



C.E.S. Cristo Rey

Prácticas Programación

Segundo Trimestre, Curso 2013-2014



Normas Generales

ATENCIÓN: No se corregirá ninguna práctica que no cumpla con la normativa expuesta a continuación.

- 1.- La práctica consiste en la **realización** de los **tres problemas** que se describen más adelante. La resolución de cada problema se realizará independientemente y será implementada en ISO C++. Para la **resolución se podrán utilizar exclusivamente los conocimientos adquiridos en las unidades didácticas 1,2,3,4,5,6 y 7.**
- 2.- Es obligatorio incluir la **documentación** del código, así como de los prototipos de los módulos propuestos para la solución de los problemas.
- 3.- Todo alumno/a deberá traer impresa una **memoria de prácticas** que incluya el enunciado de los problemas y a continuación de cada uno de ellos, el código que lo soluciona **correctamente documentado.**
- 4.- Se tendrá que demostrar el **correcto funcionamiento** de los programas realizados, por lo que el alumnado deberá traer el material necesario para compilar el código utilizando el compilador g++ y un fichero makefile que realice la compilación automática de todos los problemas de la práctica. El nombre que deberá tener cada fichero se indica junto con el problema.
- 5.- En la evaluación de la práctica, se tendrá en cuenta, además de la corrección de la solución propuesta, el estilo de la programación, que la **metodología** seguida sea la explicada en clase y la claridad del código.
- 6.- Se informa al alumnado que **se empleará software para la detección del plagio** durante la evaluación de las prácticas, por lo que en caso de advertirse plagio, se actuará tanto contra quien haya copiado, como contra quien no haya velado por la privacidad de sus soluciones.
- 7.- La **fecha y hora de entrega** de prácticas máxima será el día 26 de Marzo a las 23:59 GMT+1, mediante el envío de un **fichero comprimido llamado practicas2T.zip** en el apartado de entrega habilitado en la plataforma Moodle del CES Cristo Rey, en la asignatura de Programación. Se debe de realizar el envío con el usuario propio de cada alumno.
- 8.- El **examen y defensa de prácticas** se realizará el día 27 de Marzo a partir de las 8.00 GMT+1 y hasta las 14.30 GMT +1 en el aula de 1ºDAM. El alumnado debe estar preparado para responder a cualquier pregunta que se realice en referencia a la práctica entregada.



Descripción del Problema

Nombre fichero a entregar: simulador_sgbdr.cpp

Enunciado: Realizar un simulador de un SGBD Relacional que actuará sobre una única tabla llamada ALUMNOS. El programa deberá mostrar un menú en la consola que permitirá ir accediendo a la distinta funcionalidad que se tendrá que implementar. En el menú debe aparecer el acceso a la siguiente funcionalidad.

BIENVENIDO AL SIMULADOR DE FUNCIONAMIENTO DE UN SGBD RELACIONAL

Pulse la tecla que desee para realizar una de las siguientes acciones:

- 1) Crear tabla Alumnos
- 2) Eliminar tabla Alumnos
- 3) Imprimir tabla Alumnos
- 4) Insertar Alumno en tabla Alumnos
- 5) Eliminar Alumno en tabla Alumnos
- 6) Buscar Alumno por Atributo Nombre en tabla Alumnos (utilizar la búsqueda secuencial)
- 7) Buscar Alumno por Atributo Nota Media en tabla Alumnos (utilizar la búsqueda binaria)
- 8) Ordenar tabla Alumnos por Atributo Nota Media
- 9) Ordenar tabla Alumnos por Código del Alumno
- 10) Proyectar la tabla Alumnos por los Atributos Nombre y Nota Media.

EXTRA SUBIR NOTA: Resolver la siguiente consulta sobre la tabla: "Devolver el conjunto de alumnos alumnos que tienen una nota media mayor de 6.5".

Para la resolución del problema se deberá utilizar memoria dinámica y punteros exclusivamente para todas las estructuras de almacenamiento, ya sean vectores, matrices o estructuras de datos propias. Se deben utilizar las siguientes estructuras como base, así como los módulos pertinentes que permitan gestionarlas apropiadamente:

```
struct Alumno{  
  
    string cod_alumno;  
  
    string DNI;  
  
    string Nombre;  
  
    string Apellido1;  
  
    string Apellido 2;  
  
    string Telefono;  
  
    double NotaMedia;  
  
}
```



```
struct TablaAlumnos{  
    Alumno** punteroapuntero;  
    int totaltuplas;  
}
```

