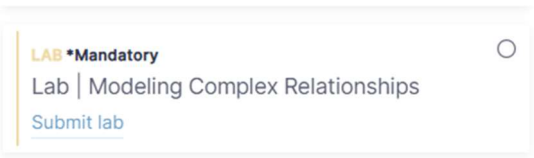
LAB Modeling Complex Relationships



Casos de uso

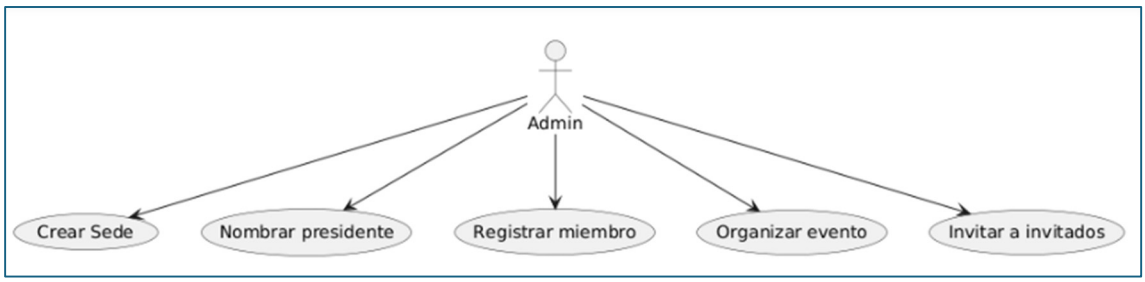
A continuación se exponen los actores principales de la aplicación Event management y sus casos de uso principales.

Actores principales

- Admin
- Member
- Guest

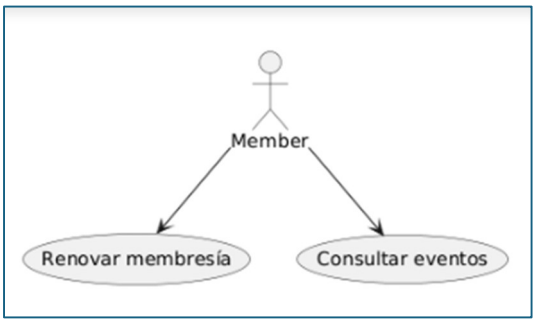
Admin

- Crear una Sede.
- Establecer un presidente
- Registrar miembros
- Organizar una evento
- Invitar a invitados al evento



Member:

- Renovar membresía
- Consultar próximos eventos.



Guest:

- Confirmar asistencia a un evento

UML Lab Modeling Complex Relationships

Elección del diseño

Se decide tener una super clase *'Event'* de la cual heredan los eventos que puedan suceder. Actualmente y, según el enunciado, constan dos posibles eventos: conferencias y exposiciones.

Con ánimo de hacer un diseño desacoplado y escalable (en un futuro podrían existir más eventos o dejar de realizar otros...) se decide diseñar una clase abstracta de la cual se especializarán los diferentes eventos que puedan ir teniendo lugar.

En el diseño propuesto se observan las siguientes relaciones:

- Un evento tiene muchos invitados (OneToMany)
- Una conferencia puede tener muchos ponentes (OneToMany)
- Las secciones/sedes tienen un presidente, que a su vez es un miembro (OneToOne)
- Las secciones/sedes tienen muchos miembros (OneToMany)

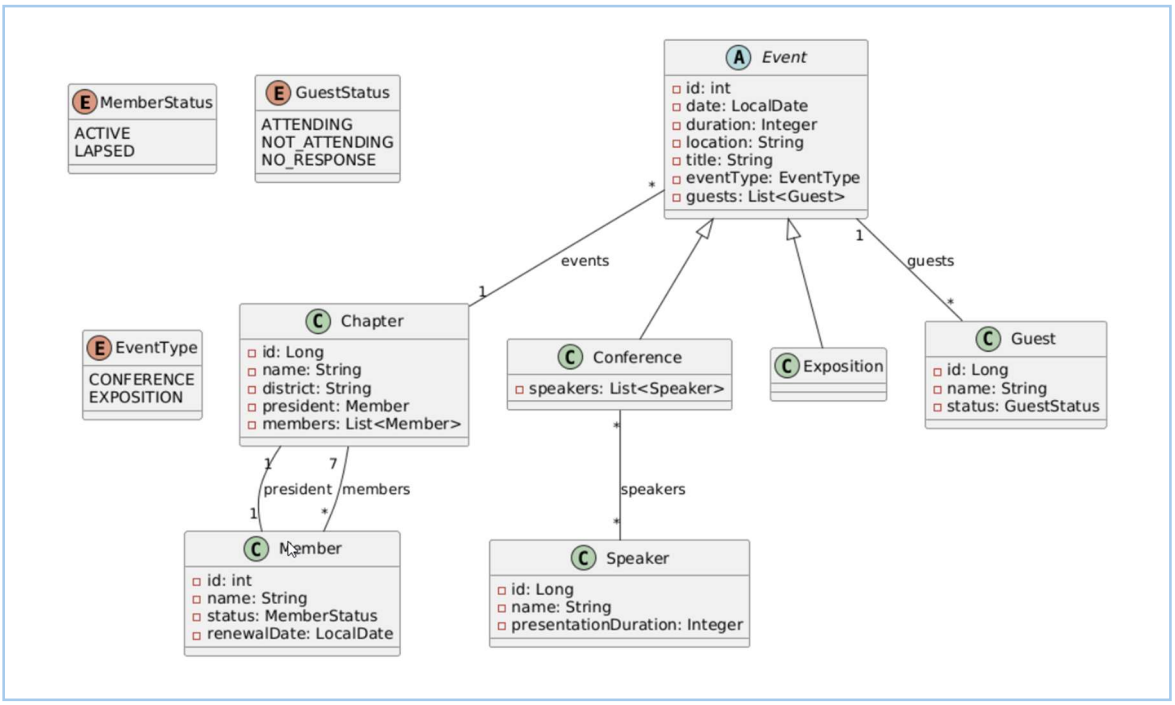
Muchos eventos pueden ser organizados por una misma sede en ciertas localizaciones como pueden ser salones de actos/espacios de conferencias, hoteles espacios expositivos/galerías....

El lugar de encuentro para llevar a cabo el evento se conoce como *'location'*

Conferencias:

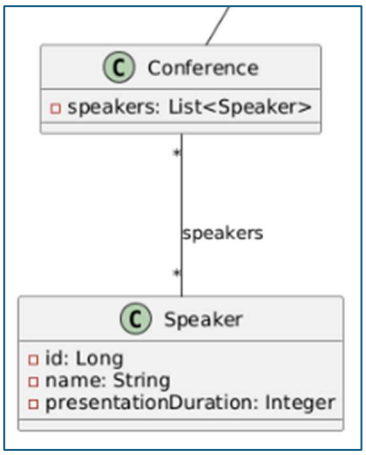
- Una Conference puede tener muchos Speaker.
- Cada Speaker podría volver a dar una conferencia, como consecuente se considera una relación de muchos a muchos (ManyToMany)

Diagrama realizado con PlantUml



Inicialmente la relación entre **Speaker y Conference** era de muchos a uno **ManyToOne**

Esto implica que, si un Speaker ha participado en una conferencia, no va a poder participar en otra próxima. Desde un planteamiento real y práctico, un Speaker que haya gustado a la audiencia, podría volver a ser contratado para dar una nueva charla en una conferencia futura. Esto conlleva modelar la relación de **Speakers – Cenference** a **Many To Many**



Un registro de una tabla (Conference) puede estar relacionado con muchos registros de otra tabla (Speaker).

Un registro de la segunda tabla (Speaker) también puede estar relacionado con muchos registros de la primera (Conference).

Se necesita una tabla intermedia (tabla Join) para representar la relación *conference_speaker*

Esquema de la Base de Datos

El esquema de tablas modelado a través de JPA DDL es el siguiente: