Actividad\_01\_Miguel\_DeJesus\_Chavez\_Barragan

# Análisis Del Sistema Para Una Aplicación De Gimnasio

## Resumen

Este trabajo presenta un análisis orientado a objetos de una aplicación de gestión de gimnasio, describiendo las clases principales del sistema (por ejemplo, *Gimnasio*, *Miembro*, *Entrenador*, *SesiónEntrenamiento*, *Equipo* y *Membresía*) 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 un diagrama de clases UML que sintetiza la estructura estática del sistema. Las referencias citadas (en formato APA) sustentan la definición de elementos UML y conceptos de análisis orientado a objetos.

## Introducción

El **análisis orientado a objetos (OOA)** modela el dominio del problema en términos de clases, objetos, atributos y métodos (Avila & Bailón, 2022)[portalacademico.cch.unam.mx](https://portalacademico.cch.unam.mx/cibernetica1/analisis-y-diseno-en-poo/clases-y-objetos#:~:text=En%20la%20programaci%C3%B3n%20orientada%20a,una%20interfaz%20gr%C3%A1fica%2C%20archivos%2C%20etc). El Lenguaje de Modelado Unificado (UML) provee una notación estandarizada para visualizar estos elementos (Lucidchart, s. f.)[lucidchart.com](https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml#:~:text=El%20UML%20se%20estableci%C3%B3%20como,interacci%C3%B3n%20entre%20clases%20y%20objetos). En particular, los *diagramas de clases UML* representan la estructura estática de un sistema mostrando clases, atributos y operaciones y sus relaciones[lucidchart.com](https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#:~:text=Los%20diagramas%20de%20clases%20representan,superposici%C3%B3n%20entre%20estos%20dos%20grupos)[gleek.io](https://gleek.io/templates/gym-management-class#:~:text=About%20class%20diagrams). Según Lucidchart (s. f.), cada clase se dibuja como un rectángulo de tres secciones (nombre, atributos, métodos)[lucidchart.com](https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml#:~:text=La%20figura%20de%20clase%20en,se%20agrupan%20para%20mostrar%20la). Los diagrams UML facilitan comprender sistemas complejos al convertir código en modelos visuals manejables (Miro, s. f.)[miro.com](https://miro.com/es/diagrama/que-es-diagrama-uml/#:~:text=Los%20diagramas%20UML%20facilitan%20la,personas%20no%20expertas%20en%20tecnolog%C3%ADa).

En este análisis se aborda una aplicación de gestión de gimnasio que integra funciones de membresías, programación de clases, control de acceso y gestión de pagos[creately.com](https://creately.com/diagram/example/YjWlCVzccFY/sistema-de-gestion-de-gimnasio#:~:text=Un%20Sistema%20de%20Gesti%C3%B3n%20de,mejora%20la%20experiencia%20del%20cliente). Se describirán las clases identificadas en el sistema, sus atributos y métodos, las relaciones entre ellas (asociaciones, herencias, composiciones) y se presentará un diagrama UML de clases. Cada decisión de modelado se basa en los principios de abstracción y modularidad de la POO (Avila & Bailón, 2022)[portalacademico.cch.unam.mx](https://portalacademico.cch.unam.mx/cibernetica1/analisis-y-diseno-en-poo/clases-y-objetos#:~:text=En%20la%20programaci%C3%B3n%20orientada%20a,una%20interfaz%20gr%C3%A1fica%2C%20archivos%2C%20etc), así como en convenciones UML (Lucidchart, s. f.)[lucidchart.com](https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#:~:text=Los%20diagramas%20de%20clases%20representan,superposici%C3%B3n%20entre%20estos%20dos%20grupos).

## 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:** Representa el centro deportivo en sí mismo. *Atributos:* nombre: String, ubicación: String, horario: String. *Métodos:* abrir(), cerrar(), registrarEntrada(Miembro), registrarSalida(Miembro). Esta clase puede considerarse un servicio o entidad global que coordina las operaciones del gimnasio.
* **Persona:** (Superclase abstracta) Modelo común para individuos en el sistema. *Atributos:* id: Int, nombre: String, correo: String, telefono: String. *Métodos:* contactar(). Se propone para agrupar atributos comunes de **Miembro** y **Entrenador** a través de herencia (generalización)[visual-paradigm.com](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/#:~:text=Generalization%20vs%20Specialization).
* **Miembro:** Hereda de *Persona*. Representa a un cliente registrado en el gimnasio. *Atributos:* fechaRegistro: Date, estadoMembresía: Boolean, tipoMembresía: Membresía (enumeración). *Métodos:* pagarCuota(), hacerCheckIn(), inscribirEnClase(SesiónEntrenamiento). Modela las operaciones típicas de un socio del gimnasio.
* **Entrenador:** Hereda de *Persona*. Representa a un instructor o entrenador del gimnasio. *Atributos:* especialidad: String, añoExperiencia: Int. *Métodos:* asignarRutina(Miembro), programarClase(SesiónEntrenamiento). Un *Entrenador* puede gestionar sesiones de entrenamiento y asignar rutinas personalizadas.
* **SesiónEntrenamiento:** Representa una clase o sesión grupal (p. ej. yoga, spinning). *Atributos:* sesiónID: Int, fechaHora: DateTime, duración: Int. *Métodos:* iniciarSesión(), cancelarSesión(). Esta clase se asocia con **Miembro** (asistencia de miembros) y **Entrenador** (conduce la sesión), sin que exista dependencia de vida fuerte entre ellas.
* **Equipo:** Representa un equipo o máquina del gimnasio (p. ej. cinta, pesa). *Atributos:* equipoID: Int, nombre: String, estado: String. *Métodos:* marcarMantenimiento(). Los equipos son utilizados en las sesiones, pero en el modelo propuesto existen como entidades propias.
* **Membresía:** (Enumeración) Define los tipos de planes disponibles: BÁSICO, PREMIUM, VIP. **No es una clase de entidad**, sino un tipo de dato especial. Se usa como atributo de *Miembro* para indicar su plan contratado. No posee métodos.

Cada clase se diseña conforme a las convenciones UML: nombre en mayúscula inicial, atributos con notación nombre:Tipo y métodos con paréntesis de parámetros[portalacademico.cch.unam.mx](https://portalacademico.cch.unam.mx/cibernetica1/analisis-y-diseno-en-poo/clases-y-objetos#:~:text=utilizaremos%20una%20lista%20de%20caracter%C3%ADsticas,donde%20se%20listan%20los%20m%C3%A9todos)[lucidchart.com](https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml#:~:text=La%20figura%20de%20clase%20en,se%20agrupan%20para%20mostrar%20la). Por ejemplo, la figura de la clase *Miembro* incluiría su identificador, nombre, correo y membresía como atributos, y las operaciones pagarCuota() o hacerCheckIn() en la sección de métodos. El uso de tres compartimentos para cada clase (nombre, atributos, métodos) sigue la convención descrita en Lucidchart[lucidchart.com](https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml#:~:text=La%20figura%20de%20clase%20en,se%20agrupan%20para%20mostrar%20la)[lucidchart.com](https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#:~:text=Los%20diagramas%20de%20clases%20representan,superposici%C3%B3n%20entre%20estos%20dos%20grupos).

## Relaciones Entre Clases

En el modelo se utilizan los siguientes tipos de relaciones UML, justificadas por la naturaleza de cada vínculo entre clases:

* **Herencia (Generalización):** Se propone la clase *Persona* como superclase de *Miembro* y *Entrenador*. Esto permite compartir atributos comunes (como nombre y contacto) en la superclase y especializarlos en cada subclase[visual-paradigm.com](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/#:~:text=Generalization%20vs%20Specialization). En UML, la herencia se representa con una flecha con cabeza de triángulo abierto dirigida de la subclase a la superclase. Visual Paradigm señala que la generalización agrupa atributos y operaciones comunes en una clase base, mientras que las subclases añaden características especializadas[visual-paradigm.com](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/#:~:text=Generalization%20vs%20Specialization). Este patrón de diseño evita duplicación y mejora la mantenibilidad del modelo.
* **Asociación:** Define una relación semántica entre objetos. Por ejemplo, *Miembro* asiste a *SesiónEntrenamiento* (muchos a muchos) y *Entrenador* conduce una *SesiónEntrenamiento* (uno a muchos). Estas son asociaciones simples: se representan con líneas sólidas. La asociación implica que ambas clases conocen la existencia de la otra pero no existe un ciclo de vida dependiente. Como indica Visual Paradigm, una asociación es un término genérico donde una clase puede utilizar la funcionalidad de otra[visual-paradigm.com](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/#:~:text=To%20sum%20it%20up%20association,If%20it%20can%20then). En este caso, la relación es bidireccional (un miembro puede tener múltiples sesiones y cada sesión puede tener varios miembros) sin que la eliminación de uno afecte al otro.
* **Agregación:** Se considera para relaciones “parte-todo” débiles. Por ejemplo, podríamos modelar que *Gimnasio* contiene varios *Equipo* mediante agregación (rombo blanco) si se interpreta que los equipos pueden pertenecer al gimnasio pero no dependen completamente de él. La agregación permite compartir los components; su eliminación no destruye necesariamente a las partes[seas.es](https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/#:~:text=La%20agregaci%C3%B3n%20es%20un%20tipo,mayor%20frecuencia%20que%20la%20composici%C3%B3n). En UML, se dibuja un diamante blanco en el extremo del “todo”[seas.es](https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/#:~:text=La%20agregaci%C3%B3n%20se%20representa%20en,clase%20que%20representa%20el%20%E2%80%9Ctodo%E2%80%9D). Sin embargo, dado que en nuestro modelo los equipos forman parte integral del gimnasio, optamos por modelar esta relación como composición (ver a continuación).
* **Composición:** Se usa cuando la relación “parte-todo” es fuerte: la parte no puede existir independientemente del todo. En el modelo, *Gimnasio* contiene a *Equipo* en una composición. Si el objeto *Gimnasio* es destruido, los *Equipo* asociados se eliminan con él. Según SEAS (2013), “la supresión del objeto compuesto conlleva la supresión de los components” en una composición[seas.es](https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/#:~:text=Composici%C3%B3n%20es%20una%20forma%20fuerte,Compuesto). En UML la composición se representa con un diamante negro en el lado del todo[seas.es](https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/#:~:text=Composici%C3%B3n%20es%20una%20forma%20fuerte,Compuesto). Esta decisión refleja que el equipo gestionado está ligado al gimnasio concreto.
* **Asociación/Atributo (Membresía):** El tipo de membresía se modela como un atributo de *Miembro* de tipo *Membresía* (enumeración). Esta no es una clase compleja, por lo que su relación es simplemente de dependencia (atributiva). Técnicamente, es una asociación débil: el cambio en la membresía de un miembro no destruye la instancia de *Membresía*.

En resumen, las relaciones gráficas se dibujan con la notación UML estándar: flechas abiertas para herencia, líneas simples para asociaciones, rombos blancos para agregaciones y rombos negros para composiciones[lucidchart.com](https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml#:~:text=La%20figura%20de%20clase%20en,se%20agrupan%20para%20mostrar%20la)[seas.es](https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/#:~:text=El%20s%C3%ADmbolo%20de%20composici%C3%B3n%20es,Compuesto). Las decisiones de usar herencia o composición se sustentan en cómo Visual Paradigm y fuentes académicas describen estas relaciones: se prefiere herencia para compartir propiedades comunes (generalización)[visual-paradigm.com](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/#:~:text=Generalization%20vs%20Specialization), composición para dependencias de ciclo de vida fuerte[seas.es](https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/#:~:text=Composici%C3%B3n%20es%20una%20forma%20fuerte,Compuesto), y agregación para dependencias más débiles[seas.es](https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/#:~:text=La%20agregaci%C3%B3n%20es%20un%20tipo,mayor%20frecuencia%20que%20la%20composici%C3%B3n).

## Diagrama De Clases UML

Error parsing Mermaid diagram!  
  
No diagram type detected matching given configuration for text:

Pasted image 20250520214141.png  
*Figura 1. Diagrama de clases UML del sistema de gestión de gimnasio.*

En la figura anterior se muestra un possible diagrama de clases UML que sintetiza las clases y relaciones descritas. Cada clase se dibuja como un rectángulo dividido en tres secciones (nombre, atributos y métodos)[lucidchart.com](https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml#:~:text=La%20figura%20de%20clase%20en,se%20agrupan%20para%20mostrar%20la)[lucidchart.com](https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#:~:text=Los%20diagramas%20de%20clases%20representan,superposici%C3%B3n%20entre%20estos%20dos%20grupos). Por ejemplo, la sección superior de *Member* (Miembro) contiene el nombre de la clase, la sección media sus atributos (MemberID: int, Name: string, Email: string, etc.) y la inferior los métodos (CheckIn(), etc.)[lucidchart.com](https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml#:~:text=La%20figura%20de%20clase%20en,se%20agrupan%20para%20mostrar%20la). Para mayor claridad, en la figura se han omitido los modificadores de visibilidad, asumiendo todos los atributos y operaciones como públicos.

Las relaciones se indican con diferentes símbolos UML: un triángulo abierto en *Trainer* y *Member* apuntando a *Persona* indica herencia (generalización)[visual-paradigm.com](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/#:~:text=Generalization%20vs%20Specialization). La flecha con rombo negro entre *Gym* y *Equipment* indica composición (Gym “contiene” Equipment)[seas.es](https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/#:~:text=Composici%C3%B3n%20es%20una%20forma%20fuerte,Compuesto), mientras que el rombo blanco sugeriría una agregación (en este caso conceptualizada para enfatizar la pertenencia débil, si se considerara). Las etiquetas en las líneas (por ejemplo, “attends” o “conducts”) describen el rol de la relación. En UML también se pueden indicar multiplicidades (e.g. 0..\* para muchos, 1 para uno) junto a los extremos de las asociaciones.

Este diagrama apoya el modelado orientado a objetos: como señala Gleek (2025), los diagrams de clase describen la estructura de un sistema mostrando clases, atributos, métodos y sus relaciones en forma gráfica[gleek.io](https://gleek.io/templates/gym-management-class#:~:text=About%20class%20diagrams). Aquí se observa claramente qué clases participan y cómo se conectan (asociaciones para asistencia de miembros a sesiones, herencia para compartir atributos de persona, composición para equipos, etc.). Las convenciones de notación (rombos, flechas, compartimentos) cumplen con la especificación UML, facilitando la comprensión del sistema a desarrolladores y demás interesados (Miro, s. f.)[miro.com](https://miro.com/es/diagrama/que-es-diagrama-uml/#:~:text=Los%20diagramas%20UML%20facilitan%20la,personas%20no%20expertas%20en%20tecnolog%C3%ADa).

## Conclusión

El análisis orientado a objetos de la aplicación de gimnasio ha identificado las entidades clave (gimnasio, personas, clases, equipamiento) y sus atributos y métodos pertinentes, apoyado en la notación UML como herramienta de modelado. Se definieron relaciones adecuadas: herencia para generalizar *Persona*, asociaciones para la interacción entre miembros, entrenadores y sesiones, y composición para vincular equipos al gimnasio. Este esquema semidesarrollado constituye la base de un ensayo completo, detallando la justificación técnica de cada clase y relación. Como evidencian las fuentes, los diagrams de clase son instrumentos valiosos en ingeniería de software para visualizar el diseño (Gleek, 2025)[gleek.io](https://gleek.io/templates/gym-management-class#:~:text=About%20class%20diagrams) y sirven de guía para la implementación posterior. En conclusión, el enfoque orientado a objetos y la notación UML facilitan la estructuración y documentación del sistema de gimnasio de manera clara y formal (Lucidchart, s. f.)[lucidchart.com](https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#:~:text=Los%20diagramas%20de%20clases%20representan,superposici%C3%B3n%20entre%20estos%20dos%20grupos).

## Referencias

* Avila, J. y Bailón, J. (2022). *Clases y objetos*. En *Análisis y diseño en POO*. Portal Académico del CCH, UNAM. Recuperado de <https://portalacademico.cch.unam.mx/cibernetica1/analisis-y-diseno-en-poo/clases-y-objetos>
* Gleek. (2025). *Gym management system – UML Class diagram example*. Recuperado de <https://gleek.io/templates/gym-management-class>
* Lucidchart. (s. f.-a). *Tutorial de diagrama de clases UML*. Recuperado de <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>
* Lucidchart. (s. f.-b). *¿Qué es el lenguaje unificado de modelado (UML)?*. Recuperado de <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>
* Miro. (s. f.). *Diagrama UML: qué es, cómo hacerlo y ejemplos*. Recuperado de <https://miro.com/es/diagrama/que-es-diagrama-uml/>
* SEAS Estudios Superiores Abiertos. (2013, 25 de enero). *Agregación vs Composición en diagrams de clases UML*. Recuperado de <https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/>
* Visual Paradigm. (2024). *UML Association vs Aggregation vs Composition*. Recuperado de <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/>