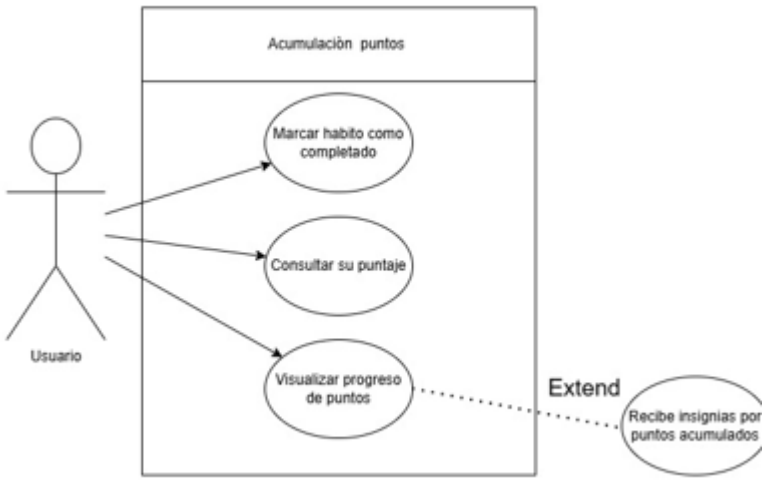
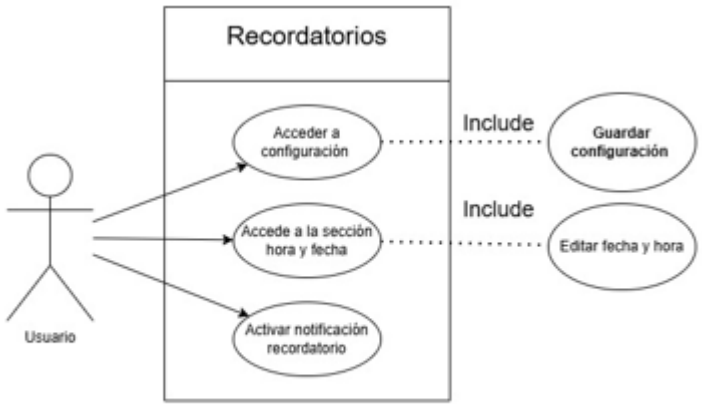
Trabajo Individual
Nikol Tatiana Ortiz
Caso de Estudio FitLife
Según el trabajo repartido por mi líder se me asigno los casos de uso Acumulación de puntos, y recordatorios.

Caso de uso Acumulación de puntos





Requerimientos funcionales

Acumulación de puntos:

- Otorgar puntos por cada hábito cumplido.
- Registrar puntos totales del usuario.
- Mostrar nivel o clasificación basado en puntos.

Diagrama UML: La clase Puntos representa cada vez que un usuario obtiene puntos**,** con información precisa de cuándo y cuántos.

Explicación Métodos, Atributos
idPuntos (privado)

- Identificador único del registro de puntos.
- Sirve para distinguir cada registro.
- Es privado porque no debe ser modificado por el usuario directamente.

idUsuario (privado)

- Relaciona estos puntos con el usuario que los obtuvo.
- Permite filtrar los puntos por usuario en la base de datos.
- Es privado porque no debe alterarse desde fuera.

Cantidad (privado)

- Cantidad de puntos obtenidos en este registro.
- Es importante para calcular el total acumulado.
- Privado para evitar manipulaciones indebidas del puntaje.

Fecha (privado)
Fecha en que se otorgaron los puntos.

- Mostrar historial por fecha
- Estadísticas de progreso
- Ver qué hábitos fueron completados cuándo
- También es privado para mantener integridad de los datos.

CLASE HABITO:

Representa un hábito que el usuario puede completar para ganar puntos.

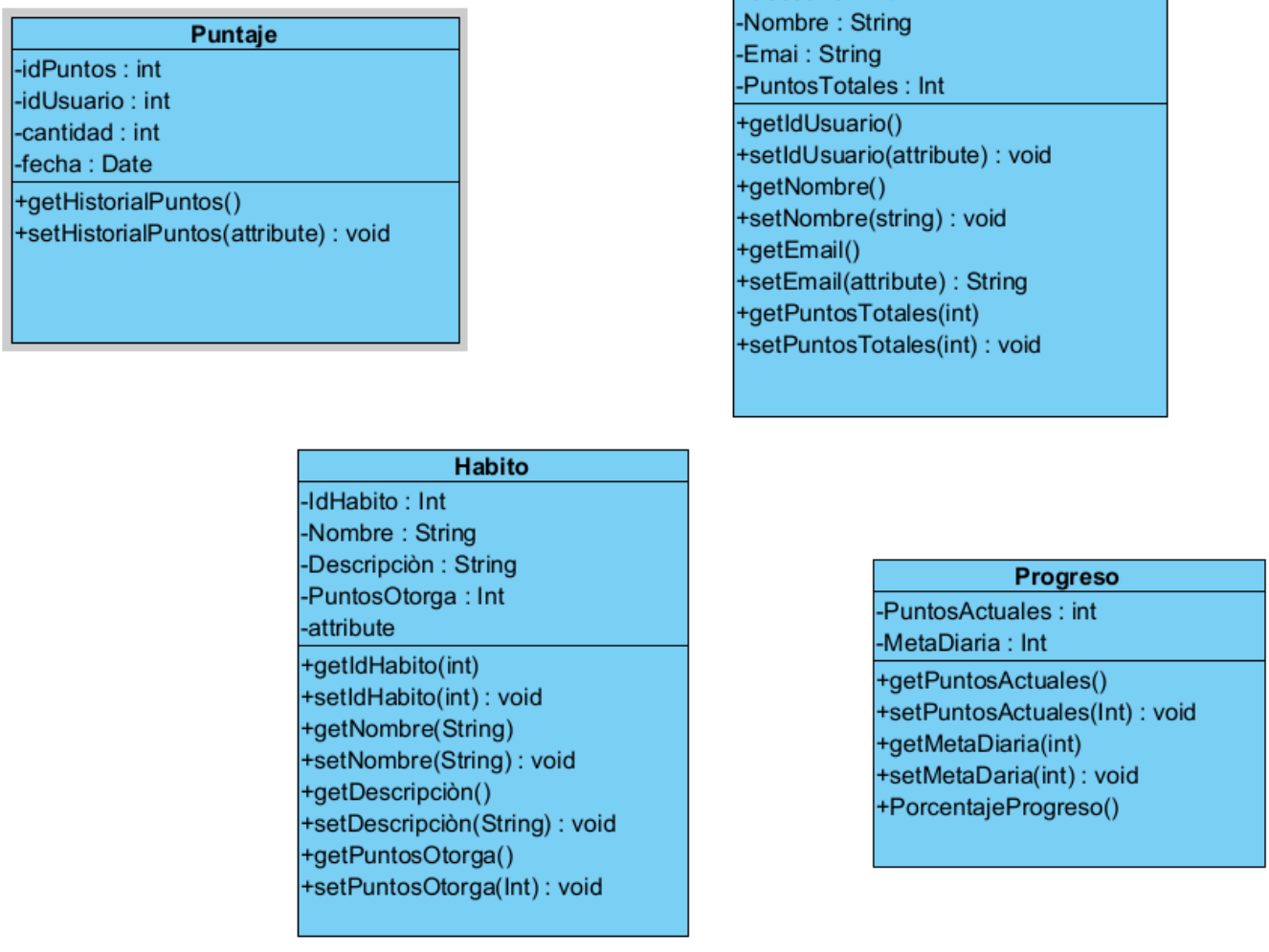
Atributos

idHabito (privado)

- Identificador único del hábito.
- Diferencia cada hábito en el sistema.

nombre (privado)

- Nombre del hábito, visible para el usuario.



descripcion (privado)

- Explicación del hábito, para que el usuario sepa qué hacer.

puntosOtorga (privado)

- Cantidad de puntos que el hábito otorga al completarlo.
- Privado para evitar manipulación de puntos.

Métodos

completar () (público)

- Marca el hábito como completado y devuelve los puntos que otorga.

getters y setters (públicos)

- Permiten acceder y modificar atributos de forma controlada.

CLASE PUNTOS

Representa cada vez que un usuario obtiene puntos, con información precisa de cuándo y cuántos.

Atributos

idPuntos (privado)

- Identificador único del registro de puntos.
- Distingue cada registro y mantiene integridad.

idUsuario (privado)

- Relaciona estos puntos con el usuario que los obtuvo.
- Permite filtrar y consultar puntos por usuario.

cantidad (privado)

- Cantidad de puntos otorgados en este registro.
- Se usa para calcular el total acumulado.

fecha (privado)

- Fecha en que se otorgaron los puntos.
- Permite ver historial y estadísticas de progreso.

Métodos

registrarPuntos () (público)

- Guarda el registro de puntos en el historial.

obtenerTotalPuntos () (público)

- Calcula el total de puntos del usuario sumando todos sus registros.

getters y setters (públicos)

- Permiten acceder y modificar los atributos de forma controlada.

CLASE PROGRESO

Representa el progreso del usuario hacia una meta específica.

Atributos

puntosActuales (privado)

- Puntos que el usuario tiene actualmente.

meta (privado)

- Puntos necesarios para alcanzar la meta o una insignia.

Métodos

porcentajeProgreso () (público)

- Calcula el porcentaje de avance: puntosActuales / meta.

getters y setters (públicos)

- Permiten acceder y modificar los atributos de forma segura.