 3.90</p> <p>Humedad relativa (%): 4.90</p> <p>=====</p> <p>Evolucion de la temperatura:</p> <p>Temperatura maxima: 1.90</p> <p>Temperatura minima: 1.00</p> <p>Temperatura media (10 u.l.): 1.84</p> <p>=====</p> <p>Evolucion de la presion atmosferica:</p> <p>La presion se mantiene estacionaria</p>
t=100 sg	<p>ESTACION METEOROLOGICA</p> <p>=====</p> <p>Wed May 31 19:41:30 2006</p> <p>=====</p> <p>Temperatura (°c): 1.90</p> <p>Presion atmosferica (mm de Hg): 2.90</p> <p>Velocidad del viento (m/sg): 3.90</p> <p>Humedad relativa (%): 4.90</p> <p>=====</p> <p>Evolucion de la temperatura:</p> <p>Temperatura maxima: 1.90</p> <p>Temperatura minima: 1.00</p> <p>Temperatura media (10 u.l.): 1.90</p> <p>=====</p> <p>Evolucion de la presion atmosferica:</p> <p>La presion se mantiene estacionaria</p>
t=125 sg	<p>ESTACION METEOROLOGICA</p> <p>=====</p> <p>Wed May 31 19:41:55 2006</p>

	<div>=====</div> <div>Temperatura (°c): 1.40</div> <div>Presion atmosferica (mm de Hg): 2.40</div> <div>Velocidad del viento (m/sg): 3.40</div> <div>Humedad relativa (%): 4.40</div> <div>=====</div> <div>Evolucion de la temperatura:</div> <div>Temperatura maxima: 1.90</div> <div>Temperatura minima: 1.00</div> <div>Temperatura media (10 u.l.): 1.75</div> <div>=====</div> <div>Evolucion de la presion atmosferica:</div> <div>la presion esta disminuyendo</div> <div><p><b>Nota:</b> funciones para control del tiempo:</p><pre>#include &lt;time.h&gt; time_t t;  t=time(NULL);          /* Obtiene la hora y fecha actual como */                         /* el n° de segundos transcurridos      */                         /* desde las 0:00 del 1-1-1970            */  printf("%s",ctime(&amp;t)); /* Imprime en pantalla el valor */                         /* de t (tipo time_t) como una        */                         /* cadena de caracteres que             */                         /* incluye la fecha y la hora:          */                         /* Mon Nov 21 11:31:54 1983\n\0        */  while(difftime(time(NULL),t)&lt;5); /* Retardo de 5 segundos */</pre></div>																																											
Ejercicio 2	<div>Considerar los siguientes archivos donde se han almacenado las medidas registradas por un escáner industrial de una pieza plana de mármol en las diferentes cuadrículas de un retículo rectangular que incluye la pieza:</div> <div><div><div>pieza.cfg</div><div>Nº filas: entero</div><div>Nº columnas: entero</div><div>Tamaño cuadrícula (cm): real</div></div><div><div>pieza.dat</div><div>Medida: entero (0,1)</div></div></div> <div><div>Las medidas se han tomado en cuadrículas de 25 cm de lado y los valores posibles son 0 ó 1, correspondiendo el valor 1 a una cuadrícula con mármol de calidad aceptable y el valor 0 a una cuadrícula con material defectuoso (sin material, incompleto, con impurezas,...). Ejemplo:</div><div><div></div><div><table><tr><th colspan="2" rowspan="2"></th><th colspan="5">Nº Columna</th></tr><tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr><tr><th rowspan="5">Nº Fila</th><th>1</th><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><th>2</th><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><th>3</th><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><th>4</th><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><th>5</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table></div></div><div><div>pieza.cfg = {5,5,25,EOF}</div><div>pieza.dat = {0,0,0,0,0,1,1,1,1,0,1,0,1,1,0,1,1,1,0,1,0,0,0,EOF}</div></div><div>El archivo pieza.cfg contiene un solo registro con el nº de filas y de columnas del retículo rectangular con datos (considerar un máximo de 100 en ambas</div></div>			Nº Columna					1	2	3	4	5	Nº Fila	1	0	1	1	1	1	2	0	1	0	1	0	3	0	1	1	1	0	4	0	1	1	1	0	5	0	0	0	0	0
				Nº Columna																																								
		1	2	3	4	5																																						
Nº Fila	1	0	1	1	1	1																																						
	2	0	1	0	1	0																																						
	3	0	1	1	1	0																																						
	4	0	1	1	1	0																																						
	5	0	0	0	0	0																																						

	<p>direcciones), y el archivo pieza.dat contiene las medidas de las diferentes cuadrículas, almacenadas por columnas empezando por la esquina superior izquierda.</p> <p>Construir un programa que dados dichos archivos genere un tercer archivo que contenga todas las piezas cuadradas, de dimensiones múltiplos de 25 cm y en posición estándar, que se pueden extraer de la misma, empezando por las de mayor tamaño posible, y con el siguiente formato:</p> <p style="text-align: center;">Nº fila:            entero  Nº columna:    entero  Ancho:            entero  Largo:            entero</p> <p>Ej:        pieza.ext = {{3,2,3,2},{1,2,4,1},{2,2,1,1},{2,4,1,1},EOF}</p>		
<b>Datos de prueba</b>	<b>Pieza.cfg</b>	<b>Pieza.dat</b>	<b>Pieza.ext</b>
	5 5 25.0	0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0	3 2 3 2 1 2 4 1 2 2 1 1 2 4 1 1
Nota: construir manualmente dos archivos de texto con los datos del ejemplo.			

<b>Asignatura</b>	Programación		
<b>Plan de Estudios</b>	Grados en Ingeniero Mecánico, Eléctrico, Electrónico Industrial y Químico Industrial		
<b>Actividad</b>	Trabajo individual	<b>Sesión</b>	14
<b>Tiempo empleado</b>			

<b>Apellidos, nombre</b>	<b>DNI</b>	<b>Firma</b>

### 1.- Documentación del diseño y de la implementación de los programas desarrollados.

Documentar de forma adecuada los productos de ingeniería obtenidos en las fases principales de desarrollo de los sub-programas de esta práctica: para el diseño utilizar la notación formal de pseudo-código propuesta en las clases de teoría y para la codificación utilizar directamente el listado fuente en lenguaje C.	
<b>Ejercicio</b>	(enunciado del ejercicio)
<b>Diseño preliminar</b>	<u>Archivos de datos externos usados</u> : (nombre físico, estructura y organización) <u>Diseño de datos</u> : (nuevas tipologías) <u>Estructura del programa</u> : (diagrama de módulos) <u>Interfaces entre módulos</u> : (nombre + lista de parámetros formales)
<b>Ejercicio</b>	
<b>Diseño preliminar</b>	
...	

**2.- Resultados de aprendizaje (reflexión):** marque con una cruz los objetivos que cree haber alcanzado tras realizar esta actividad, y rellene en el campo de observaciones aquellos aspectos que cree que necesita mejorar (si los hubiera):

Objetivos formativos	Cumplimiento
<ul style="list-style-type: none"> <li>Conocer la sintaxis de C para la manipulación de estructuras de datos en memoria secundaria (archivos): apertura (para lectura y para escritura), lectura y escritura de componentes, cierre. Diferenciar entre archivos de texto y archivos binarios.</li> <li>Implementar programas modulares en lenguaje de programación C. Identificar y corregir errores sintácticos que surgen durante la codificación.</li> <li>Resolver problemas sencillos con archivos de texto/binarios, aplicando las operaciones básicas sobre los mismos (añadir componentes detrás del último y lectura secuencial de todos los componentes). Intercambiar información entre diferentes aplicaciones a través de archivos de texto/binarios.</li> <li>Conocer los algoritmos básicos de clasificación y fusión externas. Aplicarlos a la resolución (diseño e implementación) de sub-problemas de clasificación por diferentes criterios con datos almacenados en archivos</li> </ul>	

<p>con organización secuencial y a la comparación de los componentes de diferentes archivos con diferentes objetivos.</p> <ul style="list-style-type: none"><li>• Diseñar e implementar programas que resuelven problemas de ingeniería usando operaciones abstractas sobre tipos abstractos de datos: representar el modelo de información mediante una combinación de estructuras de datos y construir operaciones complejas mediante técnicas de diseño modular y programación estructurada.</li><li>• Probar con datos operacionales la correctitud de los módulos y programas desarrollados e identificar y corregir los errores lógicos que surjan.</li></ul>	
<b>Observaciones</b>	