

Diagrama de Casos de Uso (Asociación de Enfermeras)

```
@startuml
left to right direction
actor Miembro
actor Administrador

rectangle "Asociación de Enfermeras" {
    Miembro -- (Ver información del capítulo)
    Miembro -- (Actualizar información personal)
    Miembro -- (Renovar membresía)
    Administrador -- (Crear nuevo capítulo)
    Administrador -- (Editar información del capítulo)
    Administrador -- (Gestionar miembros del capítulo)
    Administrador -- (Asignar presidente del capítulo)
}
@enduml
```

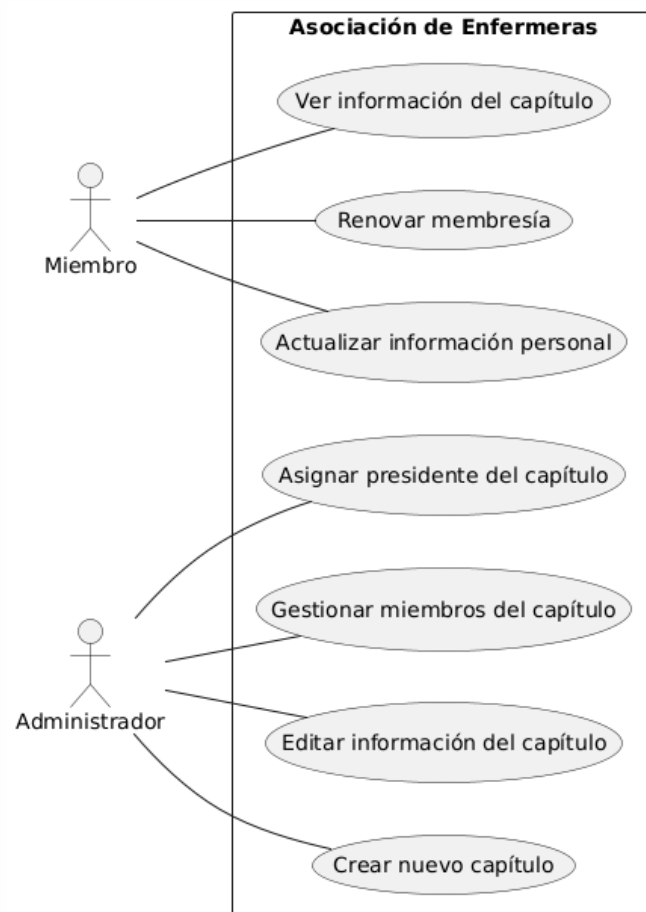


Diagrama de Clases UML (Asociación de Enfermeras)

```

}startuml
class Chapter {
    - Long id
    - String name
    - String district
    - Member president
    - List<Member> members
    + Chapter(name, district, president, members)
    + getId(): Long
    + getName(): String
    + getDistrict(): String
    + getPresident(): Member
    + getMembers(): List<Member>
    + setPresident(president: Member): void
    + addMember(member: Member): void
}

class Member {
    - Long id
    - String name
    - String status
    - LocalDate renewalDate
    - Chapter chapter
    + Member(name, status, renewalDate)
    + getId(): Long
    + getName(): String
    + getStatus(): String
    + getRenewalDate(): LocalDate
    + getChapter(): Chapter
    + setChapter(chapter: Chapter): void
}

```

```

+ getStatus(): String
+ getRenewalDate(): LocalDate
+ getChapter(): Chapter
+ setChapter(chapter: Chapter): void

```

```

chapter "1" -- "*" Member : members
chapter "1" -- "1" Member : president

```

```

member --o Chapter : chapter
enduml

```

www.plantuml.com/plantuml/png/bL8B3eH048pp5HrEhC9XhyIOJToBYHhZV178T5OI7Xkwq0FntsscQPWifprDJrLrKIcdGEm67jicgHf714zVxi8um6y2KC

PREVIEW FOR ENVIAT

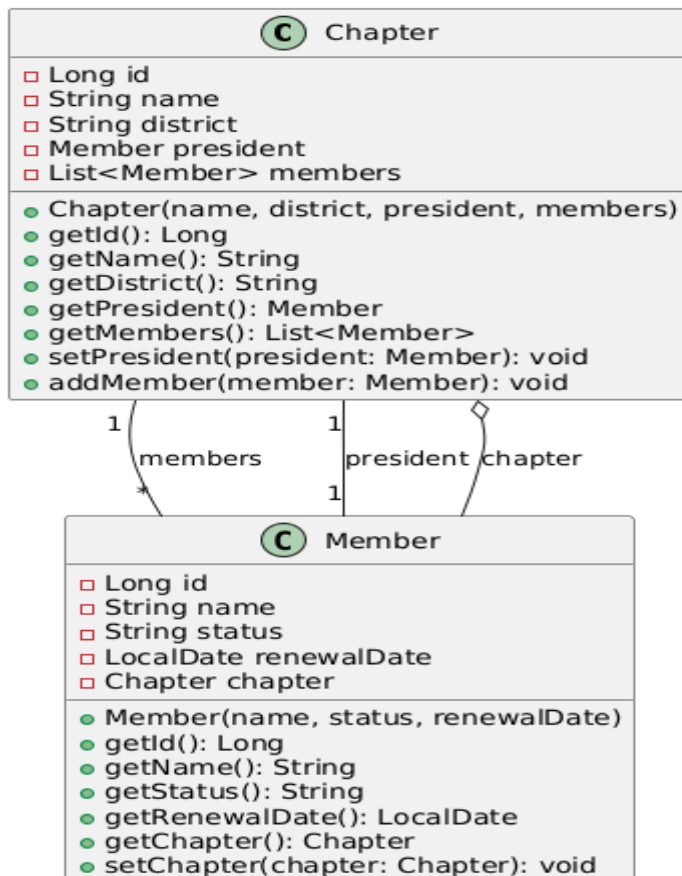


Diagrama de Casos de Uso (Gestión de Eventos)

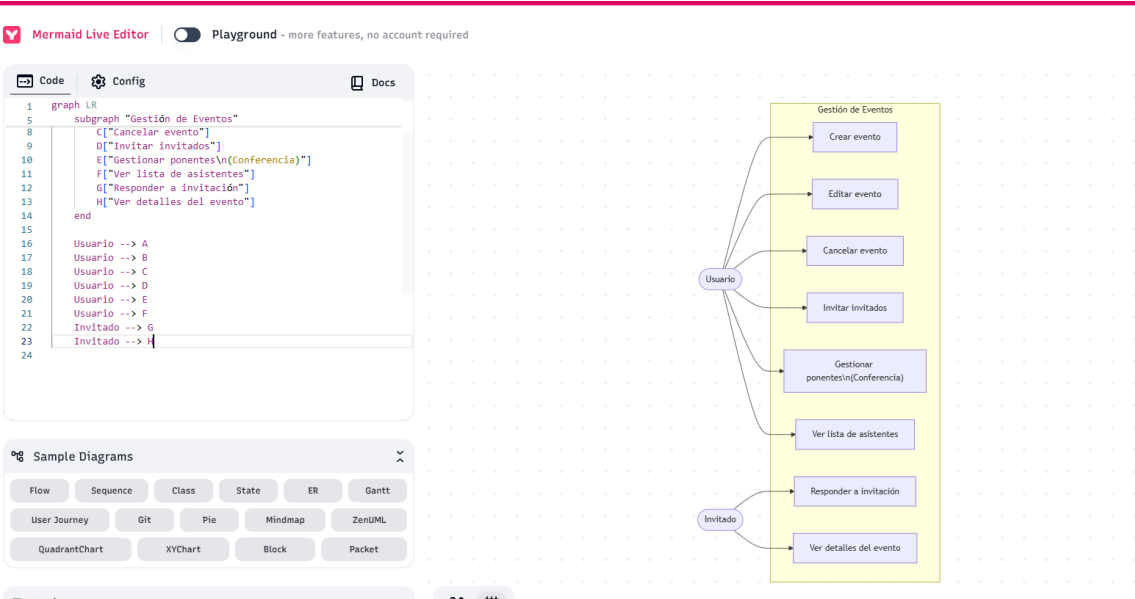


Diagrama de Clases UML (Gestión de Eventos)

```
@startuml
class Event {
    - Long id
    - LocalDate date
    - String duration
    - String location
    - String title
    - List<Guest> guests
    + Event(date, duration, location, title, guests)
    + getId(): Long
    + getDate(): LocalDate
    + getDuration(): String
    + getLocation(): String
    + getTitle(): String
    + getGuests(): List<Guest>
    + addGuest(guest: Guest): void
}
abstract class Conference extends Event {
    - List<Speaker> speakers
    + Conference(date, duration, location, title, guests, speakers)
    + getSpeakers(): List<Speaker>
    + addSpeaker(speaker: Speaker): void
}
class Exposition extends Event {
    + Exposition(date, duration, location, title, guests)
}

class Guest {
    - Long id
    - String name
    - String status
    - List<Event> events
    + Guest(name, status)
    + getId(): Long
    + getName(): String
    + getStatus(): String
    + getEvents(): List<Event>
}

class Speaker {
    - Long id
    - String name
    - String presentationDuration
    - List<Conference> conferences
    + Speaker(name, presentationDuration)
    + getId(): Long
    + getName(): String
    + getPresentationDuration(): String
    + getConferences(): List<Conference>
}
```

Event "1" -- "*" Guest : guests
 Guest "*" -- "*" Event : events

Conference --|> Event : extends
 Conference "1" -- "*" Speaker : speakers
 Speaker "*" -- "*" Conference : conferences

Exposition --|> Event : extends
 @endum1

