Actividad 1, Extracción De Clases De Análisis Con Enfoque Orientado a Objetos

Miguel De Jesus Chavez

Metodologías, Desarrollo Y Calidad En la Ingeniería De Software

Bárbaro Jorge Ferrero Castro

Mayo 26, 2025

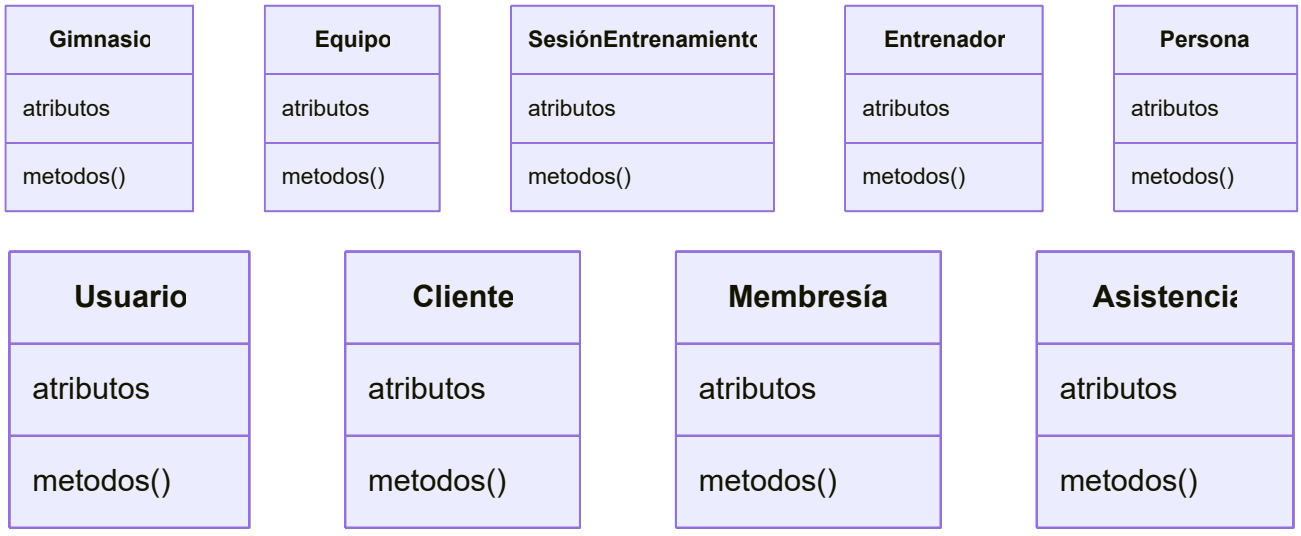
Unir – Universidad Internacional De la Rioja

× indice

Introducción

Este trabajo presentara un análisis orientado a objetos de una aplicación de gestión de gimnasio, describiendo las clases principales del sistema por ejemplo, *Gimnasio*, *Equipo*, *Clase*, *Entrenador*, *Usuario*, *Cliente*, *Membresía*, *Asistencia* con sus atributos y métodos.

Se identifican las relaciones entre estas clases (asociación, herencia, agregación y composición) y se justifica el uso de cada una conforme a las reglas de UML. Se incluye además varios diagrams de clases UML que sintetiza la estructura que se propone para el sistema.



Partiendo de las necesidades que nos comparte el cliente y gestionando los casos de usos podemos analizar los siguientes casos de uso que tienen los actores para llevar acabo sus actividades diversas.

1. **Gestion de usuarios y acceso**
   - **Registrar Nuevo usuario**: El sistema debe permitir dar de alta nuevos clientes y personal.
   - **Autentificar usuario**: Control de acceso mediante credenciales.
   - **Gestionar perfiles**: El sistema de permitir la actualización de información tanto metas, como personal.
2. **Administración de instalaciones**
   - **Abrir/cerrar gimnasio**: Debemos de enter un control de quien se encarga de cerrar y abir el gimnasio
   - **Gestionar Espacios:** Asignación de espacios disponibles con equipos asignados a esa area
   - **monitorear aforo:** Control de Capacidad de personas en instalaciones para el balanceo de entrenadores por horarios de demanda
3. **Control de equipo**
   - **Inventariar equipos**: permite que un empleado pueda añadir equipo nuevo.
   - **Programar mantenimiento**: Calendarización de revisiones y reparaciones.
   - **Verificar disponibilidad**: Consulta del estado actual de los equipos.
4. **Programación de actividades**
   - **Crear Classes Grupales**: Permitir entrenador crear una clase ejemplo de yoga etc.
   - **Inscribirse**: Permite que clientes pueden ingresar a las Classes.
   - **Cancelar / Reprogramar sesiones**: permite que los usuarios cancelen su inscripción a una clase o que entrenador reagende enviando notificación previa.
   - **Notificación de cambios**: Notificar de cancelaciones, o reprogramaciones de clases.
5. **Gestion de membresías**
   - **Suscribir clientes**: Registro de nuevas membresías.
   - **Renovar planes**: Extensión de membresías existentes.
   - **Consultar beneficios**: Verificación de servicios incluidos.
6. **Control de asistencia**
   - **Registrar entrada/salida**: Seguimiento de visitas al gimnasio.
   - **Controlar participación en clases**: Registro de asistentes a sesiones grupales.
7. **Entrenamiento personalizado**
   - **Asignar rutinas**: Creación de planes de entrenamiento individuales.
   - **Realizar seguimiento**: Monitoreo del progreso de los clientes.
   - **Programar sesiones personales**: Organización de entrenamientos uno a uno.

Análisis De Actores Y Entidades Del Sistema De Gestión De Gimnasio

Los actores son los usuarios que interactúan directamente con el sistema y tienen roles específicos. A partir del análisis de los casos de uso, podemos identificar los siguientes

actores:

1. Cliente/Miembro

Actor principal que utilize los servicios del gimnasio.

- **Características**: Persona que ha contratado una membresía para usar las instalaciones.
- **Responsabilidades**: Inscribirse en clases, registrar entrada/salida, consultar su historial de asistencia, renovar membresías.

2. Entrenador

Personal especializado que imparte clases y asesora a los clientes.

- **Características**: Professional con experiencia en actividades físicas y entrenamiento.
- **Responsabilidades**: Crear clases grupales, asignar rutinas personalizadas, gestionar la programación de sesiones.

3. Administrador/Recepcionista

Personal encargado de la gestión operativa del gimnasio.

- **Características**: Empleado con permisos para administrar usuarios y recursos.
- **Responsabilidades**: Registrar nuevos usuarios, gestionar membresías, controlar el acceso, supervisar instalaciones.

4. Técnico De Mantenimiento

Personal especializado en el cuidado de los equipos.

- **Características**: Empleado técnico responsible del estado de las máquinas.
- **Responsabilidades**: Programar y realizar mantenimientos, actualizar el estado de los equipos.

Las entidades representan los objetos principales que son gestionados por el sistema:

1. Gimnasio

- **Descripción**: Representa la instalación física donde se desarrollan las actividades.
- **Relevancia**: Es la entidad contenedora principal que coordina todas las operaciones.
- **Justificación**: Necesaria para gestionar horarios de apertura/cierre, espacios y aforo.

2. Usuario

- **Descripción**: Representa a cualquier persona registrada en el sistema.
- **Relevancia**: Clase abstracta que define atributos comunes para clientes y personal.

- **Justificación**: Permite unificar la gestión de accesos y datos personales.

3. Cliente

- **Descripción**: Especialización de Usuario que representa a los miembros del gimnasio.
- **Relevancia**: Principal usuario de los servicios ofrecidos.
- **Justificación**: Necesaria para gestionar membresías, asistencias e inscripciones.

4. Entrenador

- **Descripción**: Especialización de Usuario que representa al personal instructor.
- **Relevancia**: Responsible de las actividades físicas dirigidas.
- **Justificación**: Necesaria para asignar responsabilidades de clases y entrenamientos.

5. Membresía

- **Descripción**: Representa la formade iniciar y renovar un plan de suscripción.
- **Relevancia**: Define el nivel de acceso y beneficios de cada cliente.
- **Justificación**: Necesaria para la renovcacion de control de acceso.

6. Clase/Sesión

- **Descripción**: Representa una actividad grupal programada.
- **Relevancia**: Organiza las actividades colectivas del gimnasio.
- **Justificación**: Necesaria para la programación, asistencia y asignación de recursos.

7. Equipo

- **Descripción**: Representa las máquinas y material deportivo disponible.
- **Relevancia**: Recurso físico fundamental para las actividades.
- **Justificación**: Necesaria para inventario, mantenimiento y asignación a espacios.

8. Asistencia

- **Descripción**: Representa el registro histórico de participación.
- **Relevancia**: Permite seguimiento de la actividad de los clientes.
- **Justificación**: Necesaria para análisis de uso, facturación y control de acceso.

Clases Del Sistema

A continuación se detallan las clases principales del sistema de gimnasio, indicando para cada una su nombre, propósito, atributos y métodos.

Gimnasio

Propongo adicionalmente la classe de Gimnasio. La cual representa la entididad principal del sistema o las instalciones. Lo cual el modelo se centra en el manego de los usarios, p

Atributos principales:

- `nombre:`
  Nombre official del gimnasio.
- `ubicacion:`
  Dirección física o localización del gimnasio.
- `horario:`
  Horario de apertura y cierre, o bien, estructura de turnos de funcionamiento.

Métodos clave:

- `abrir():`
  Cambia el estado del gimnasio a abierto, permitiendo el acceso a usuarios y el inicio de actividades.
- `cerrar():`
  Cambia el estado del gimnasio a cerrado, restringiendo el acceso y finalizando las operaciones del día.
- `registrarEntrada():`
  Registra la entrada de un usuario al gimnasio, validando su membresía y actualizando el historial de acceso.
- `registrarSalida():`
  Registra la salida de un usuario, permitiendo llevar un control de aforo y asistencia.

Equipo

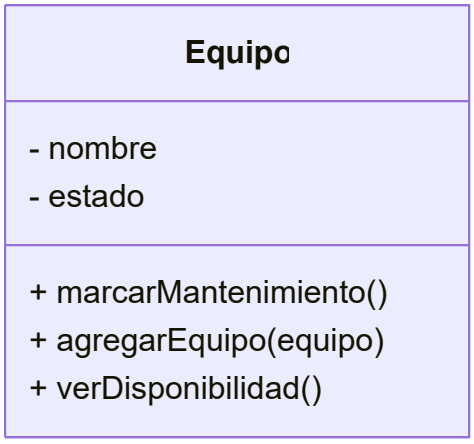
La clase **Equipo** representra cualquier máquina o aparato disponible que esta en el gimnasio, como cintas de correr, bicicletas, pesas, entre otros. Esta entidad es ayudara a la gestión de inventario, el control de disponibilidad y la programación de mantenimientos.

Atributos principales:

- `nombre`
  Descripción o denominación del equipo.
- `estado`
  Indica la condición actual del equipo (por ejemplo: operativo, en mantenimiento, fuera de servicio).

Métodos clave:

- `marcarMantenimiento()`
  Cambia el estado del equipo para indicar que require revisión o reparación, y registra la fecha del mantenimiento.
- `agregarEquipo(equipo)`
  Permite incorporar un nuevo equipo al inventario del gimnasio.
- `verDisponibilidad()`
  Devuelve información sobre si el equipo está disponible para uso o reservado para alguna actividad.



Usuario

La clase **Usuario** funciona como una superclase abstracta que modela los atributos y comportamientos comunes de todas las personas que interactúan con el sistema, ya sean clientes, entrenadores u otro personal. Su propósito es facilitar la reutilización de de la clase para simplificar y unir con la gestión de datos personales.

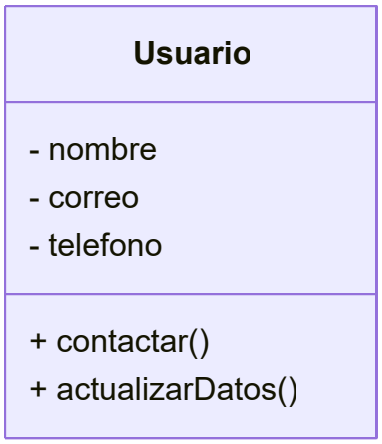
Atributos principales:

- `nombre`
  Nombre completo del usuario.
- `correo`
  Dirección de correo electrónico para contacto y notificaciones.

- `telefono`
  Número de teléfono asociado al usuario.

Métodos clave:

- `contactar()`
  Permite enviar un mensaje o notificación al usuario a través de los medios registrados (correo o teléfono).
- `actualizarDatos()`
  Permite modificar la información personal del usuario.



La creacion de esta class como super classe es nos permite crear classes como cliente, entrenador, personal, que hereden atributos y metodos comunes. Los atributos los mantemmos privados y usamos metodos para poder cambiar o actualizar los atribbutos asegudranod que solo la clase o subclases que puedamos crear puedan accedidar o modificar la information.

De esta forma tambien podemos al tener una super clase nos permite incporar nuevos metodos en el futuro y centraliza la gestion de information sensible.

Cliente

La clase **Cliente** representa a un cliente registrado en el gimnasio, generado a partir de la superclase abstracta **Usuario**. Modela las operaciones y atributos propios de un socio, permitiendo gestionar su relación con el gimnasio y su participación en actividades.

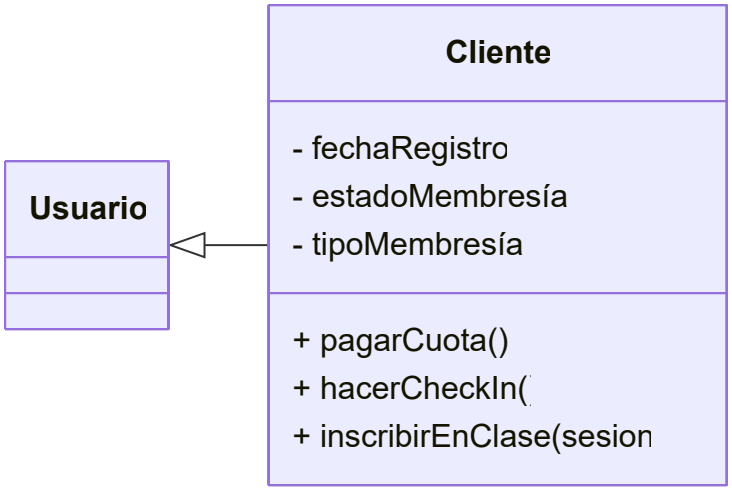
Atributos principales:

- `nombre`
  Nombre completo del miembro (heredado de Usuario).
- `correo`
  Correo electrónico para notificaciones (heredado de Usuario).
- `telefono`
  Número de contacto (heredado de Usuario).

- `fechaRegistro`
  Fecha en la que el miembro se inscribió al gimnasio.
- `estadoMembresía`
  Indica si la membresía del miembro está activa o no.
- `tipoMembresía`
  Plan de membresía asociado al miembro.

Métodos clave:

- `pagarCuota()`
  Permite al miembro realizar el pago de su cuota periódica.
- `hacerCheckIn()`
  Registra la entrada del miembro al gimnasio.
- `inscribirEnClase(sesion)`
  Permite al miembro inscribirse en una sesión grupal o clase.



Membresía

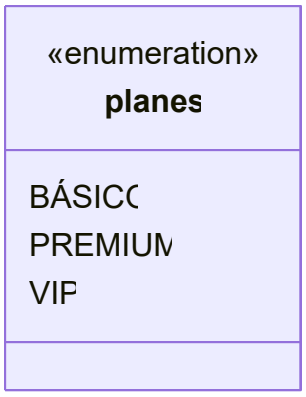
La clase **Membresía** modela el contrato de suscripción que permite a un cliente acceder a los servicios del gimnasio bajo ciertas condiciones y beneficios. Es fundamental para controlar el acceso, gestionar renovaciones y definir los privilegios de cada usuario

Atributos Principales

- `plan:planes
  Indica el tipo plan contradado, esto se hace atraves de una classe de tipo ennumeracion con la intencion de regstringir los valres posiblibles.
- `fecha_inicio`:
  Fehca en la que inicia la vigencia de la membresia
- `fecha date`:
  fecha en la que termina el plazo de la membresia

- `costo`:
  Monto associado con el plan
- `estado`:
  indicacion si la membresia esta activida o vencida

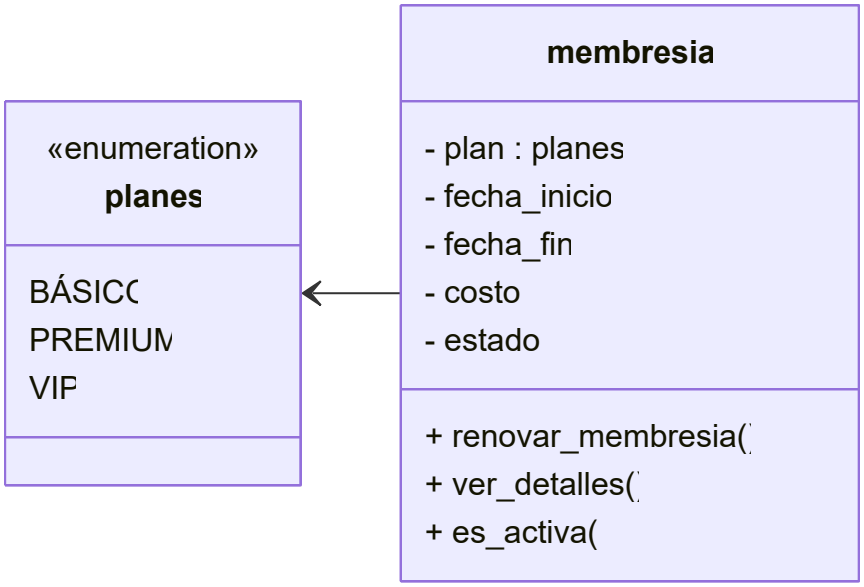
Enumeracion Asociada



- **Tipos de membresías**:
  - `BÁSICO`: Acceso a instalaciones básicas y horarios estándar
  - `PREMIUM`: Acceso completo a instalaciones y classes grupales
  - `VIP`: Total acceso, classes ilimitadas, y incluyendo entrenamiento personalizado

Metodos

- `renovar_membresia()` - Extiende la fecha de finalización de la membresía según la duración especificada que esto puede set mensual o annual.
- `ver_detalles()` - Devuelve una descripción detallada de la membresía, incluyendo beneficios, duración y costo.
- `es_activa()` - verifica si la membresia esta vigente con la fecha actual



Dejando los atributos privados podemos aprovechar de su encapsulamiento para que} solo los metodos de la clase puden modificar el estado de la membresia. Usando una clase de tipo enumeracion nos permite simplificar y ampliar los planes desde un punto centrar.

La classe se encarga exclusivamente de resguardar la suscripbcion, la renovacion y permite que consulta rs los detalle de la misma clase, siguento el principio de la responsibilidad unica.

Entrenador

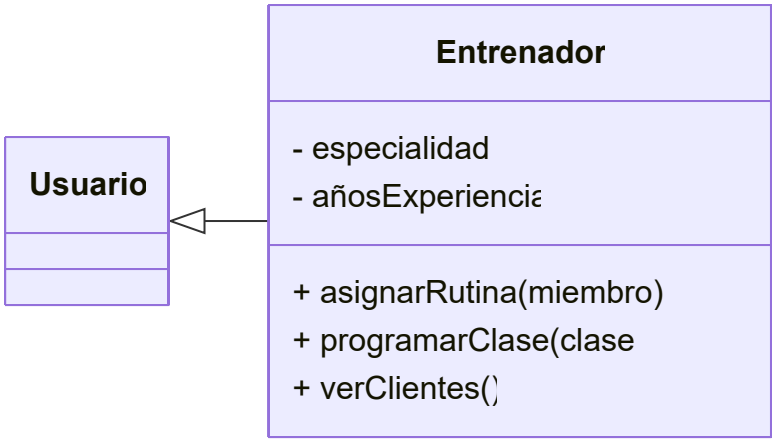
La clase **Entrenador** representa a un instructor o professional encargado de guiar y supervisar a los miembros del gimnasio. Es una especialización de la clase **Usuario**, lo que permite heredar atributos y métodos comunes, y añadir funcionalidades específicas relacionadas con la gestión de entrenamientos y clases.

Atributos principales:

- `nombre`
  Nombre completo del entrenador (heredado de Usuario).
- `correo`
  Correo electrónico para notificaciones (heredado).
- `telefono`
  Número de contacto (heredado).
- `fechaRegistro`
  Fecha de alta en el gimnasio (heredado).
- `especialidad`
  Área de conocimiento o disciplina principal del entrenador.
- `añosExperiencia`
  Tiempo de experiencia professional en el ámbito deportivo.

Métodos clave:

- `asignarRutina(miembro)`
  Permite diseñar y asignar un plan de entrenamiento personalizado a un miembro.
- `programarClase(clase)`
  Permite crear o programar una nueva sesión grupal o clase.
- `verClientes()`
  Devuelve la lista de miembros bajo su supervisión o entrenamiento.



SesiónEntrenamiento (clase)

La clase **SesiónEntrenamiento** representa una actividad grupal programada dentro del gimnasio, como yoga, spinning o cualquier otra clase colectiva. Esta entidad permite organizar, gestionar y controlar la participación de miembros y la asignación de entrenadores.

Atributos principales:

- `Nombre`
  Nombre de la clase para identificar de que va cubir.
- `fechaHora`
  Memento programado para el inicio de la sesión.
- `duracion`
  Tiempo estimado de la sesión.
- `estado`
  Indica si la sesión está programada, en curso, finalizada o cancelada.

Métodos clave:

- `iniciarSesion()`
  Cambia el estado de la sesión a "en curso" y permite el acceso de los participantes.
- `cancelarSesion()`
  Cambia el estado de la sesión a "cancelada" y notifica a los inscritos.
- `agregarParticipante(participante)`
  Permite registrar a un miembro como asistente a la sesión.
- `asignarEntrenador(entrenador)`
  Asocia un entrenador responsible de la sesión.

Asistencia

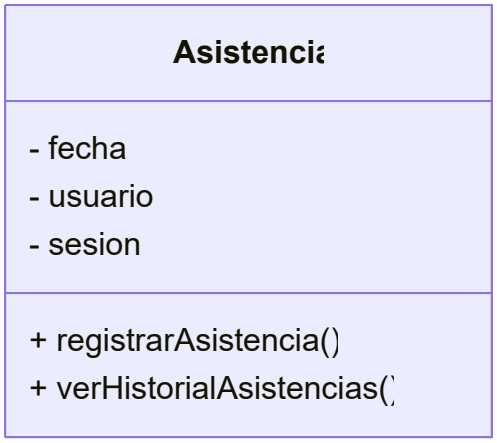
La clase **Asistencia** representa el registro de la participación de un usuario en una clase o sesión dentro del gimnasio. Esta entidad es fundamental para llevar un control histórico de la actividad de los clientes y para la gestión de reportes y análisis de uso.

Atributos principales:

- `fecha`
  Fecha y hora en la que se registra la asistencia.
- `usuario`
  Referencia al miembro que asiste (asociación).
- `sesion`
  Referencia a la sesión o clase a la que asiste el miembro.

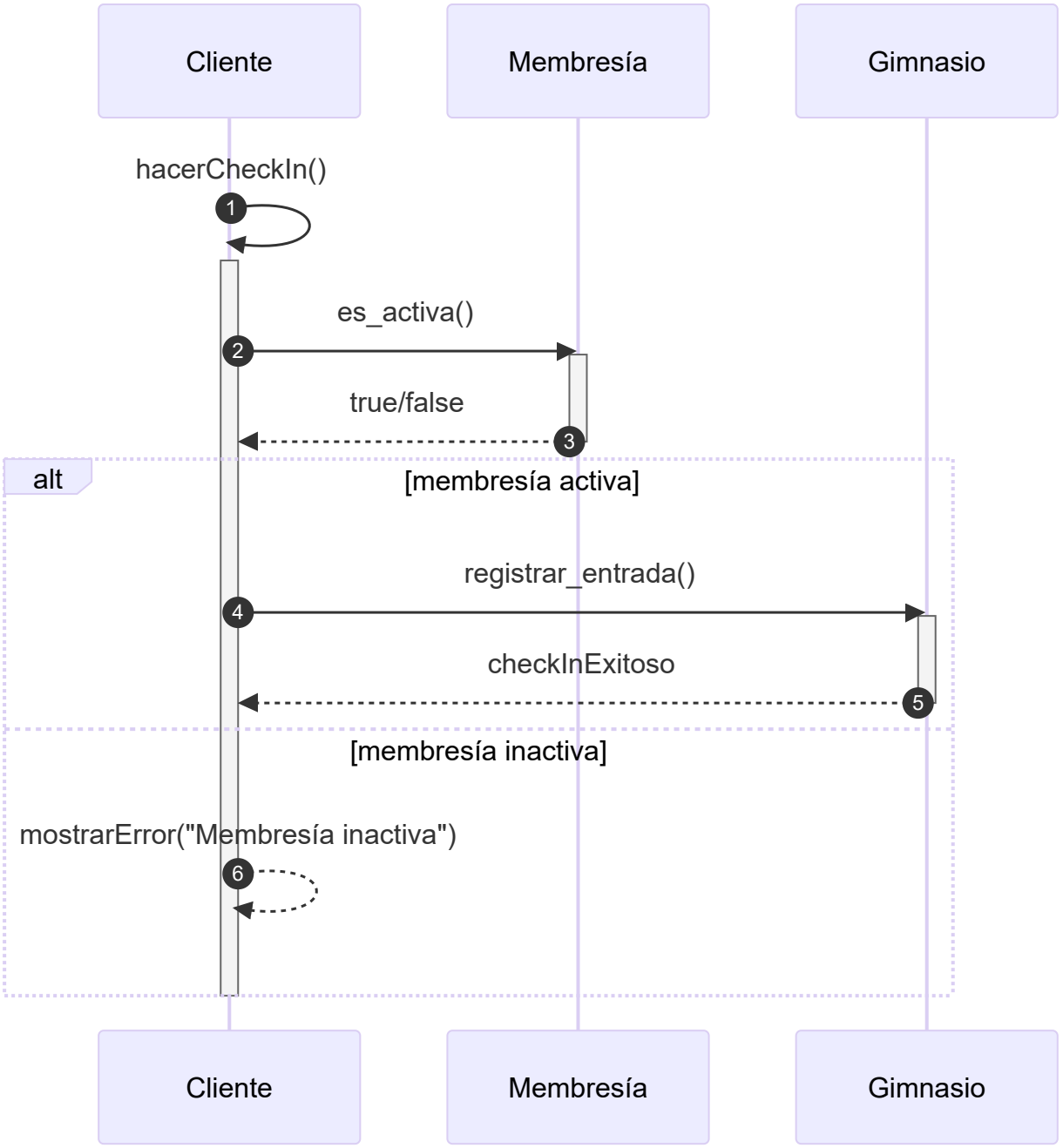
Métodos clave:

- `registrarAsistencia()`
  Registra la asistencia de un miembro a una clase, añadiendo una nueva entrada al historial.
- `verHistorialAsistencias()`
  Permite consultar el historial de asistencias de un miembro, facilitando el seguimiento de su participación en actividades.



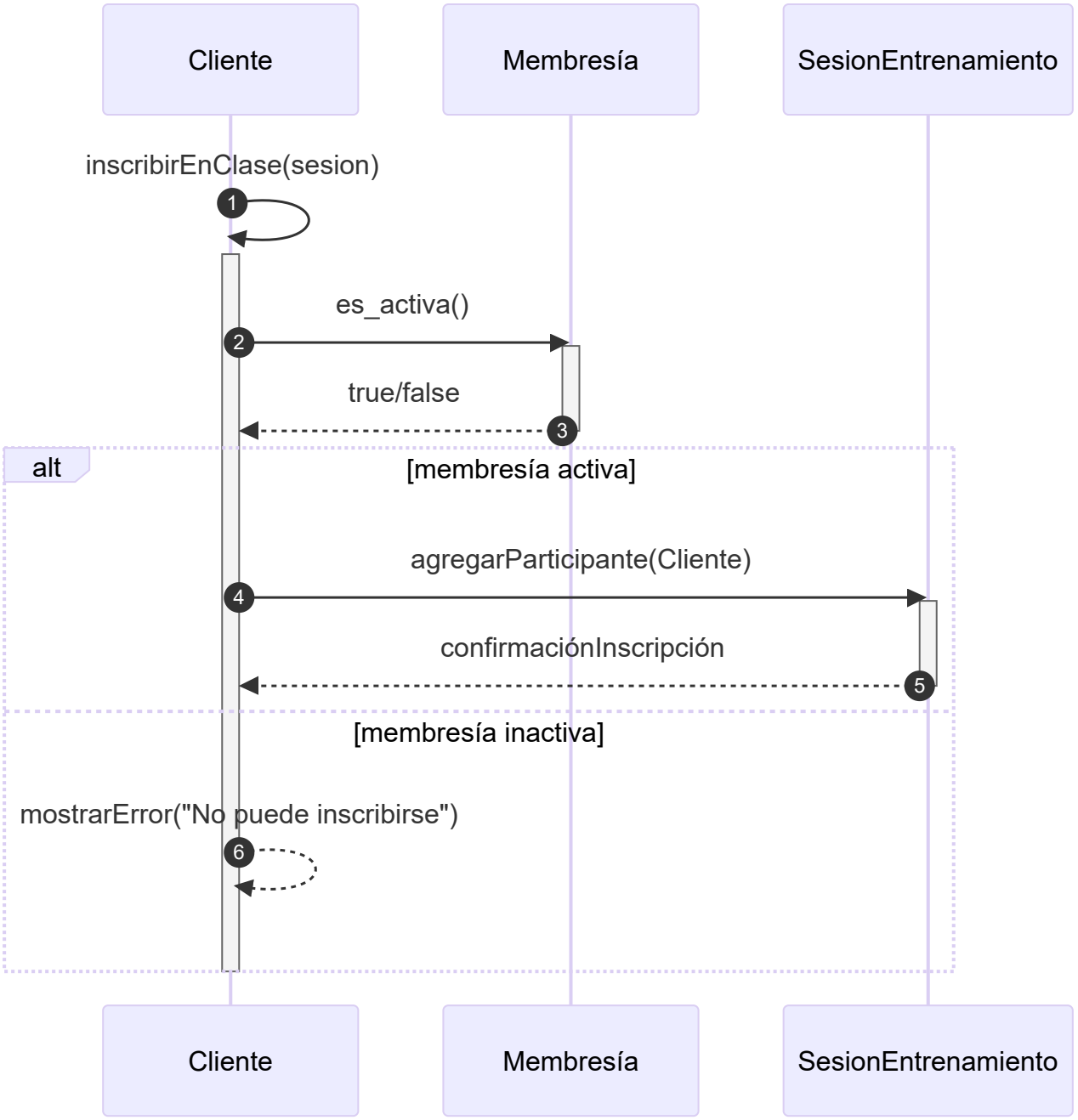
Relaciones Dinámicas Entre Clases (Diagrams De Secuencia)

1. Registro De Entrada De Un Cliente



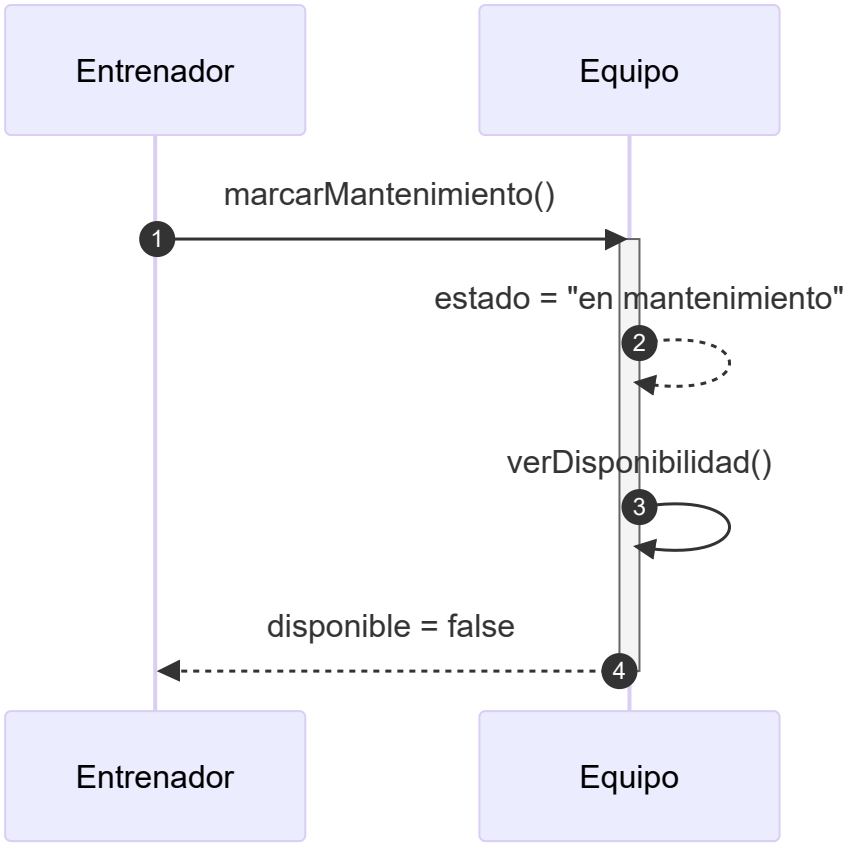
- **Clases implicadas**: `Cliente` (subclase de `Usuario`), `Gimnasio`, `Membresía`, `Asistencia`.
- **Tipo de relación**:
  - Asociación (mensaje entre instancias).
  - Composición implícita: `Gimnasio` crea/registra un objeto `Asistencia`.

2. Inscripción En Una Clase Grupales



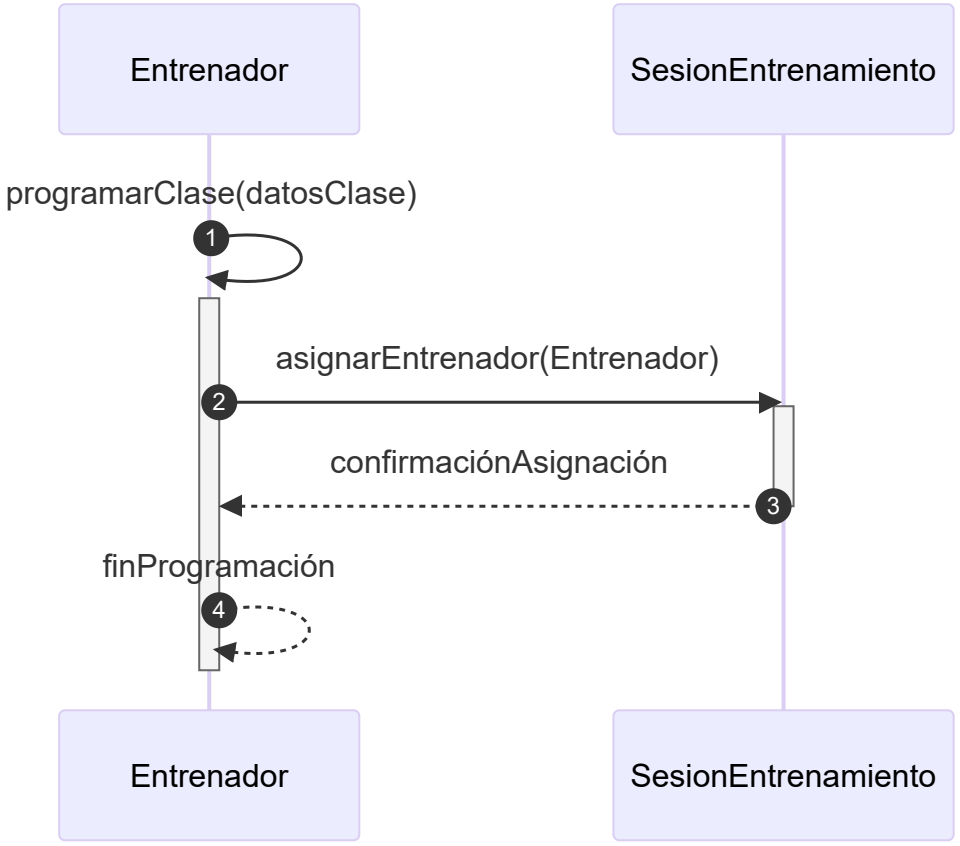
- **Clases implicadas**: `Cliente`, `SesiónEntrenamiento`.
- **Tipo de relación**:
  - Asociación bidireccional implícita (cliente envía solicitud y la sesión mantiene lista de participantes).

3. Programación De Una Clase Grupal Por Un Entrenador



- **Clases implicadas**: `Entrenador`, `Equipo`
- **Tipo de relación**:

4. Programar Mantenimiento De Un Equipo



- **Clases implicadas**: `Entrenador`, `SesiónEntrenamiento`, `Gimnasio`.
- **Tipo de relación**:
  - Asociación: el entrenador solicita al gimnasio verificar recursos.
  - Composición: `Gimnasio` mantiene agregados los espacios que usa la sesión.

Relaciones Entre Clases

> **Agregar explicacion de las relaciones entre las clases y como se enlazan**

Análisis De Los Flujos Principales E Interacciones

Basado en los casos de usos que se identifican podemos intuir y analizar los siguientes flujos principales sobre las interactions entre los actores y components del sistema:

1 . Gestion De Usuarios Y acceso

- **Durante el registro de un nuevo usuario**
  1. Un administrador inicia el proceso de registro en el Sistema
  2. El sistema solicita datos básicos (nombre, contacto, etc)
  3. Se selecciona el tipo de usuario (cliente o personal, entrenador)
  4. Si es cliente se vincula con una membership
  5. El sistema valida la information proporcionada
  6. Se generan credenciales para el ingreso
  7. El usuario recibe notificación de registro existo con credenciales
- **Interacciones clave:**
  - `Usuario` ↔ `Membresía`: Al registrar un cliente, debe asociarse con un tipo de membresía
  - `Usuario` ↔ `Gimnasio`: Los usuarios registrados pueden acceder a las instalaciones atreves de credenciales

2. Administration De instalaciones

- **Gestión de espacios:**
  1. El administrador accede al módulo de gestión de espacios
  2. Visualiza la distribución actual de las instalaciones
  3. Asigna equipos específicos a cada área
  4. Configura horarios de disponibilidad de instalaciones
  5. El sistema actualiza la información y la hace visible para los usuarios

Interacciones clave:

- `Gimnasio` ↔ `Equipo`: El gimnasio contiene diversos equipos distribuidos en sus espacios

- `Gimnasio` ↔ `Clase`: Las clases se asignan a espacios específicos dentro del gimnasio

3. Control De equipo

Programación de mantenimiento:

1. El administrador identifica equipos que requieren mantenimiento
2. Programa fechas para revisión o reparación
3. El sistema marca el equipo como "En mantenimiento"
4. Se notifica al personal relevante
5. Al completar el servicio, se actualiza el estado del equipo
6. El sistema registra el historial de mantenimiento

Interacciones clave:

- `Equipo` ↔ `Clase`: La disponibilidad de equipos afecta la programación de clases
- `Equipo` ↔ `Gimnasio`: El inventario de equipos es parte integral del gimnasio

4. Programación De actividades

Creación de clases grupales:

1. El entrenador accede al módulo de programación
2. Define el tipo de clase (yoga, spinning, etc.)
3. Establece fecha, hora y duración
4. Asigna un espacio específico
5. Define capacidad máxima
6. El sistema verifica disponibilidad y no conflictos
7. La clase se publica y queda disponible para inscripción

Interacciones clave:

- `Clase` ↔ `Entrenador`: Cada clase es impartida por un entrenador específico
- `Clase` ↔ `Cliente`: Los clientes se inscriben en las clases
- `Clase` ↔ `Asistencia`: El sistema registra la asistencia a cada clase

5. Gestión De membresías

Flujo principal - Renovación de planes:

1. El cliente o recepcionista inicia el proceso de renovación
2. El sistema muestra el estado actual y opciones disponibles
3. Se selecciona el nuevo período de vigencia
4. Se aplican promociones si corresponde
5. Se procesa el pago

6. El sistema actualiza la fecha de vencimiento
7. Se notifica la renovación exitosa

Interacciones clave:

- `Membresía` ↔ `Cliente`: Cada cliente tiene asociada una membresía con características específicas
- `Membresía` ↔ `Clase`: El tipo de membresía podría determina el acceso a ciertas clases

6. Control De asistencia

Registro de entrada/salida:

1. El cliente se identifica en el punto de acceso (tarjeta, biometría, etc.)
2. El sistema verifica la vigencia de la membresía
3. Se registra la fecha y hora de entrada
4. Al salir, el cliente se identifica nuevamente
5. El sistema registra la hora de salida
6. Se actualiza el historial de asistencia del cliente

Interacciones clave:

- `Asistencia` ↔ `Cliente`: Se registra cada visita del cliente
- `Asistencia` ↔ `Clase`: Se controla la participación en actividades específicas
- `Asistencia` ↔ `Gimnasio`: Permite monitorear el aforo en tiempo real

7. Entrenamiento personalizado

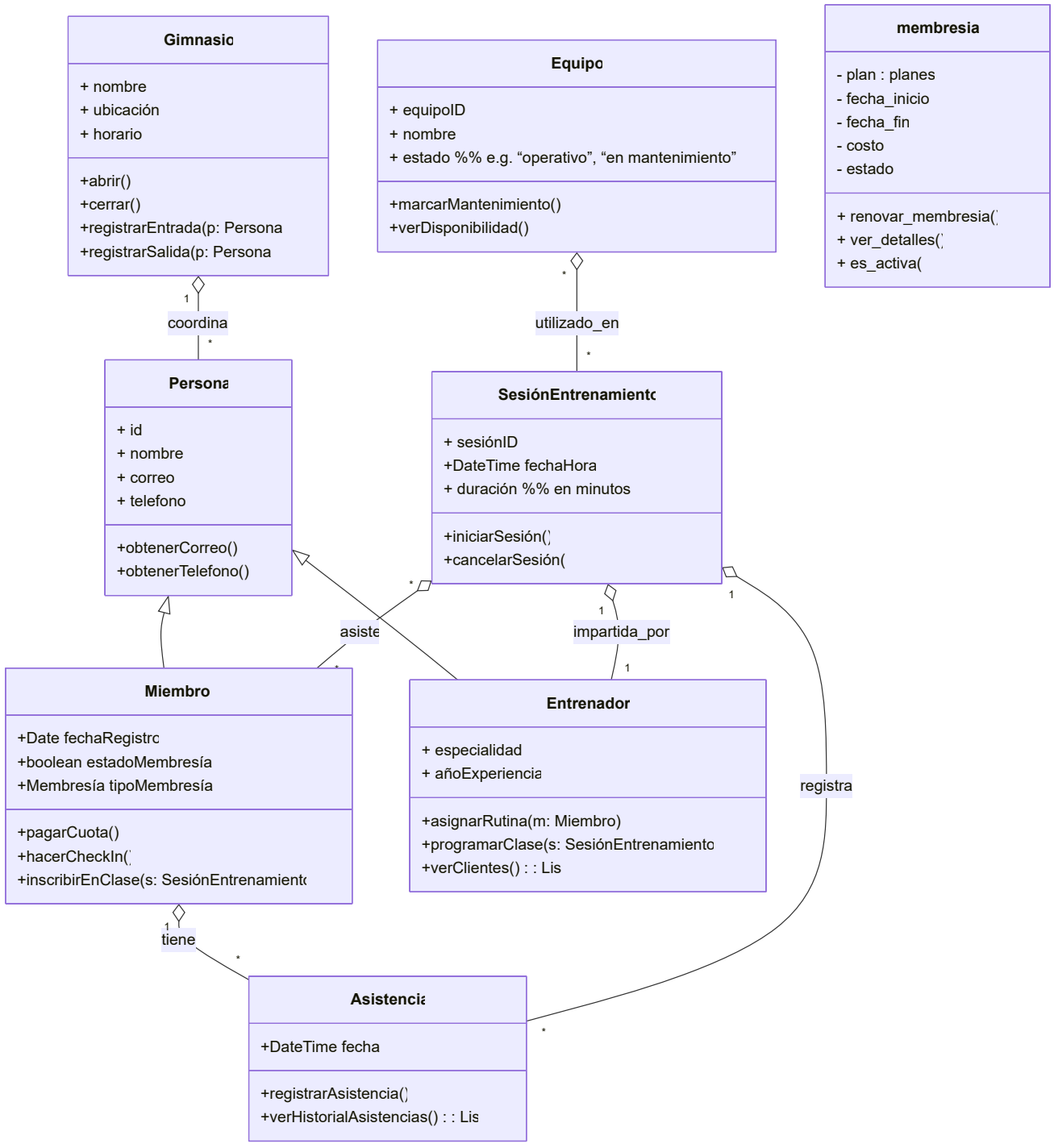
Flujo principal - Asignación de rutinas:

1. El entrenador accede al perfil del cliente
2. Evalúa su condición física y objetivos
3. Diseña una rutina personalizada
4. Asigna ejercicios específicos con series y repeticiones
5. Establece progresión esperada
6. El cliente recibe notificación de su nueva rutina
7. El sistema permite seguimiento del cumplimiento

Interacciones clave:

- `Entrenador` ↔ `Cliente`: Relación directa para la personalización del entrenamiento
- `Cliente` ↔ `Equipo`: Las rutinas incluyen el uso de equipos específicos

Diagrama De Clases UML


Gimnasio
+ nombre
+ ubicación
+ horario
+abrir()
+cerrar()
+registrarEntrada(p: Persona
+registrarSalida(p: Persona
Equipo
+ equipoID
+ nombre
+ estado %% e.g. “operativo”, “en mantenimiento”
+marcarMantenimiento()
+verDisponibilidad()
membresia
- plan : planes
- fecha_inicio
- fecha_fin
- costo
- estado
+ renovar_membresia(
+ ver_detalles(
+ es_activa(
1
coordina
*
*
utilizado_en
*
Persona
+ id
+ nombre
+ correo
+ telefono
+obtenerCorreo()
+obtenerTelefono()
SesiónEntrenamiento
+ sesiónID
+DateTime fechaHora
+ duración %% en minutos
+iniciarSesión()
+cancelarSesión(
*
asiste
*
1
impartida_por
1
1
registra
Miembro
+Date fechaRegistro
+boolean estadoMembresía
+Membresía tipoMembresía
+pagarCuota()
+hacerCheckIn()
+inscribirEnClase(s: SesiónEntrenamiento
Entrenador
+ especialidad
+ añoExperiencia
+asignarRutina(m: Miembro)
+programarClase(s: SesiónEntrenamiento
+verClientes() : : Lis
1
tiene
*
Asistencia
+DateTime fecha
+registrarAsistencia()
+verHistorialAsistencias() : : Lis
*

Conclusión