

#### Pregunta 4:

Un club de socios tiene una sala de juego con acceso restringido. Cuenta con un número muy limitado de máquinas recreativas que representan el *SalaJuego::AforoMaximo* de la sala. No está permitido que más personas que el aforo limitado entren en dicha sala. En ocasiones este aforo podría superarse por lo que se ha establecido un sistema de control para evitarlo, controlando también otros aspectos ajenos a este problema.

Si el aforo no se supera, cualquier miembro del club podría entrar. Para ello su móvil tiene una aplicación donde muestra un símbolo a modo de QR-code que lo identifica y le permite entrar con *Socio::entrar()*, donde se registra la hora de entrada (*Socio::horaEntrada*) con la horaActual del sistema (*SalaJuegos::horaActual*, siempre actualizada). Cuando el cliente sale por la puerta, automáticamente se registra su hora de salida (*Socio::horaSalida*) y el aforo del sistema se decrementa (*SalaJuegos::decrementaAforo()*). A dicho socio se le acumula entonces el tiempo dentro de la sala con *Socio::tAcumulado* (la clase Hora dispone de los operadores que se deseen. La resta devuelve minutos).

Si el aforo se supera, entonces la petición del cliente queda registrada en una cola de espera. Esta cola permitirá entrar en primer lugar a los miembros VIP por orden de llegada (*Socio::vip=true*). En segundo lugar el resto de miembros que no hayan superado el tiempo máximo en el local (*Socio::TMaximo*) para así limitar algo el acceso a los potenciales ludópatas. Finalmente, si el aforo lo permite entrarán el resto de los miembros del club, aunque hayan sobrepasado dicho tiempo máximo actualizado mensualmente.

Cuando alguien sale de la sala *Socio::salir()* se permite la entrada al socio más prioritario. Con *SalaJuego::llamadaASocio()* se le manda un mensaje que le aparece en pantalla (usar *cout << mensaje*) indicando que ya puede pasar (*Socio::mensaje()*). La petición correspondiente se pone a *Peticion::petConcedida=true*. El socio puede ahora volver a pasar por la puerta con *Socio::entrar()*, devolviendo ahora un *true* la llamada a *SalaJuego::peticionEntrada()*. Como no es seguro que el socio quiera volver a entrar (puede haberse ido a casa o a otro establecimiento), el sistema comprueba automáticamente cada 5 minutos el aforo con *SalaJuegos::comprobarAforo()*; si pasado este tiempo la petición más prioritaria está a *true*, es decir, *SalaJuegos::masPrioritaria().petConcedida==true*, significa que el socio no ha entrado, así que se elimina *SalaJuego::eliminaMasPrioritaria()* y se llama al siguiente. Por el contrario, si el socio ha entrado, la petición se elimina definitivamente de la cola.

Se pide la implementación de las tres clases del esquema. La gestión de peticiones debería realizarse con la estructura o estructuras de datos más convenientes para tener acceso en  $O(1)$ .

