

Diagrama de casos de uso:

@startuml

left to right direction

actor Administrador

actor Miembro

actor Invitado

package "Asociación de Enfermeras" {

 usecase "Actualizar membresía"

 usecase "Asignar presidente"

 usecase "Consultar miembros del capítulo"

}

package "Gestión de Eventos" {

 usecase "Crear evento"

 usecase "Invitar participantes"

 usecase "Responder invitación"

 usecase "Añadir oradores"

}

Administrador --> "Actualizar membresía"

Administrador --> "Asignar presidente"

Administrador --> "Consultar miembros del capítulo"

Administrador --> "Crear evento"

Administrador --> "Invitar participantes"

Administrador --> "Añadir oradores"

Miembro --> "Actualizar membresía"

Miembro --> "Consultar miembros del capítulo"

Invitado --> "Responder invitación"

@enduml

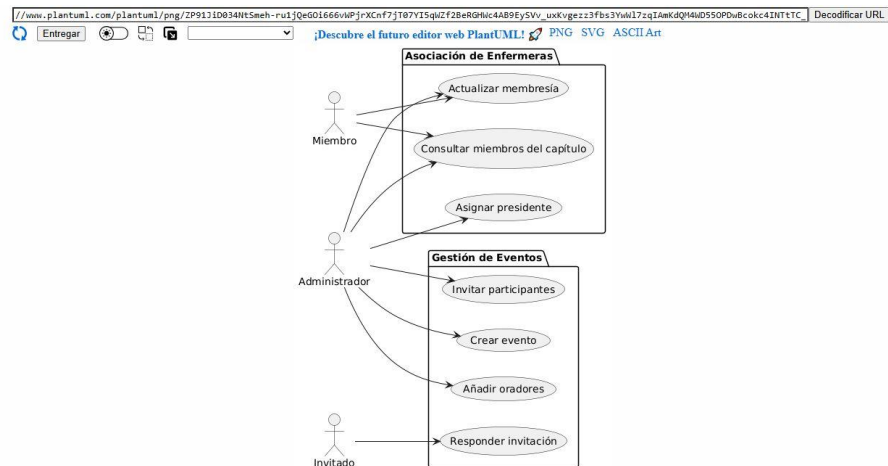


Diagrama de clases UML:

@startuml

skinparam classAttributeIconSize 0

'===== Asociación de Enfermeras =====

package "Asociación de Enfermeras" {

class Chapter {

- id: Long

- name: String

- district: String

- president: Member

- members: List<Member>

}

class Member {

- id: Long

- name: String

- status: MemberStatus

- renewalDate: LocalDate

}

enum MemberStatus {

ACTIVO

CADUCADO

}

Chapter "1" --> "1" Member : president

Chapter "1" --> "*" Member : members

}

'===== Gestión de Eventos =====

package "Gestión de Eventos" {

abstract class Event {

- id: Long

- date: LocalDate

- duration: int

- location: String

- title: String

- guests: List<Guest>

}

```

class Conference {
    - speakers: List<Speaker>
}

class Exposition

class Guest {
    - id: Long
    - name: String
    - status: GuestStatus
}

enum GuestStatus {
    ATTENDING
    NOT_ATTENDING
    NO_RESPONSE
}

class Speaker {
    - id: Long
    - name: String
    - presentationDuration: int
}

Event <|-- Conference
Event <|-- Exposition
Event "1" --> "*" Guest
Conference "1" --> "*" Speaker
}

@enduml

```

