

Requerimientos no funcionales

1. Hay una base de datos donde se almacenan (crear) los planes de alimentación
2. Hay una base de datos donde se modifican los planes de alimentación existentes
3. Hay una base de datos donde se pueden eliminar los planes de alimentación existentes
4. Hay una base de datos donde se almacenan (crear) los planes de ejercicio
5. Hay una base de datos donde se modifican los planes de ejercicios existentes
6. Hay una base de datos donde se pueden eliminar los planes de ejercicios existentes
7. Hay una base de datos donde se almacenen (crear) los anuncios informativos
8. Hay una función almacenada que permite habilitar anuncios informativos existentes en la base de datos
9. Hay una función almacenada que permite programar los anuncios informativos existentes en la base de datos
10. Hay una función almacenada que permite eliminar los anuncios informativos existentes en la base de datos
11. Hay una función almacenada que permite modificar los anuncios informativos existentes en la base de datos

- 12.** No se permitira que el atributo username contenga caracteres especiales mediante condicionales
- 13.** Hay una base de datos donde hay un registro de todos los usuarios existentes en la plataforma
- 14.** Hay una función almacenada que permite agregar un nuevo usuario al registro de usuarios
- 15.** Hay una función almacenada que permite gestionar los atributos de los registros en la tabla de usuarios
- 16.** Hay una función almacenada que permite al usuario ingresar, si es que el usuario y contraseña proporcionado existe en la base de datos
- 17.** Hay una función almacenada que permite eliminar planes de alimentación existentes en la base de datos
- 18.** La aplicación estara prograamada de modo que, de acuerdo al atributo que indica el tipo de usuario, mostrara el menú de usuario normal
- 19.** La aplicación estara prograamada de modo que, de acuerdo al atributo que indica el tipo de usuario, mostrara el menú de administrador
- 20.** La aplicación estara prograamada de modo que, de acuerdo al atributo que indica el tipo de usuario, mostrara el menú de entrenador
- 21.** Habrá un sistema que permita la subida de archivos de tipo imagen, esto para permitir el cambio de icono/foto de perfil del usuario

- 22.** Habrá una función almacenada que le permitira al usuario cambiar su nickname (nombre visible para otros usuarios), siempre y cuando el contador de esta acción no sea mayor a 0
- 23.** El atributo de contador de veces que el usuario ha cambiado su nickname, se restablece a 0 cada 3 meses desde que se actualizo (desde que se realizo la acción)
- 24.** Cuando se agrega una cuenta nueva a la base de datos, se le asigna un ID único
- 25.** Habrá una base de datos donde se almacenen (crear) logros/emblemas
- 26.** Habrá una base de datos donde se modifiquen los atributos de los logros/emblemas existentes
- 27.** Habrá un sistema en el que, de acuerdo a los atributos del usuario, si se cumple el requisito de algún logro, se le "entregará" inmediatamente
- 28.** Habrá una base de datos donde se crearan amistades entre usuarios
- 29.** Habrá una base de datos donde se puedan modificar amistades existentes entre usuarios (tendra un atributo que se llame "Estatus", el cual seria de tipo enum, siendo los casos: "Pendiente" o "Activa".
- 30.** Habrá una base de datos donde se puedan eliminar amistades existentes entre usuarios