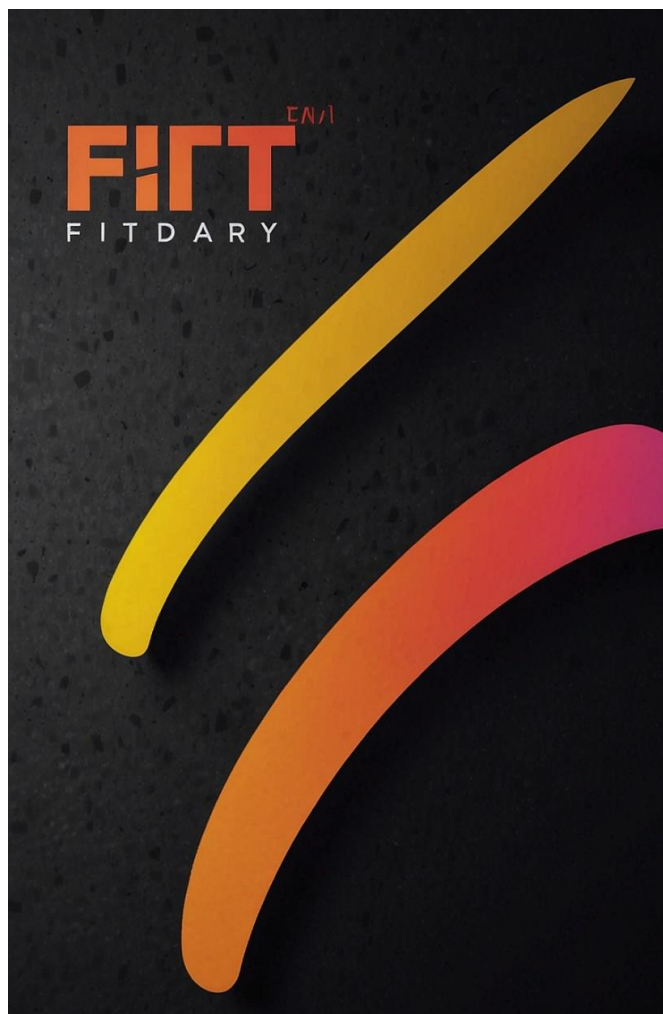


FitDiary

By

Àngel Pellicer Grau



Contenido

Introducción:	3
Herramientas y Métodos:.....	3
E/R (Entidad-Relación):	3
Paso a patrones:	4
CREATE de MongoDB:	5
UPDATE de MongoDB:.....	6
Usuarios i Roles de MongoDB:	7
CRUD (Delete) en MongoDB	8
Sketch:	8
Casos de usos 1:	9
Casos de usos 2:	9
Casos de usos 3:	9
Casos de usos 4:	9
Casos de usos 5:	10
Casos de usos 6:	10
Casos de usos 7:	10
Conclusiones:	10
Bibliografía y Webgrafía:	11
Contactos:.....	11

Introducción:

FitDiary es una aplicación con la que podrás registrar tus entrenamientos, la duración de estos y tu progreso.

Podrás ver el progreso de tus días y el tiempo invertido en este, gracias a la implementación de una base de datos que tiene la misma.

De esta forma tendrás tu propio diario de entrenamiento, en el cual veras de forma clara todo el progreso durante tus días de entrenamiento.

Herramientas y Métodos:

El proyecto comienza con la creación de un esquema conceptual de la base de datos. En lugar de convertir este esquema en tablas, definimos colecciones y documentos que se utilizarán en MongoDB. Una base de datos MongoDB, nos permite trabajar con documentos JSON-like, proporcionando una estructura flexible y dinámica.

Utilizamos la biblioteca Pymongo de Python para gestionar nuestra base de datos MongoDB. Esto nos permite realizar operaciones de inserción, actualización y consulta de manera eficiente.

Posteriormente, se desarrolla una interfaz gráfica para interactuar con la base de datos y el programa en sí. Utilizamos Tkinter, una biblioteca estándar de Python para la creación de interfaces gráficas de usuario (GUIs), que permite a los usuarios introducir, modificar y consultar datos de manera intuitiva y visualmente atractiva.

Todo el desarrollo de este proyecto se ha realizado en un entorno Windows 10. Las palabras clave como "MongoDB", "Pymongo", "Python", "Visual Studio Code" y "Tkinter" son elementos clave de este proyecto.

E/R (Entidad-Relación):

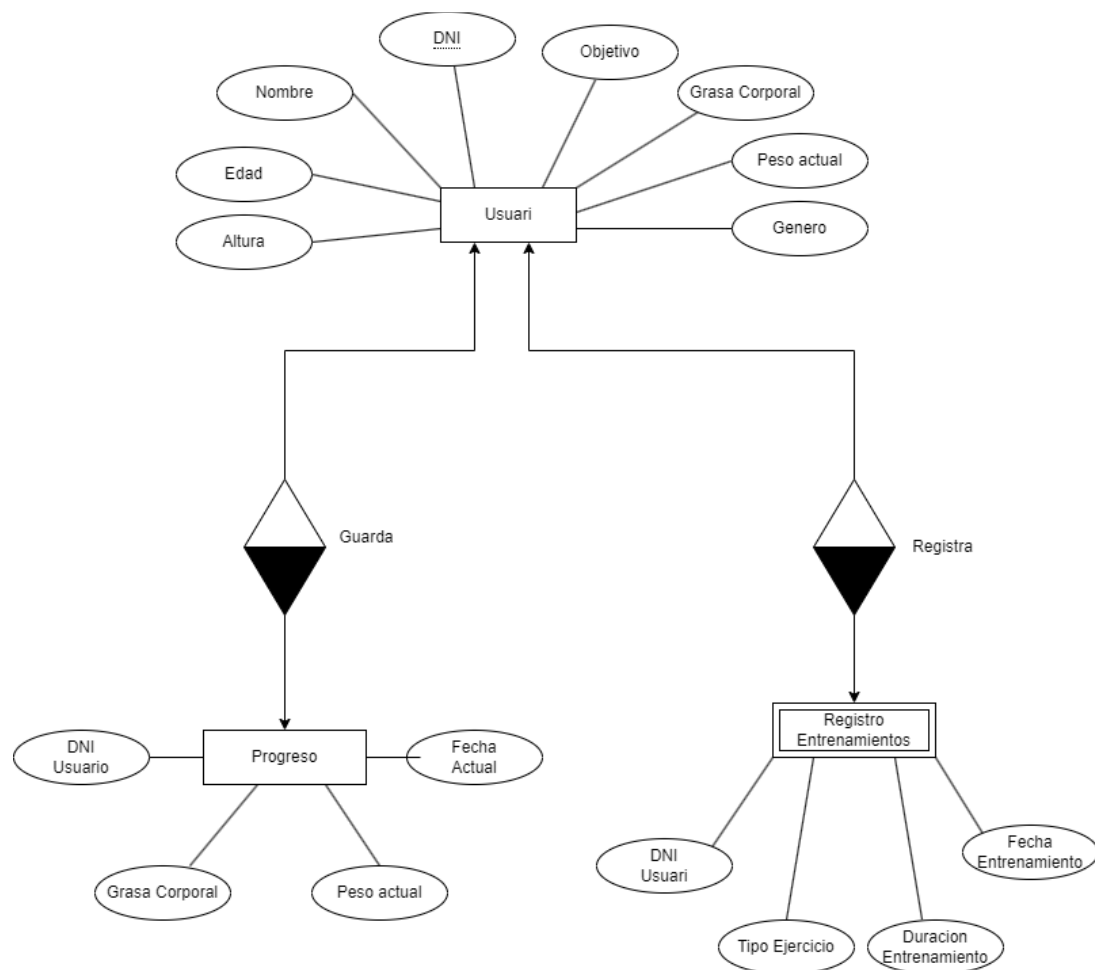
En la siguiente imagen se muestra el esquema de Entidad-Relación de FitDiary.

En la parte superior se muestra la información que tendrá el usuario, la clave primaria del usuario es el DNI.

La relación entre el usuario y el progreso es de 1:M un usuario guardara muchos progresos, los progresos necesitan el DNI del usuario, la data la grasa y el pes.

La relación entre el usuario y el registro de entrenamiento es que un usuario puede tener muchos registros de entrenamiento, pero los registros de entrenamiento están enlazados a un usuario.

La información principal que guardara el registro de entrenamiento es el DNI del usuario el tipo de entrenamiento que se ha hecho, la duración del entrenamiento y la fecha en la que se ha creado el usuario



Paso a patrones:

El patron que se usa en FitDiary es el patron de subconjunto, por diversas razones, la primera es que la relación de los datos es de 1:M.

El enfoque que se utiliza es un enfoque híbrido, ya que se implementa tanto la incorporación y la referencia.

La información se puede actualizar y luego la mostrará actualizada.

El propio documento se va actualizando y también la información que se guarda en este, para posteriormente mostrar el progreso del usuario y sus entrenamientos.

CREATE de MongoDB:

Esta función registra un nuevo usuario en la colección usuarios tras verificar que todos los campos están completos y que el nombre de usuario no existe ya en la base de datos.

```
def registrar_usuario(self):  
    usuario = self.usuario_entry_registro.get()  
    contrasena = self.contrasena_entry_registro.get()  
    dni = self.dni_entry_registro.get()  
    edad = self.edad_entry_registro.get()  
    genero = self.genero_entry_registro.get()  
    peso = self.peso_entry_registro.get()  
    grasa = self.grasa_entry_registro.get()  
  
    if not usuario or not contrasena or not dni or not edad or not genero or not peso or not grasa:  
        messagebox.showerror("Error", "Todos los campos son obligatorios.")  
        return  
  
    if usuarios_collection.find_one({"usuario": usuario}):  
        messagebox.showerror("Error", "El nombre de usuario ya existe.")  
        return  
  
    usuarios_collection.insert_one({  
        "usuario": usuario,  
        "contrasena": contrasena,  
        "dni": dni,  
        "edad": edad,  
        "genero": genero,
```

```
"peso": peso,  
"grasa": grasa })
```

```
messagebox.showinfo("Éxito", "Usuario registrado exitosamente.")  
self.mostrar_inicio_frame()
```

READ de MongoDB:

Esta función permite al usuario iniciar sesión verificando su nombre de usuario y contraseña en la colección usuarios.

```
def iniciar_sesion(self):  
    if not wait_for_mongodb_connection():  
        messagebox.showerror("Error", "No se pudo establecer conexión con la base de datos.")  
        return  
  
    usuario = self.usuario_entry_inicio.get()  
    contrasena = self.contrasena_entry_inicio.get()  
  
    user = usuarios_collection.find_one({"usuario": usuario, "contrasena": contrasena})  
  
    if user:  
        self.usuario_actual = user  
        self.mostrar_bienvenida_frame()  
    else:  
        messagebox.showerror("Error", "Usuario o contraseña incorrectos.")
```

UPDATE de MongoDB:

Esta función actualiza los datos personales del usuario actual en la colección usuarios.

```
def guardar_datos_modificados(self):  
    dni = self.dni_entry_modificar.get()  
    edad = self.edad_entry_modificar.get()  
    genero = self.genero_entry_modificar.get()
```

```
peso = self.peso_entry_modificar.get()
grasa = self.grasa_entry_modificar.get()
```

```
usuarios_collection.update_one(
    {"_id": self.usuario_actual["_id"]},
    {"$set": {
        "dni": dni,
        "edad": edad,
        "genero": genero,
        "peso": peso,
        "grasa": grasa
    }}
)
```

```
messagebox.showinfo("Éxito", "Datos personales modificados exitosamente.")
self.mostrar_bienvenida_frame()
```

Usuarios i Rols de MongoDB:

Este ejemplo muestra cómo gestionar usuarios y roles, aunque no está implementado en el código proporcionado.

```
// Crear un usuario administrador
db.createUser({
    user: "admin",
    pwd: "adminpassword",
    roles: [{ role: "userAdminAnyDatabase", db: "admin" }]
});

// Crear un usuario con permisos de lectura/escritura en 'fitdiary'
db.createUser({
    user: "fitdiaryUser",
    pwd: "userpassword",
    roles: [{ role: "readWrite", db: "fitdiary" }]
});
```

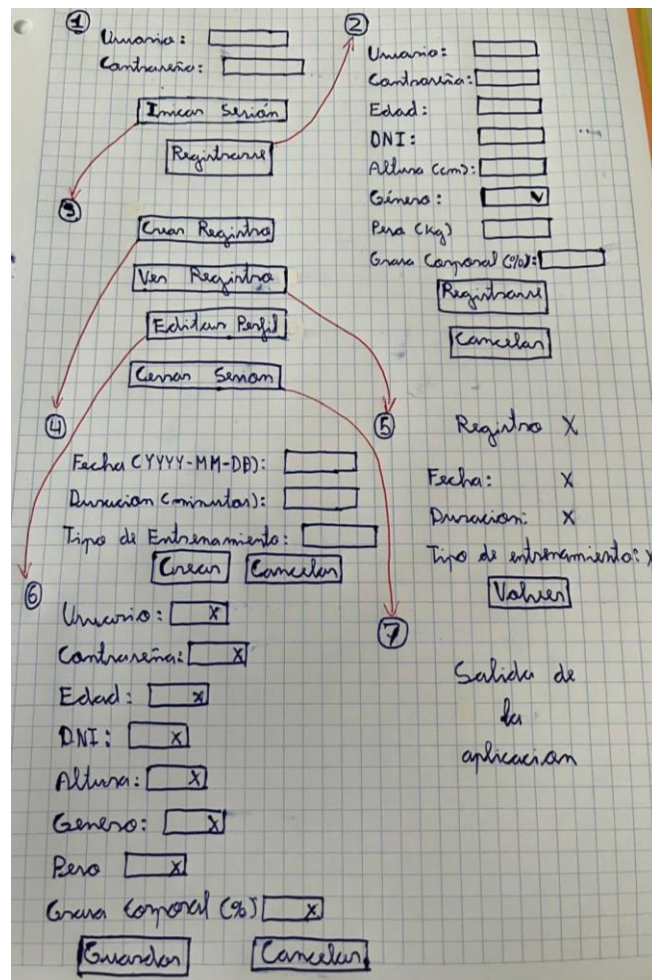
CRUD (Delete) en MongoDB

Esta función elimina al usuario actual de la colección usuarios usando su identificador único (_id).

```
def eliminar_usuario(self):  
    usuario_id = self.usuario_actual["_id"]  
    usuarios_collection.delete_one({"_id": usuario_id})  
    messagebox.showinfo("Éxito", "Usuario eliminado exitosamente.")  
    self.mostrar_inicio_frame()
```

Sketch:

En la siguiente imagen se muestra un sketch del funcionamiento de la aplicación FitDiary, con este sketch se puede ver de una forma rápida todas las acciones de la aplicación.



Casos de usos 1:

Lo que se muestra en el primer punto de la aplicación es menú de inicio de sesión, en el que se pueden hacer dos cosas, la primera que es iniciar sesión si ya tenemos una cuenta creada en la aplicación, o la segunda que es la que tendrá que hacer un usuario nuevo, que es crear un usuario.

Esta última opción nos lleva al segundo caso de uso.

Casos de usos 2:

En este punto se le pide al usuario nuevo una serie de datos personales para poder crear su usuario, los datos que se pedirán son los siguientes:

- Nombre de Usuario
- Contraseña
- Edad
- DNI
- Altura (cm)
- Genero
- Peso (kg)
- Grasa corporal (%)

También tendrá la opción de cancelar este caso de uso o por lo contrario registrarse, una vez registrado la aplicación de forma automática entrará en el menú de bienvenida.

Casos de usos 3:

En el menú de bienvenida nos muestra 4 opciones que son las siguientes:

- Crear Registro
- Ver Registros
- Editar Perfil
- Cerrar Sesión

De esta forma el usuario elige la opción que más necesita en ese momento

Casos de usos 4:

En este caso vamos a representar en el momento en el que el usuario decide crear un registro de entrenamiento. Una vez seleccionada esta opción nos pide que indiquemos 3 cosas.

Fecha

Duración de entrenamiento

Tipo de entrenamiento

Luego de esto nos muestra dos opción es, la primera que es la de guardar el registro de entrenamiento o la de cancelar el registro, una vez elegida la opción nos moverá al menú de bienvenida.

Casos de usos 5:

En el caso de uso de ver los registros como el propio nombre indica nos muestra una lista de los registros que tenemos creados de forma ordenada, si el usuario no tiene ningún registro de entrenamiento no nos mostrara nada. También tiene un botón para poder volver al menú de bienvenida.

Casos de usos 6:

En el caso de uso de editar el perfil como el nombre indica nos permite modificar i cambiar todos los datos proporcionados por el usuario desde el momento de crear la cuenta, luego de esto se actualizará la base de datos.

Casos de usos 7:

En el caso de uso de cerrar sesión cerrará la sesión de un usuario en ese momento y nos moverá al menú de inicio, donde podremos iniciar sesión con otro usuario o crear un usuario como hemos mostrado antes.

Conclusiones:

Resumen de los resultados obtenidos.

El programa FitDiary es un programa creado para la gente que le gusta guardar su progreso para una mejor motivación y ver como progresa.

De esta forma el mismo usuario se motiva para seguir utilizando la aplicación y también se motiva a seguir aumentando los entrenamientos.

Reflexiones sobre el proceso y posibles mejoras futuras.

El programa FitDiary se podría mejorar tanto en interfaz grafica como en algunos aspectos para facilitar a los Usuarios la forma de usarlo.

También podría implementar una apartado de dietas con los usuarios Prime, etc.

Y también se le podría implementar una serie de niveles al usuario, de esta forma cuando un usuario lleve una racha de días de entrenamiento se multiplica la experiencia que obtendrá a partir de cada entrenamiento.

Bibliografía y Webgrafía:

Gracias a las siguientes fuentes de información ha sido posible la creación de la aplicación de FitDiary.

- YouTube.
- Apuntes de Mongo DB
- Paginas tutoriales de Mongo DB
- PDF de Mongo DB

Contactos:

Gmail: pellicergrauangel@gmail.com

Tel: +34 684 454 683

GitHub: <https://github.com/PellicerAngel/Fitdiary-MongoDB.git>

Realizado por ÀNGEL PELLICER GRAU