

## Programación Orientada a Objetos 2021 - Recuperatorio 1er Parcial - 5/10/2021

### Ejercicio 1 (30 pts)

a) Escriba una clase llamada *Cancion* para modelar un tema musical. Debe guardar el nombre de la canción, la duración en segundos y el intérprete. Codifique un constructor y los métodos que considere necesarios.

b) Se desea crear una clase *Playlist* que permita gestionar una lista de canciones. Debe tener un constructor para inicializar el nombre de la playlist, y métodos para: agregar una canción; consultar una canción dada su ubicación en la lista; obtener el número total de canciones; reemplazar de una canción de la lista dado su nombre (recibe un string y una nueva *Cancion*; devuelve true si la encontró y la reemplazó, false si el nombre no estaba en la playlist); y obtener el tiempo total (en segundos) de las canciones de un determinado intérprete que se pasa como parámetro.

### Ejercicio 2 (35 pts)

a) Diseñe clase *Carrera* que permita almacenar y consultar el nombre de la carrera, el promedio histórico de los egresados de la misma (es un solo número real), y la lista de materias del plan. Agregue además un método que reciba un nombre de materia y retorne en qué posición del plan de carrera se encuentra (si es la 1ra materia, la 2da, o cual).

b) A través de alguna relación de clases utilice la clase *Carrera* para diseñar las clases *IngMecanica* y *Arquitectura*. Estas clases deberán tener un método para calcular el promedio de un alumno que la haya completado, a partir de un vector con los datos de todos los sus exámenes (vector de *struct Examen { int fecha; string materia; int nota; }* ).

- En la carrera de Ing. Mecánica, que tiene 42 materias, para calcular el promedio no se toman en cuenta los insuficientes (nota menor a 6), y se deben considerar las últimas 12 materias del plan con peso doble (en el promedio, las notas de estas materias se deben sumar dos veces, y luego se debe dividir por 54 en lugar de 42).

- En Arquitectura se calcula el promedio habitual (sumar una vez cada nota y dividir por la cantidad real), considerando todos los exámenes (incluyendo los insuficientes).

*Nota:* el vector de exámenes que se recibe está ordenado por la fecha en que el alumno rindió, y no por el orden de las materias en el plan.

c) Un alumno puede acceder a una beca de posgrado si su promedio supera al promedio histórico y tiene menos de 3 insuficientes. Escriba una función *beca\_posgrado* que reciba una carrera (debe poder recibir cualquiera de las dos) y un vector de exámenes correspondiente a un alumno, y retorne un booleano indicando si el alumno está en condiciones de acceder a la beca.

### Ejercicio 3 (35 pts)

a) Escriba una función que reciba dos punteros indicando el comienzo y el final de un arreglo de enteros, y un tercer argumento de tipo entero. La función debe retornar verdadero si el valor que indica ese tercer parámetro se encuentra dentro del arreglo.

b) Escriba una función que reciba dos punteros indicando el comienzo y el final de un arreglo de enteros, y genere un nuevo arreglo sin elementos repetidos. c) Escriba un programa cliente que utilice esta función.

*Nota:* para este ejercicio no puede utilizar la clase `std::vector` (debe gestionar los arreglos con `new/delete`), ni el operador `[ ]` (debe utilizar solo notación de punteros).