

## Indicaciones Generales

- Los grupos deberán tener exactamente tres (3) integrantes. En caso de que la cantidad total de estudiantes no sea múltiplo de 3, se admitirá excepcionalmente algún grupo de dos (2) integrantes.
- El trabajo obligatorio consistirá en la resolución del problema presentado más adelante. La fecha de entrega es el **Martes 12 de Diciembre de 2017**. Se debe enviar un mail al docente conteniendo un archivo zippeado con los archivos correspondientes a la entrega. En el mail deben estar debidamente especificados los nombres de todos los integrantes del grupo.
- Se destinará una semana y media del curso para que los estudiantes trabajen en el obligatorio durante el horario de clase, contando con el docente para orientarlos y contestar sus dudas. No obstante, es importante tener presente que los grupos deberán destinar tiempo **fuera** del horario de clase a efectos de completar satisfactoriamente el trabajo.

## Objetivo

- Aplicación de diversos conocimientos adquiridos en todo el curso y trabajo en grupo mediante la implementación de la siguiente situación.

## Planteo del Problema

La **S.M.U.R.F** (*Sociedad de Magos Unidos Recontra Fántasticos*) necesita un programa para registrar los datos de sus socios junto con las habilidades fantásticas que ellos manifiestan.

Habiendo transcurrido varios milenios desde su fundación, y considerando que muchos de los socios son inmortales o viven durante bastantes siglos, la sociedad cuenta con una cantidad de socios cada vez más elevada y requiere con urgencia la informatización. Cada socio está identificado por su cédula mágica, interesando registrar además su nombre, dimensión de origen, fecha de nacimiento y categoría mágica (que puede ser *brujo*, *hada* o *hechicero*).

Por ejemplo, el brujo *Gagarmel* tiene la cédula mágica 6661313, nació el 13 de agosto de 1213 y proviene de la dimensión de *Pitufilondria*. Se quiere almacenar a los socios ordenados por cédula mágica dentro del programa.

También interesa mantener un registro de las habilidades fantásticas de los socios, de las que se conoce el nombre de la habilidad, la fecha de manifestación, la cédula mágica del socio a la que pertenece y el tipo de habilidad, que puede ser *natural*, *poco natural* o *sobrenatural*.

Las habilidades de todos los socios se quieren mantener juntas y en orden cronológico inverso (es decir, primero las más recientes y luego más antiguas) sin importar a qué socio pertenezca cada una. Cada nueva habilidad se ingresa al programa en el momento en que algún socio la manifiesta por primera vez, por lo que cada nuevo registro de habilidad tendrá una fecha de manifestación igual o posterior a la de la última habilidad ingresada. Al igual que con la cantidad de socios, no existe ninguna cota para la cantidad de habilidades.



Puede suceder que haya socios ingresados que aún no han manifestado ninguna habilidad, como también puede ocurrir que se ingresen varias habilidades diferentes en una misma fecha. Incluso puede llegar a pasar (aunque no es muy frecuente) que un mismo socio manifieste la misma habilidad en fechas diferentes a lo largo del tiempo. Por ejemplo, el brujo *Gagarmel* manifestó la *temura* (habilidad poco natural) en dos fechas diferentes, lo cual causó un enorme revuelo dentro de la sociedad. La primera vez fue el 13/08/1214, en ocasión de la torta de musgo que su mamá le hizo por su primer cumpleaños, y la segunda vez fue el 21/06/1523 cuando, en ocasión del solsticio de verano en la europa medieval, esbozó una leve sonrisa al darle vida a *Pitufea*.

En base a la información anterior, se desea que el sistema resuelva los siguientes requerimientos:

### **Registro de datos**

- Ingresar los datos de un nuevo socio. Se debe verificar previamente que no exista otro socio registrado con la misma cédula mágica y que su fecha de nacimiento sea válida.
- Ingresar una nueva habilidad. Se debe verificar previamente que la cédula mágica del socio que la manifiesta esté registrada y que la fecha de manifestación sea válida y mayor o igual a la fecha de la habilidad más reciente registrada hasta el momento.

### **Consultas**

- Contar cuántos socios de cada categoría mágica (brujos, hadas y hechiceros) hay registrados en el sistema.
- Contar cuántas habilidades de cada tipo (naturales, poco naturales y sobrenaturales) hay registradas en el sistema.
- Dada una fecha de nacimiento, contar cuántos de los socios registrados en el sistema nacieron antes de dicha fecha. Se debe verificar previamente que la fecha ingresada sea válida.
- Dadas dos fechas, contar cuántas habilidades fueron ingresadas dentro de dicho rango de fechas (incluidas ambas fechas). Se debe verificar previamente que ambas fechas ingresadas sean válidas y que la primera fecha sea menor o igual que la segunda.

### **Listados**

- Listar todos los socios registrados en el sistema, ordenados por cédula de menor a mayor. Los datos de un mismo socio (cédula mágica, nombre, fecha de nacimiento, dimensión de origen y categoría mágica) deberán listarse todos en una misma línea.
- Listar todos aquellos socios registrados en el sistema que aún no han manifestado ninguna habilidad, ordenados por cédula de menor a mayor. Los datos de un mismo socio (cédula mágica, nombre, fecha de nacimiento, dimensión de origen y categoría mágica) deberán listarse todos en una misma línea.
- Dada la cédula mágica de un socio, listar todas las habilidades correspondientes a dicho socio, ordenadas en forma cronológica inversa (desde las más recientes hacia las más antiguas). Los datos de una misma habilidad (nombre, fecha de manifestación, cédula mágica del socio y tipo de habilidad) deberán listarse todos en una misma línea. Se debe verificar previamente que el socio correspondiente exista en el sistema.
- Dada una fecha, listar todas las habilidades que hayan sido manifestadas en dicha fecha. Los datos de una misma habilidad (nombre, fecha de manifestación, cédula mágica del socio y tipo de habilidad) deberán listarse todos en una misma línea. Se debe verificar previamente que la fecha ingresada sea válida.
- Listar los datos del socio más poderoso registrado en el sistema. Es decir, del socio que ha manifestado la mayor cantidad de habilidades sobrenaturales de entre todos los socios registrados. En caso de que haya varios, basta con devolver uno cualquiera de ellos. En caso de que no haya ninguno, emitir un mensaje que así lo indique.

Los datos manejados por el programa deberán permanecer respaldados en dos archivos en el disco cuando el programa no esté en ejecución. El archivo "Socios.txt" se utilizará para respaldar los datos de los socios, mientras que el archivo "Habilidades.txt" se utilizará para respaldar los datos de las habilidades. Ambos archivos se crearán automáticamente por el programa la primera vez que se ejecute. Cada vez que el programa inicie su ejecución, deberá recuperar los datos de los archivos y cargarlos a memoria. Cada vez que el programa finalice su ejecución, deberá respaldar nuevamente los datos en los archivos. Durante la ejecución se trabajará **exclusivamente** con los datos en memoria.

**Observación:**

- Por cuestiones prácticas, vamos a suponer que los datos manejados por el programa caben en memoria mientras que está en ejecución. Sin embargo, tener presente que en un programa real los datos pueden ser muy voluminosos para estar todos en memoria simultáneamente y será necesario definir otras estrategias para garantizar que sean bien manejados. No obstante, **no** abordaremos estos temas pues escapan al alcance del curso.

**Se pide:**

- a) Pensar cuáles son las estructuras de datos más adecuadas para representar los elementos de la realidad propuesta y escribir en C++ todos los tipos de datos correspondientes.
- b) A partir de los tipos de datos definidos, dibujar el esquema que ilustre la jerarquía de módulos necesaria para el sistema, definiendo cuidadosamente las inclusiones necesarias entre los distintos módulos.
- c) Escribir en C++ los archivos **.h** correspondientes a cada uno de los módulos anteriores, incluyendo en cada uno la definición del tipo correspondiente junto con los encabezados de las operaciones del módulo. Para cada operación, definir cuidadosamente si corresponde implementarla como función o procedimiento. Detallar además toda **precondición** que sea necesaria. Analizar cada requerimiento con detenimiento y definir los encabezados sintácticos de todas las operaciones **antes** de sentarse a programar.
- d) Implementar en C++ los archivos **.cpp** correspondientes a los módulos definidos. Empiece por los módulos más simples y luego siga con los módulos más complejos. Si bien **no** se exige como parte de la entrega, se recomienda hacer un programa de prueba por cada módulo implementado **antes** de pasar al siguiente módulo. **No** intente probar el programa final todo de una sola vez. Probablemente tendrá muchos errores que podrían haberse evitado probando cada módulo por separado y seguramente no le dará el tiempo para corregirlos antes de la entrega.
- e) Programar en C++ el programa principal (main) que invoque a las operaciones de los módulos anteriores a efectos de dar solución a todos los requerimientos solicitados. Hacer uso de menús de opciones desde el main (recordar definir un módulo adicional para los menús de opciones, e incluirlo en el diagrama de la parte b).

**Observación:** Definir cuidadosamente en qué módulo corresponde definir cada operación del sistema. Utilizar todas aquellas operaciones auxiliares de otros módulos que sean necesarias en cada caso a efectos de lograr una correcta distribución del trabajo entre los diferentes módulos.