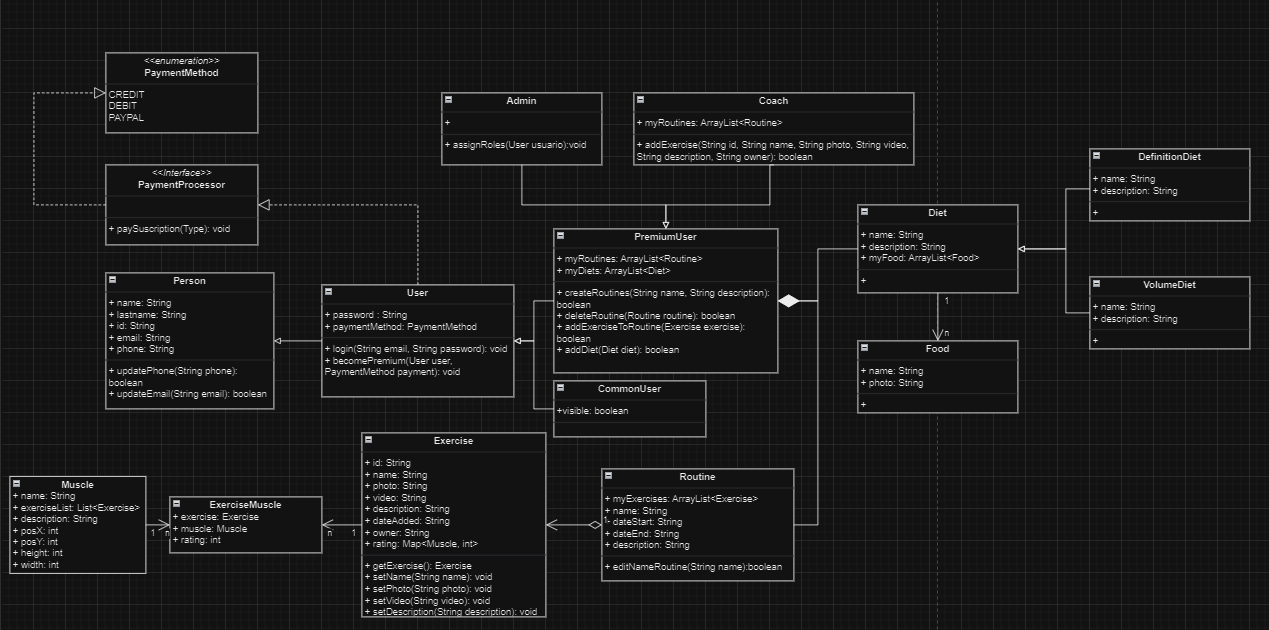
**Diseño del sistema: descomposición del sistema**

[Miguel Angel Perez Clavijo](mailto:miguela.perezc@autonoma.edu.co)

[Juan Esteban Briceño Marin](mailto:juane.bricenom@autonoma.edu.co)

[Camilo Andres Molano Aristizabal](mailto:camiloa.molanoa@autonoma.edu.co)

**Objetivos de diseño del sistema**Antes de continuar con los objetivos de diseño del sistema, es crucial tener en consideración el diagrama de clases de "GymHouse". Durante el desarrollo y análisis del proyecto, se han incorporado varias clases adicionales, tal como se ilustra en la siguiente imagen:



Se realizaron cambios significativos para mejorar la estructura de la relación entre músculos, ejercicios y su clasificación intermedia musculoEjercicio. En el diagrama de clases de la entrega anterior, la clase de músculo no estaba incluida, lo que generaba una relación confusa de muchos a muchos. La modificación busca optimizar la experiencia del usuario Premium. Por ejemplo al seleccionar un músculo específico, como el "pecho", al elegir dicho músculo, el ejercicio prioritario será el "press de banca", que obtiene una calificación de 10. Sin embargo, si se selecciona el músculo "tríceps", el "press de banca" no será la primera opción, ya que su calificación para ese músculo es de 4. Estos ajustes garantizan la coherencia y eficacia del sistema, como se ilustra en el ejemplo anteriormente mencionado.

Además, se implementó un cambio sustancial al incorporar dietas exclusivas para usuarios Premium, quienes pueden recibir múltiples planes alimenticios diseñados por sus entrenadores, ya sea para aumentar volumen muscular o definición. Cada dieta está compuesta por una lista detallada de comidas, que incluyen nombre, descripción y foto correspondiente.

Para diseñar un sistema como el proyecto "Gym House", es crucial identificar y priorizar los objetivos de diseño, teniendo en cuenta los factores de calidad del software según ISO/IEC 25000. Estos factores de calidad ayudan a garantizar que el sistema cumpla con las expectativas de los usuarios y las necesidades del negocio. A continuación, se describen los objetivos de diseño priorizados para este proyecto:

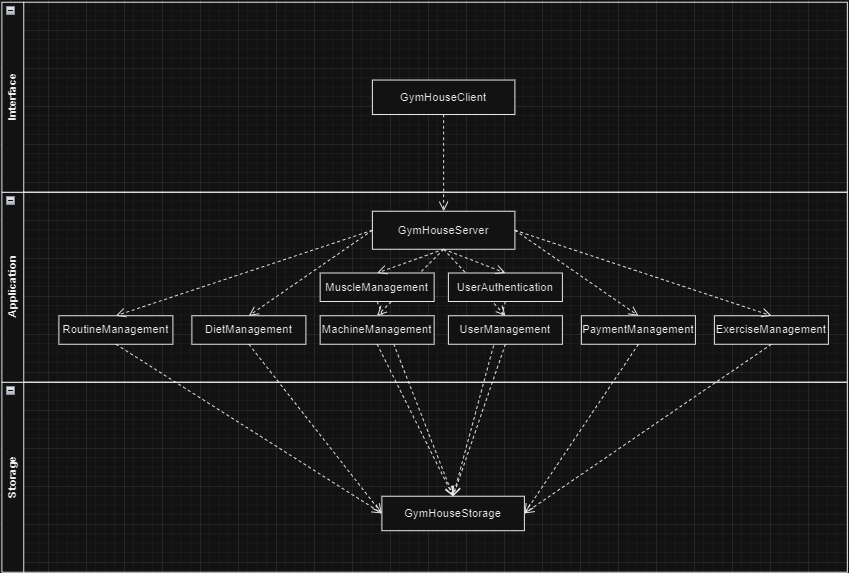
* **Funcionalidad**:
  + Este factor se refiere a la capacidad del sistema para cumplir con los requisitos funcionales especificados. Es fundamental garantizar que los usuarios puedan realizar todas las acciones necesarias, como agregar ejercicios, crear rutinas, asignar roles y visualizar detalles de los ejercicios.
  + **Prioridad:** Alta, debido a que es importante que el sistema cumpla con todos los requerimientos especificados para así brindar un servicio óptimo y garantizar que el usuario pueda aprovechar la totalidad de las funcionalidades planteadas.
* **Fiabilidad:**
  + La fiabilidad se refiere a la capacidad del sistema para realizar sus funciones de manera confiable y consistente. En el contexto de Gym House, la fiabilidad es crucial para garantizar que los usuarios puedan acceder al sistema en todo momento y que las funcionalidades como agregar ejercicios y crear rutinas funcionen correctamente sin errores o fallos inesperados, además de que no ocurra el caso en que el usuario llegue a perder las rutinas y dietas creadas.
  + **Prioridad:** Alta, puesto que es crucial que el sistema garantice que la información de los usuarios siga reflejando correctamente las tareas que se han completado con anterioridad para garantizar un buen servicio al usuario y al mismo tiempo, evitar problemas a largo plazo de pérdida de información.
* **Usabilidad:** 
  + La usabilidad se refiere a la facilidad de uso y la experiencia del usuario al interactuar con el sistema. En Gym House, es importante que la interfaz de usuario sea intuitiva y fácil de navegar, permitiendo a los usuarios realizar sus tareas de manera eficiente y sin confusiones. Además, los usuarios deben poder acceder fácilmente a las funcionalidades relevantes según su rol en el sistema.
  + **Prioridad:** Media, si bien es relevante garantizar una correcta interacción con el sistema, se pueden hacer mejoras a la misma con el pasar del tiempo, tratando de mantener una interfaz actualizada y acorde a las necesidades del mercado.
* **Eficiencia del rendimiento:**
  + Este factor se refiere al rendimiento del sistema en términos de velocidad, tiempo de respuesta y uso de recursos. En Gym House, es importante que el sistema responda rápidamente a las solicitudes de los usuarios, especialmente durante momentos de alta demanda, como cuando muchos usuarios intentan acceder al sistema simultáneamente o cuando se realizan operaciones intensivas, como la creación de rutinas con muchos ejercicios.
  + **Prioridad:** Alta, los tiempos de respuesta del usuario son una prioridad en el proyecto, puesto que es importante garantizar que los usuarios no van a sufrir de “cuelgues” en sus dispositivos y que de igual manera no se sufra de sobrecarga en el servidor, obteniendo así un rendimiento eficiente constante.
* **Mantenibilidad:**
  + La mantenibilidad se refiere a la facilidad con la que el sistema puede ser modificado, actualizado o reparado. En Gym House, es importante que el código sea modular y bien estructurado, lo que facilita la incorporación de nuevas funcionalidades o la corrección de errores sin afectar otras partes del sistema. Además, es crucial contar con una documentación clara y completa que facilite la comprensión y el mantenimiento del sistema por parte de los desarrolladores.
  + **Prioridad:** Alta, tener la aplicación correctamente diseñada implica que se pueda actualizar con facilidad, implementar nuevas funcionalidades y desarrollar mejoras en lo planteado con anterioridad. Esto garantiza que con el tiempo, los desarrolladores puedan acceder a módulos creados con mucha anterioridad y manipularlos fácilmente.
* **Portabilidad:**
  + La portabilidad se refiere a la capacidad del sistema para funcionar de manera efectiva en diferentes entornos de hardware o software. En Gym House, es importante que el sistema sea compatible con una variedad de dispositivos y navegadores web, lo que garantiza que los usuarios puedan acceder al sistema desde diferentes plataformas, como computadoras de escritorio, tabletas y dispositivos móviles, y utilizando diferentes navegadores web populares.
  + **Prioridad:** Media, esta capacidad del sistema se puede desarrollar con facilidad teniendo en cuenta las múltiples tecnologías presentes en el desarrollo web para este tipo de necesidades; por lo tanto se puede implementar en poco tiempo y no requiere de gran dedicación para lograr un correcto resultado.
* **Seguridad**:
  + La seguridad se refiere a la protección del sistema contra amenazas internas y externas, garantizando la confidencialidad, integridad y disponibilidad de los datos. En Gym House, es importante implementar medidas de seguridad robustas para proteger la información confidencial de los usuarios, como datos de inicio de sesión y detalles de pago, así como garantizar que el sistema esté protegido contra ataques maliciosos, como el acceso no autorizado y la manipulación de datos.
  + **Prioridad:** Alta, garantizar a los usuarios que su información estará protegida es parte crucial del sistema, puesto que se busca que cada persona tenga aislada su información y que únicamente él sea quien puede acceder a ella. Para ello se implementan diferentes herramientas que permiten la manipulación segura y encriptada de la información que se envía de los usuarios al servidor.

Priorización de objetivos de diseño:

* Funcionalidad
* Fiabilidad
* Usabilidad
* Eficiencia del rendimiento
* Mantenibilidad
* Portabilidad
* Seguridad

Al priorizar estos objetivos de diseño, se puede garantizar que el sistema Gym House cumpla con las expectativas de los usuarios y las necesidades del negocio, proporcionando una experiencia de usuario satisfactoria y segura.

Para refinar la descomposición inicial del sistema "Gym House" con base en los objetivos de diseño previamente establecidos, es importante revisar cada componente y asegurarse de que esté alineado con estos objetivos. A continuación, se describe cómo cada componente contribuye a los objetivos de diseño prioritarios:

* ****Interfaz de Usuario (UI):
  + **Usabilidad:** La interfaz de usuario debe ser intuitiva y fácil de usar para garantizar una experiencia positiva para los usuarios. Se deben diseñar páginas web y aplicaciones móviles que sean accesibles y estén optimizadas para diferentes dispositivos y tamaños de pantalla.
  + **Seguridad:** La UI debe implementar medidas de seguridad, como la autenticación de usuarios, para proteger la privacidad de la información y prevenir el acceso no autorizado a cuentas de usuario.
* Capa de Presentación:
  + **Usabilidad:** Los elementos de la capa de presentación deben ser diseñados de manera que faciliten la navegación y la interacción del usuario con el sistema. Se deben utilizar técnicas de diseño centradas en el usuario para garantizar que la interfaz sea intuitiva y fácil de entender.
  + **Eficiencia del rendimiento:** Los componentes de la capa de presentación deben optimizarse para garantizar tiempos de carga rápidos y una respuesta ágil del sistema, lo que contribuirá a una experiencia de usuario fluida y satisfactoria.
* Capa de Lógica de Negocio:
  + **Funcionalidad:** La capa de lógica de negocio debe implementar todas las funcionalidades requeridas por el sistema, asegurando que los usuarios puedan realizar todas las acciones necesarias, como agregar ejercicios, crear rutinas y asignar roles.
  + **Mantenibilidad:** El código en esta capa debe estar bien estructurado y modularizado para facilitar su mantenimiento y futuras expansiones del sistema.
* Capa de Acceso a Datos:
  + **Fiabilidad:** La capa de acceso a datos debe garantizar la integridad y la consistencia de los datos almacenados en la base de datos, evitando la pérdida o corrupción de información crítica.
  + **Eficiencia del rendimiento:** Se deben implementar técnicas de optimización de consultas y cacheo de datos para mejorar el rendimiento de las operaciones de lectura y escritura en la base de datos.

**MuscleManagement:** maneja la interfaz de usuario relacionada con la gestión de rutinas de entrenamiento, seguimiento de progreso muscular y acceso a recursos relacionados con el desarrollo muscular.

**UserAuthentication:** maneja la interfaz de usuario relacionada con la autenticación de usuarios, como el inicio de sesión y el registro de cuentas nuevas.

**RoutineManagement:** se encarga de manejar la interfaz de usuario relacionada con la gestión de rutinas de entrenamiento, permitiendo a los usuarios interactuar con ellas. En esta capa, se presentan las opciones para ver, crear, modificar y eliminar rutinas, así como también para asignarlas a usuarios específicos.

**DietManagement:** se encarga de gestionar la interfaz de usuario relacionada con la gestión de dietas y nutrición, permitiendo a los usuarios interactuar con sus planes de alimentación. En esta capa, se presentan opciones para visualizar, crear, modificar y eliminar planes de dieta, así como también para asignarlos a usuarios específicos.

**MachineManagement:** se encarga de gestionar la interfaz de usuario relacionada con la administración de máquinas y equipos en el gimnasio, permitiendo a los usuarios interactuar con las instalaciones disponibles.

**UserManagment:** maneja la interfaz de usuario relacionada con la gestión de usuarios en el sistema del gimnasio, permitiendo a los administradores y empleados interactuar con la información de los usuarios. En esta capa, se presentan opciones para crear, modificar y eliminar cuentas de usuario, así como para gestionar los roles y permisos.

**PaymentManagement:** gestiona la interfaz de usuario relacionada con los pagos en el sistema del gimnasio, permitiendo a los usuarios realizar transacciones financieras y a los administradores llevar un registro de los pagos.

**ExerciseManagement:** gestiona la interfaz de usuario relacionada con la administración de ejercicios en el sistema del gimnasio, permitiendo a los usuarios acceder a información sobre los ejercicios disponibles y a los entrenadores crear y modificar rutinas de ejercicios. En esta capa, se presentan opciones para buscar ejercicios, ver detalles de cada uno y seleccionarlos para su inclusión en rutinas de entrenamiento.

* Base de Datos:
  + **Fiabilidad:** La base de datos debe estar diseñada para garantizar la disponibilidad y la durabilidad de los datos, minimizando el riesgo de pérdida de información debido a fallos del sistema o errores humanos.
  + **Seguridad:** Se deben implementar medidas de seguridad robustas, como la encriptación de datos y el control de acceso basado en roles, para proteger la confidencialidad y la integridad de la información almacenada en la base de datos.
* Servicios Externos:
  + **Fiabilidad:** Los servicios externos utilizados por el sistema deben ser confiables y estar disponibles en todo momento para garantizar el funcionamiento adecuado de las funcionalidades que dependen de ellos.
  + **Seguridad:** Se deben establecer conexiones seguras con los servicios externos y autenticar correctamente las solicitudes para evitar el acceso no autorizado a datos sensibles.
* Seguridad:
  + **Seguridad:** Este componente debe implementar todas las medidas necesarias para proteger el sistema contra amenazas de seguridad, como ataques de inyección SQL, ataques de denegación de servicio (DDoS) y vulnerabilidades de seguridad en la capa de presentación y la capa de lógica de negocio.
* Gestión de Sesiones:
  + **Seguridad:** La gestión de sesiones debe implementar mecanismos de autenticación y autorización sólidos para garantizar que solo los usuarios autorizados puedan acceder a las funcionalidades del sistema y que sus sesiones se mantengan seguras y protegidas contra ataques de sesión.

Al refinar la descomposición del sistema con base en los objetivos de diseño, podemos garantizar que cada componente contribuya de manera efectiva a cumplir con estos objetivos, lo que resultará en un sistema más robusto, seguro y fácil de usar para los usuarios finales.