**Jhojan Stiven Moreno Bejarano**

**Informe sobre el Diagrama de Clases de FitLife**

En este diagrama de clases, diseñé las partes principales del sistema para que se entienda cómo funciona la aplicación FitLife y cómo se relacionan sus componentes.

**Clase Usuario**

La clase **Usuario** es el centro de todo porque representa a la persona que usa la app. Aquí puse atributos como nombre, edad, foto, objetivos, email y contraseña, que son datos básicos para que el usuario pueda crear su perfil y acceder a la aplicación. También tiene una lista de hábitos, porque cada usuario puede tener varios hábitos personalizados.

Los métodos de esta clase están relacionados con lo que un usuario puede hacer, como iniciar sesión, registrarse, cambiar su perfil, administrar sus hábitos, logros y estadísticas. Por eso están todos aquí, porque son acciones que hacen directamente los usuarios.

**Clase Hábito**

Esta clase guarda la información de cada hábito que un usuario crea o selecciona, como el nombre del hábito, categoría, fecha de creación, descripción y recordatorios. Está conectada a Usuario porque cada usuario tiene muchos hábitos, y se hace una relación de composición porque si el usuario desaparece, sus hábitos también.

Los métodos aquí permiten marcar un hábito como cumplido, dar recordatorios y acumular puntos, porque la app debe registrar cuándo el usuario cumple sus hábitos y también enviar notificaciones para motivarlos.

**Clase Recordatorio**

Está conectada directamente con Hábito porque cada hábito puede tener varios recordatorios para avisar al usuario cuándo debe realizar esa actividad. Tiene atributos básicos como título, mensaje y hora, y métodos para enviar la notificación, abrir la app y marcar si se cumplió o no.

**Clase Gamificación**

La idea de esta clase es manejar todo lo relacionado con el sistema de puntos, logros e incentivos que la app usa para motivar a los usuarios. Por eso tiene métodos para acumular puntos y brindar logros. Está conectada con Logros porque es la que administra esos premios, y también tiene una relación de dependencia con Usuario, porque depende de que un usuario use la app para funcionar.

**Clase Logros**

Esta clase representa los premios o insignias que un usuario puede ganar. Tiene atributos como nombre, descripción, cantidad y un mensaje de motivación. Se conecta con Gamificación porque esta es la que se encarga de otorgar los logros. Además, también está asociada a Usuario, porque cada usuario puede tener varios logros.

Los métodos aquí son para marcar automáticamente logros, dar medallas y mostrar mensajes para motivar al usuario, porque eso es lo que mantiene al usuario enganchado y con ganas de seguir usando la app.

**Clase Estadísticas**

La clase Estadísticas está para mostrarle al usuario un resumen de su progreso con gráficos, informes y porcentajes. Está conectada con Usuario porque cada usuario tiene sus propias estadísticas y datos históricos.

Los métodos se encargan de crear esos gráficos y resúmenes, para que el usuario pueda ver cómo va avanzando y se motive más.

**Relaciones y decisiones generales**

* Usé composiciones entre Usuario y Hábito, y entre Hábito y Recordatorio, porque estas partes no tienen sentido sin su "dueño". Si un usuario desaparece, sus hábitos y recordatorios también.
* Entre Usuario y Estadísticas usé composición también, porque las estadísticas son datos propios del usuario.
* Las relaciones entre Gamificación y Logros, y entre Usuario y Logros, son agregaciones porque los logros existen como objetos separados pero relacionados a los usuarios y a la gamificación.
* La dependencia de Gamificación hacia Usuario es para indicar que Gamificación utiliza al usuario para sus operaciones, pero no guarda su propia copia.

En resumen, el diagrama busca mostrar claramente cómo los datos y funciones principales de la app se relacionan y dependen unos de otros, facilitando la gestión de hábitos, el perfil del usuario, los recordatorios y la parte motivacional a través de gamificación y logros.

Quise que fuera sencillo de entender para cualquiera que vea el diagrama y, a la vez, completo para poder desarrollar la aplicación sin perder detalles importantes.