

Algunas aclaraciones:

- El archivo de clases tiene el id de la clase, el nombre y la hora. Cada clase, dependiendo de la hora a la que se dicte, tiene un id único. Dentro de este mismo archivo también están las clases de musculación, con sus respectivos id y horario.

por ejemplo:

```
idClase,nombre,horario
1,Spinning,8
2,Spinning,10
3,Spinning,12
...
32,Boxeo,18
33,Boxeo,19
34,Musculacion,7
35,Musculacion,7.30
...
```

- La estructura del archivo binario es la misma. Puede ser que haya clientes que estén anotados en la misma clase 2 veces, eso es algo que uds tienen que contemplar y filtrar, también deberían chequear antes de ingresar un cliente a una clase que éste ya no fué inscripto.

por ejemplo: (Esta es una representación de como está estructurado el archivo binario)

```
idCliente,cantInscriptos,{Inscripción(idCurso, fechaInscripción)}
20, 3, {(12, 109029092), (12, 109029323), (30, 1032093211)} <-- este cliente se
repite en la misma clase (12) y no lo deberían contar como parte del cupo actual.
23, 1, {(42, 132322131)}
34, 2, {(1, 1231242321), (4, 1032010035)}
```

- `{}` <- significa que es un array
- `()` <- es la estructura Inscripción
- El registro es el timestamp del horario en que se anotó, no el horario ni la fecha a la cual se anotó.

Por último, el archivo binario se escribió de la siguiente forma

```
ofstream archibinwr("asistencias_1697673600000.dat", ios::binary);
if (archibinwr.is_open()) {
    for (int i=0; i<cantAsistencias; i++) {
        archibinwr.write((char*)&Asistencias[i].idCliente, sizeof(uint));
        archibinwr.write((char*)&Asistencias[i].cantInscriptos, sizeof(uint));

        for(uint j = 0; j < Asistencias[i].cantInscriptos; j++) {
            archibinwr.write((char*)&Asistencias[i].CursosInscriptos[j],
```

```
sizeof(Inscripcion));  
    }  
}  
archibinwr.close();
```

Esto es necesario que lo sepan para poder leerlo.