



Manual Técnico

Alejandro Leal Astorga

INDICE

1. Introducción.....	4
2. Análisis del problema	5
2.1 Problemática	5
2.2 Clientes potenciales	5
2.3 Análisis DAFO	5
2.4 Monetización y benéficos	6
3. Diseño de la solución	6
3.1 Tecnologías elegidas	6
3.1.1 Aplicación move	6
3.1.2 API MoveAPI	6
3.1.3 MailJS	6
3.1.4 Google Maps cloud platform	6
3.2 Arquitectura	7
3.3 Diagrama de clases	7
3.4 Diagrama E/R	8

3.5 Consideraciones técnicas	8
3.5.1 BBDD	8
3.5.2 API	9
3.5.2 APP move	13
4. Documentación de la solución	14
5. Enlaces de interés	14

1. Introducción

Move es una red social pensada y diseñada para establecer una relación social entre los usuarios a través del deporte.

En move podrás crear tu perfil, añadir imágenes a tu perfil que podrás mostrar al resto de los usuarios, podrás seguir a los demás usuarios para poder ver sus perfiles e imágenes y tendrás multitud de eventos deportivos en los que podrás participar solo o con tus amigos.

Move dispone de un chat para todos los participantes del evento donde se establece dicha relación social antes de que se realice en el evento, y poder resolver dudas sobre el evento tales como reservas si es necesaria para el evento o materiales deportivos, etc.

Si el usuario participa en los eventos se cumplirá el propósito por el cual se diseñó esta aplicación, conocer gente y hacer nuevos amigos por medio del deporte.

- La aplicación se compone de tres bloques:

PERFILES

AMIGOS

EVENTOS

2. Análisis del problema

2.1 Problemática

Las redes sociales que todo el mundo conoce suelen ser para compartir videos imágenes, ligar, etc. Pero ninguna de ellas incita a sus usuarios a salir a la calle y moverse, simplemente quieren tenerte todo el día mirando el móvil y gastando tu tiempo mirando su aplicación.

Nosotros queremos hacer justo lo contrario queremos que conozcas a gente nueva incitando a los usuarios a salir a la calle mientras realiza una actividad sana como es el deporte.

2.2 Clientes potenciales

La aplicación va dirigida a personas de todas la edades ya que el deporte no entiende de edades y todo el mundo puede realizarlo, si es verdad que la aplicación tiene un limite de edad de 16 años, esto se debe a que es una edad adecuada para que una persona joven pueda relacionarse a través de una red social.

2.3 Análisis DAFO



2.4 Monetización y beneficios

No se pretende cobrar por la aplicación pero si que se necesita un poco de beneficios para mejorar la aplicación y el equipo de desarrollo, por lo que se podría pensar en hacer patrocinios de algunas marcas deportivas en la aplicación para obtener dicho beneficio, pero no se contempla hacer planes de pago ni cobrar por la aplicación.

3. Diseño de la solución

3.1 Tecnologías elegidas

3.1.1 Aplicación move

- **IDE** : Visual Studio Code
- **Lenguaje**: Dart
- **Framework**: Flutter

3.1.2 API MoveAPI

- **IDE** : Visual Studio Community 2022
- **Lenguaje**: C#
- **Framework**: ASP .NET Web API

3.1.3 MailJS

- **API** : servicio de envío de emails

3.1.4 Google Maps cloud platform

- **API** : consumo de la api de Google Maps para Android

3.2 Arquitectura

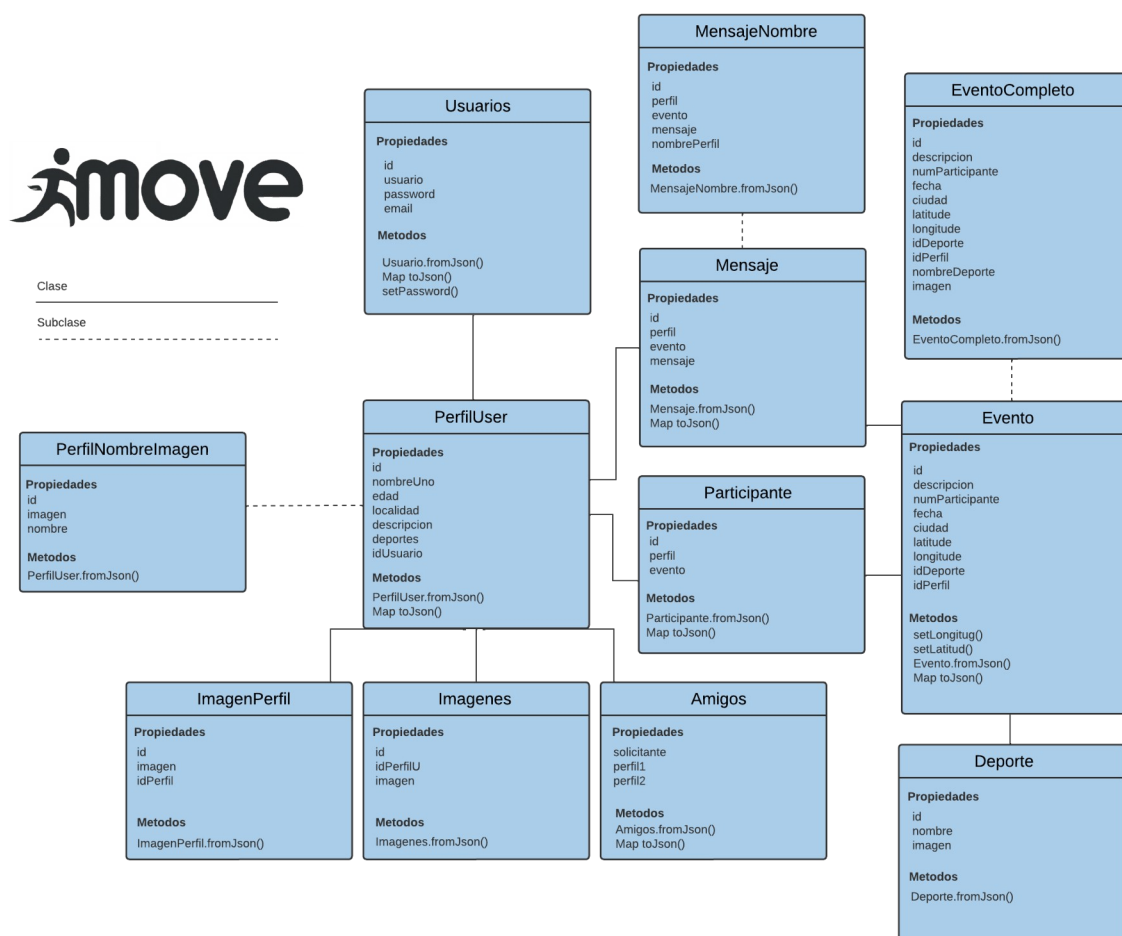
Cliente – Servidor

Esta arquitectura se compone de dos partes:

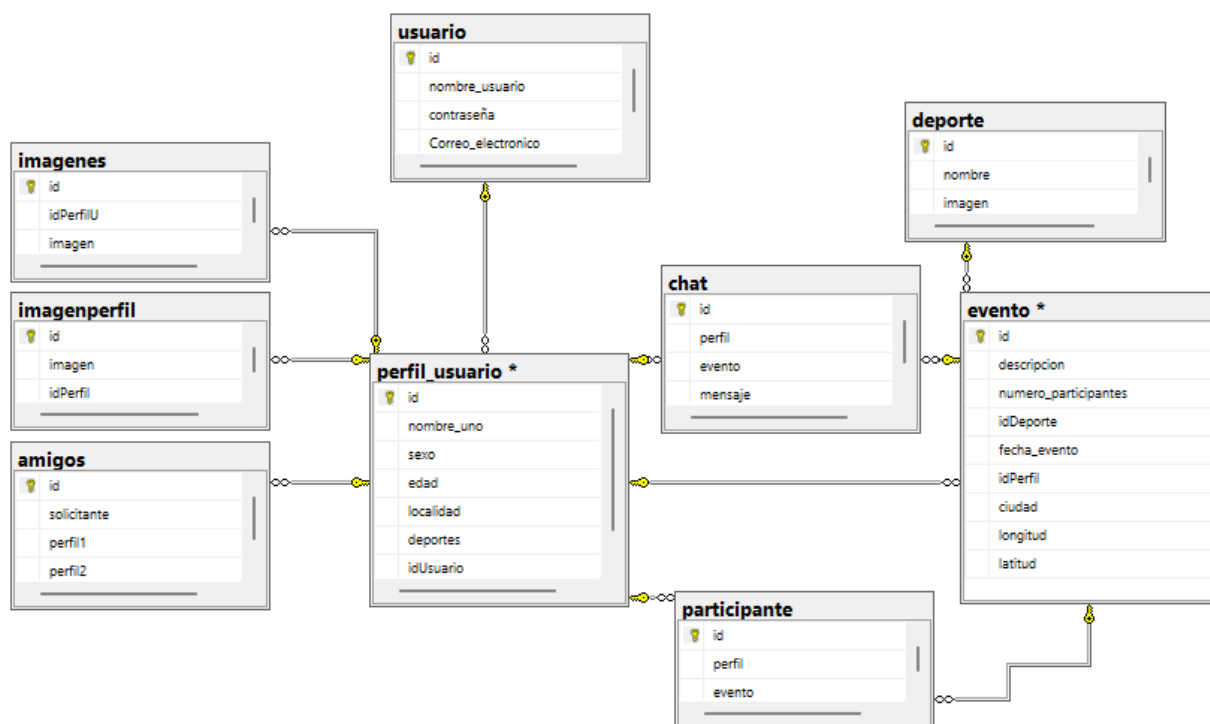
Cliente: La parte del cliente es move que se ejecuta en el dispositivo móvil del usuario y se comunica con la API para obtener y procesar datos.

Servidor: La parte del servidor es la API, que proporciona acceso a un conjunto de datos o funciones a través de una interfaz web. El servidor procesa las solicitudes del cliente y devuelve los datos solicitados en un formato estandarizado JSON.

3.3 Diagrama de clases



3.4 Diagrama E/R



3.5 Consideraciones técnicas

3.5.1 BBDD

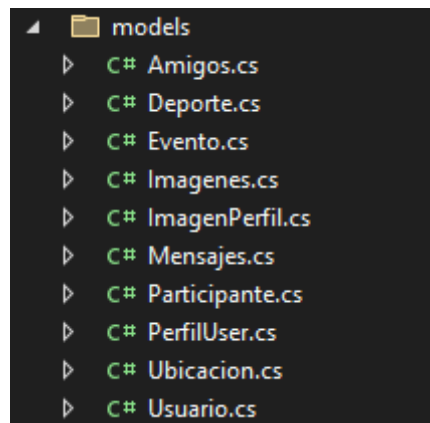
La base de datos que estamos utilizando en el proyecto es una base de datos SQL Server 2022, la sintaxis es muy parecida a MySQL por lo que podrás utilizarlo en ambos casos o en la base de datos de tu hosting.

En el fichero del manual técnico hay un fichero con toda la creación de tablas, insertar valores de la tabla deporte, y un procedure que utilizamos para eliminar todos los datos del usuario cuando desea eliminar su perfil completo.

3.5.2 API

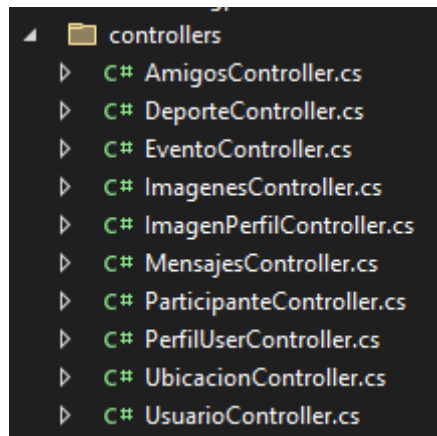
Para usar la API en modo local necesitas instalar Visual Studio e importar el proyecto MoveAPI, el proyecto se divide en tres partes:

- **Modelos:** modelos que utilizamos para el tratamiento de datos con la bbdd.

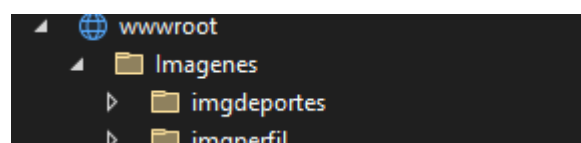


-Controladores:

controladores de la API que contiene los métodos donde implementaremos los verbos que actuarán con la base de datos (GET,POST,PUT,DELETE).



- **wwwroot:** contiene los ficheros donde se almacenarán las imágenes de los usuarios y las imágenes de los deportes, que se verán al crear el evento, cada deporte tiene su imagen.

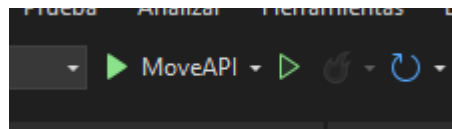


Para que el proyecto funcione con tu base de datos debes cambiar la propiedad "connection" de el archivo appsetting.json que debe apunta a la conexión de tu base de datos.

```
"Logging": {
  "LogLevel": {
    "Default": "Information",
    "Microsoft.AspNetCore": "Warning"
  }
},
"AllowedHosts": "*",
"ConnectionStrings": {
  "connection": "workstation id=Apimove.mssql.somee.com;packet size=4096;user id=Alela677_SQLLogin_1;pwd=ngd8u4apy"
```

3.5.3 Peticiones a la API

Para probar los métodos de la API Visual Studio ofrece un entorno de pruebas llamado Swagger al que podemos acceder con el botón del play verde que podemos encontrar en la barra de herramientas en la parte superior.



Usuarios:

GET	/api/Usuario	▼
GET	/api/Usuario/{id}	▼
GET	/api/Usuario/comprobar/{email}	▼
GET	/api/Usuario/{usuario}/{password}	▼
POST	/api/Usuario/insert	▼
PUT	/api/Usuario/update	▼
DELETE	/api/Usuario/delete/{id}	▼

Perfil usuario:

POST	/api/PerfilUser/insert	▼
GET	/api/PerfilUser	▼
GET	/api/PerfilUser/perfiles/{id}	▼
GET	/api/PerfilUser/misamigos/{id}	▼
GET	/api/PerfilUser/{id}	▼
GET	/api/PerfilUser/idUser/{idUser}	▼
PUT	/api/PerfilUser/update	▼
DELETE	/api/PerfilUser/delete/{id}	▼
DELETE	/api/PerfilUser/deletePerfil/{idUser}/{idPerfil}	▼

Evento:

POST	/api/Evento/insert	▼
GET	/api/Evento	▼
GET	/api/Evento/activos/{id}	▼
GET	/api/Evento/activos/user/{id}	▼
GET	/api/Evento/participando/{id}	▼
GET	/api/Evento/{id}	▼
PUT	/api/Evento/update	▼
DELETE	/api/Evento/delete/{id}	▼
GET	/api/Evento/ultimo/{id}	▼

Imágenes:

GET	/api/Imagenes/viewimage/{imagenname}	✓
GET	/api/Imagenes/userimages/{id}	✓
GET	/api/Imagenes/{id}	✓
POST	/api/Imagenes/insert	✓
PUT	/api/Imagenes/update	✓
DELETE	/api/Imagenes/delete/{id}	✓

Imagen perfil:

POST	/api/ImagenPerfil/insert	✓
DELETE	/api/ImagenPerfil/delete/{id}	✓
GET	/api/ImagenPerfil/idPerfil/{id}	✓
GET	/api/ImagenPerfil/viewimage/{imagenname}	✓

Deporte:

POST	/api/Deporte/insert	✓
GET	/api/Deporte/viewimage/{imagenname}	✓
GET	/api/Deporte	✓
GET	/api/Deporte/{id}	✓
PUT	/api/Deporte/update	✓
DELETE	/api/Deporte/delete/{id}	✓

Participante:

POST	/api/Participante/insert	✓
GET	/api/Participante	✓
GET	/api/Participante/numParticipantes/{id}	✓
GET	/api/Participante/idUsuario(id:int	✓
GET	/api/Participante/participantesEvento/{id}	✓
DELETE	/api/Participante/delete/{idEvento}/{idPerfil}	✓

Amigos:

GET	/api/Amigos	▼
POST	/api/Amigos/insert	▼
DELETE	/api/Amigos/delete/{id}/{idAmigo}	▼

Mensajes:

GET	/api/Mensajes/{idEvento}	▼
POST	/api/Mensajes/insert	▼
DELETE	/api/Mensajes/delete/{idPerfil}	▼

3.5.3 APP move

Para importar el código de la aplicación necesitas tener Visual Studio Code, Android Studio para crear un emulador donde poder compilar y ver los resultados de la aplicación, también debes instalar las extensiones de Dart y Flutter en Visual Studio Code, también necesitas instalar el SDK de Dart en tu PC y el SDK de Flutter, y añadir la ruta del fichero bin del SDK de Flutter a tus variables de entorno.

Cuando tengas todo el entorno en funcionamiento puedes ejecutar el comando "flutter doctor" o "flutter doctor -v" que realizara un chequeo para comprobar que todo esta bien configurado.

Dirígete a la barra de herramientas "Archivos" → "Abrir carpeta" he importa el proyecto, puede que te de fallos con los "packages" puedes solucionar esto ejecutando los comando "flutter clean" y luego "flutter pub get" para limpiar y volver a reinstalar los paquetes.

También puede ser que encuentres un inconveniente con la versión de flutter puedes poner manualmente la version de SDK que quieras utilizar o ejecutar el comando "flutter upgrade" para actualizar o "flutter downgrade" para bajar la versión de flutter.

4. Documentación de la solución

Todos los documento, ficheros los puedes encontrar en un repositorio de mi cuenta de GitHub el nombre del repositorio es "TFG_move" el repositorio se divide en 3 ramas.

- **Main:** contiene todos lo fichero de la aplicación
- **API :** contiene todos los ficheros de la API y el archivo MoveAPI.sln
- **Imágenes:** contiene las imagenes que se utilizaron para el logo, y para los deportes. Son imágenes sin copyright, el logo fue generado por una IA y las imagenes de los deportes se han conseguido en Freepik
- **Documentación:** documentación técnica y documentación de usuario.

5. Enlaces de interés

Mantén Ctrl + clic sobre el texto para acceder a los enlaces.

[Repositorio GitHub](#)

[Descargar Visual Studio](#)

[Descargar Visual Studio Code](#)

[SDK Dart](#)

[SDK Flutter](#)

[Tutorial WEB API](#)

[Freepik](#)

[DALL - E IA imagenes](#)