

PADELMATCH



Autor: Andreu Bustos Aznar

Ciclo: Desarrollo Aplicaciones Multiplataforma

Tutor: Joaquín Vicente Alonso Saiz

Centro: I.E.S La Vereda

IES LA VEREDA



Resumen

PadelMatch es una App para reservar pistas de pádel, orientada a un futuro donde poder gestionar clubs españoles de forma intuitiva sincronizar las reservas de las pistas entre servicios externos y los nuestros para que el club tenga la máxima ocupación de sus instalaciones y así maximizar sus beneficios.

Resum

PadelMatch és una aplicació per a reservar pistes de pàdel, orientada a un futur on poder gestionar clubs espanyols de forma intuïtiva. Sincronitza les reserves de les pistes entre serveis externs i els nostres perquè el club tinga la màxima ocupació de les seues instal·lacions i així maximitzar els seus beneficis.

Abstract

PadelMatch is an app for booking padel courts, oriented to a future where you can manage Spanish clubs intuitively synchronize the booking of the courts between external services and ours so that the club has the maximum occupancy of its facilities and thus maximize their profits.

Índice

1. INTRODUCCION.....	6
1.1. Explicación	6
1.2. Funcionalidad de la App	6
1.3 Módulos/Asignaturas	7
2. PLAN DE TRABAJO.....	8
3. DISEÑO.....	10
3.1. Modelo Entidad / Relación.....	11
4. IMPLANTACIÓN.....	12
4.1. Base de datos.....	12
4.2. DDL	13
4.2.1. Procedimientos	13
4.2.2. Secuencias	15
4.3. DML.....	16
4.3.1. Backup.....	17
4.4. API PADELMATCH	17
4.5. APLICACIÓN MÓVIL.....	21
4.5.1. CARPETAS.....	22
4.5.2. ENDPOINTS	23
5. TECNOLOGÍAS UTILIZADAS	23
6. RECURSOS	24
6.1. Software	24
6.2. Hardware	25
6.3. Sistemas Operativos	25
6.4. Equipo	25
7. PUBLICACIÓN	25
8. CONCLUSIONES.....	25
8.1. GRADO DE CONSECUCIÓN DE OBJETIVOS.....	26

8.2. PROBLEMAS ENCONTRADOS	27
8.3. MEJORAS	29
9. ANEXOS Y DOCUMENTACIÓN COMPLEMENTARIOS.....	30
10. REFERENCIAS BIBLIOGRAFÍA	31
11. ÍNDICE DE IMÁGENES.....	32

1. INTRODUCCION

1.1. Explicación

Reservas PadelMatch, encuentra tu pista.

PadelMatch actualmente se centra en una aplicación para la creación de reservas en clubs asociados, aquellos que quieran reservar una pista solo tendrán que entrar en la aplicación y reservar la pista que esté disponible a la hora que escojan.

En un futuro cercano haremos las integraciones necesarias para estar también accesibles a través de nuestra web y sin dejar de lado la parte de negocio donde obtendremos un pequeño porcentaje por la visibilidad y gestión de las reservas de los clubs.

Tenemos una documentación para servicios externos para poder llevar un control de reservas eficaz y sin fallos.

1.2. Funcionalidad de la App

Nuestra aplicación empieza con una animación ya que la primera impresión con un usuario es muy importante puede dar pie a su fidelización y uso recurrente.

Posteriormente tenemos un inicio de sesión bonito y sencillo con registro incluido en el que buscamos que el usuario no sienta rechazo, también disponemos de un inicio anónimo para así dar pie a que el usuario explore sin ningún tipo de impedimento.

Se podrán buscar clubs por provincia con una fecha y hora determinadas por el usuario, estos clubs tendrán pistas que a su vez tendrán horas disponibles para ser reservadas.

Los usuarios tendrán un perfil donde podrán ver sus estadísticas y datos de cuenta, también si el usuario es el dueño de un club hablaremos con él y daremos el rol de "Admin" para que así pueda crear sus pistas.

1.3 Módulos/Asignaturas

En este proyecto se han utilizado los conocimientos obtenidos gracias al ciclo Superior de Desarrollo Aplicaciones Multiplataforma y, los que he obtenido trabajando como programador en Conmuta Soluciones empresa donde realicé las prácticas del Superior.

- **SISTEMAS INFORMÁTICOS**

Conocimientos técnicos en sistemas operativos para poder publicar los diferentes proyectos dentro del TFC.

Se utiliza un Windows Server 2021, Virtual Point Network (FortiClient), el acceso remoto de Windows y el desarrollo desde Windows 10 pro en la sala de desarrollo.

- **BASES DE DATOS**

Creación de base de datos desde PgAdmin4 con PostgreSQL 16.

Utilización del software DataGrip 2023.3.4 el cual se ha utilizado para una gestión más avanzada de los schemas, tablas, secuencias, datos y procedimientos de la App.

Gracias a esta asignatura he podido realizar el modelo entidad relación de PadelMatch.

- **PROGRAMACIÓN**

Módulo el cual me ayudó a saber resolver problemas, que hubiera y me encontrara por el camino, analizarlos y desfragmentarlos en problemas más pequeños para así poder resolver lo planteado en un principio.

También mencionar el aprendizaje sobre el uso de objetos, clases, enums, structs, paquetes NuGet y dependencias.

- **LENGUAJE DE MARCAS**

Aprendizaje de programación web uso de html, css, php y, otros lenguajes más con los cuales he podido desarrollar incluso proyectos propios externos.

Mencionar el uso de json y xml el cual me ha proporcionado conocimientos para el desarrollo con diferentes estructuras de datos y la utilización de padres e hijos.

- **ENTORNOS DE DEASARROLLO**

Aprendizaje sobre SCRUM y Agile, el cual se ha utilizado para desarrollar el proyecto y tener progresos semanales.

- **ACCESO A DATOS**

Utilización diaria del apartado Data Acces Layer (DAL) con el cual se han podido gestionar el acceso, actualización, eliminación y actualización sobre los datos almacenados o recuperados.

- **DESARROLLO DE INTERFACES**

El desarrollo de la Interfaz de Usuario (UI) de PadelMatch se ha puesto especial énfasis en que el diseño sea intuitivo, atractivo y eficiente. A continuación, se detallan los aspectos clave del desarrollo de interfaces en PadelMatch utilizando JavaScript y React Native.

- **PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES**

Llamadas asíncronas para el correcto funcionamiento y control de concurrencia de los diferentes métodos.

- **PROGRAMACIÓN DE SERVICIOS Y PROCESOS**

Despliegue de la aplicación, creación dela apk y métodos async para el correcto funcionamiento y la concurrencia de los diferentes Endpoints.

- **SISTEMAS DE GESTIÓN EMPRESARIAL**

Se generan las reservas en la propia aplicación.

2. PLAN DE TRABAJO

Para poder realizar este proyecto se han utilizado las habilidades aprendidas tanto dentro del curso como en el trabajo y los objetivos que quería abarcar son:

Objetivo 1: Conseguir la implementación de un sistema de reservas de pistas de pádel con el requisito de al menos tener acceso a internet.

Objetivo 2: Poder proporcionar al propio Club/Cliente una gestión de sus reservas de forma sencilla.

Objetivo3: Documentación de desarrolladores para servicios externos y así tener una integración con acceso a crear reservas vía API.

Línea temporal



Kanban

ieslavereda-projects / Projects / @andbusazn Pista Padel

Search: Type to search

@andbusazn Pista Padel

Backlog Team capacity Current iteration Roadmap My items New view

Assignees: No Assignees

Filter by keyword or by field

Title	Status
No Priority 11 Estimate: 0	
1 Modelo ER BBDD	Done
2 HomeScreen(Buscador Clubs)	Done
3 LoginScreen actualizar	Done
4 HorasScreen	Done
5 FormationScreen	Done
6 PagarScreen	Done
7 PistasScreen y scheduler reservas	Done
8 ReservasScreen	Done
9 DocumentacionProyecto	To Review
10 ApiPadelMatch	In Progress
11 Intento de implementación de un debugger en react native para Apps	Todo

+ Add item

3. DISEÑO

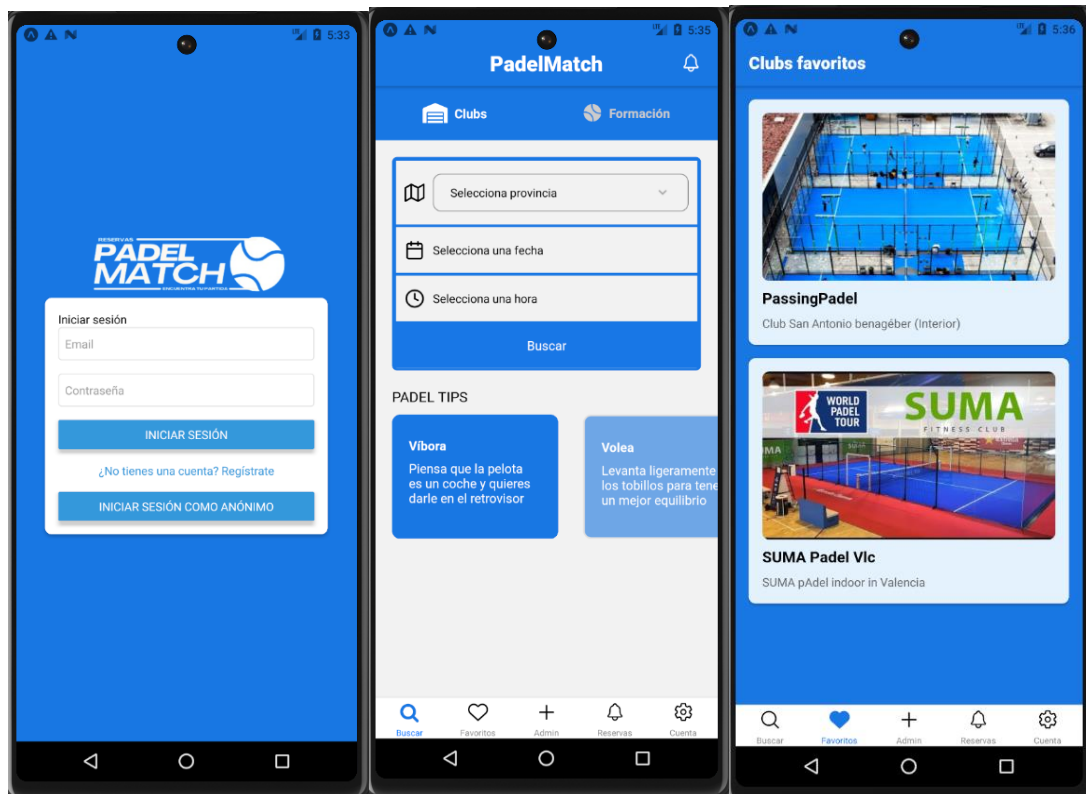
-Diseño general (parte de análisis)

- Se ha tratado de hacer una aplicación con un diseño amigable y sencillo con animaciones que resaltan el cuidado e implicación en dar una UI agradable. También comentar que se ha seguido la misma escala de colores para todo el proyecto.
- Los códigos de colores son:

	
# 3498DB	# 1877E3
	
# 1DF5DE	# E3F2FD

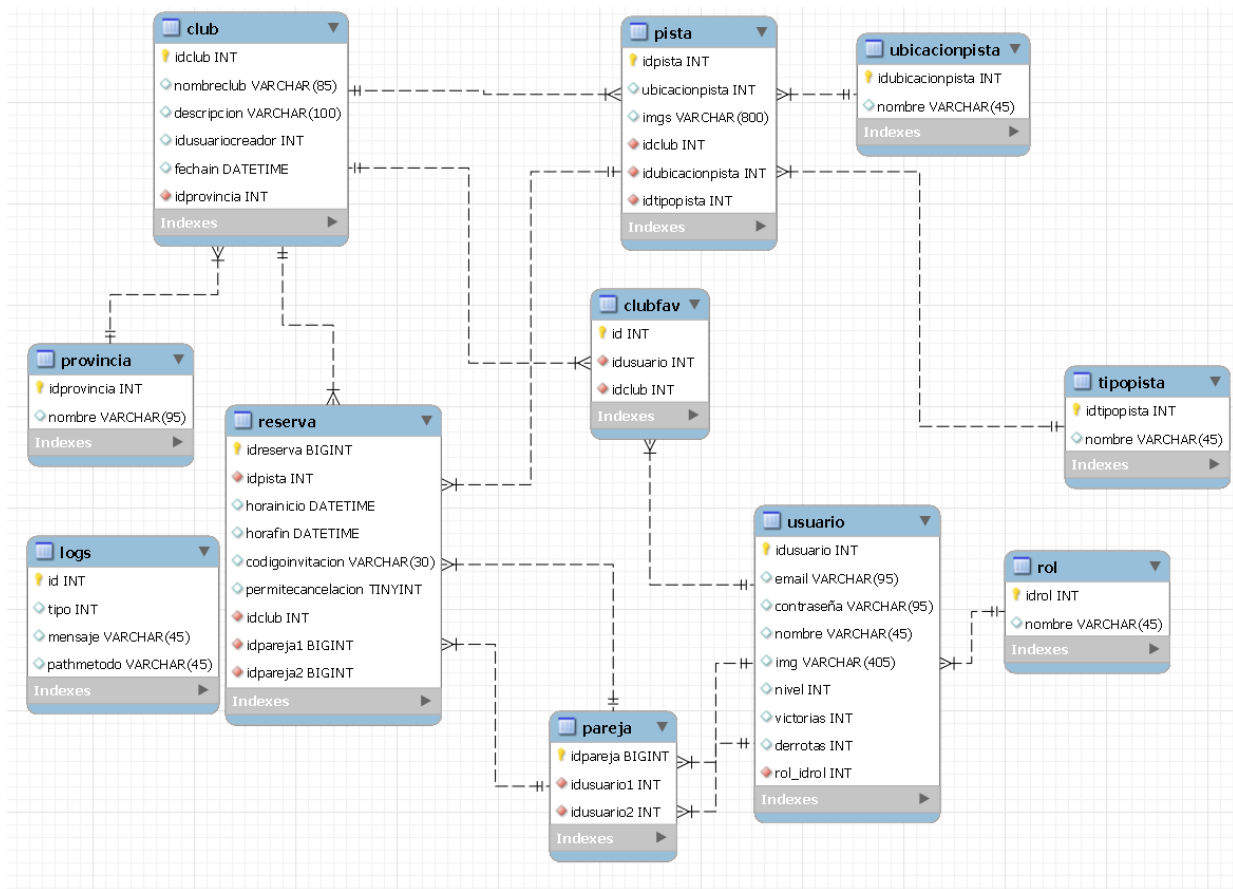
-Diseño detallado (parte de diseño)

- El diseño del logo tiene integrada una pelota, la cual identifica fácilmente hacia donde se dirige PadelMatch y con el nombre se puede intuir fácilmente el objetivo. Incluso se ha realizado una animación del mismo para así incluirla en diferentes ubicaciones donde se pueda integrar gifs o mp4.
- Como se puede ver a simple vista se ha realizado un diseño simple, además de, asignar los colores corporativos de PadelMatch.



3.1. Modelo Entidad / Relación

Una vez concluido el E/R empecé con el deep back con el cual pude desarrollar los diferentes endpoints que atacaban a la base de datos para poder trabajar con la aplicación y mostrar los datos de la BDI (Base de datos interna).



La única tabla que no está relacionada con ninguna otra es la tabla “Logs”, ya que solo se utiliza para registrar errores no contemplados que puedan aparecer en los Endpoints.

4. IMPLANTACIÓN

En este apartado se explica de forma sencilla e intuitiva todos los apartados del proyecto.

4.1. Base de datos

Instalación postgresql: postgresql-16.2-1-windows-x64.exe

Instalación pgAdmin4: pgadmin4-8.6-x64.exe gestionar las bases de datos

Instalación MySQL Workbench: MySQLWorkbench.exe desarrollar el modelo ER

4.2. DDL

El DDL se utiliza para definir y gestionar los objetos de la base de datos, como tablas.

En el repositorio existe una carpeta con el nombre de BD, que contiene todo lo relacionado con la base de datos y se encuentra el archivo de Backup y el de Scripts con el cual podrás ver todos los create table con sus columnas y tipos.

4.2.1. Procedimientos

Los procedimientos se utilizan para realizar acciones controladas sobre la base de datos, en este caso se ha creado uno para poder controlar las reservas.

```
CREATE PROCEDURE CrearReserva
```

```
    @idPista INT,
```

```
    @horaInicio DATETIME,
```

```
    @horaFin DATETIME,
```

```
    @codigoInvitacion VARCHAR(50),
```

```
    @permiteCancelacion BIT,
```

```
    @idClub INT,
```

```
    @idPareja1 INT,
```

```
    @idPareja2 INT,
```

```
    @idUsuarioReserva INT
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON;
```

```
BEGIN TRANSACTION;

BEGIN TRY

    -- Verificar si ya existe una reserva para la misma pista
    IF EXISTS (

        SELECT 1

        FROM Reserva

        WHERE idPista = @idPista

            AND @horaInicio < horaFin

            AND @horaFin > horaInicio

    )

        BEGIN

            THROW 50000, 'Ya existe una reserva para esta pista en esa hora', 1;

        END

    -- Insertar nueva reserva

    INSERT INTO Reserva (idPista, horaInicio, horaFin, codigoInvitacion,
permiteCancelacion, idClub, idPareja1, idPareja2, idUsuarioReserva)

    VALUES (@idPista, @horaInicio, @horaFin, @codigoInvitacion,
@permiteCancelacion, @idClub, @idPareja1, @idPareja2, @idUsuarioReserva);

    -- Confirmar la transacción

    COMMIT TRANSACTION;

END TRY

BEGIN CATCH

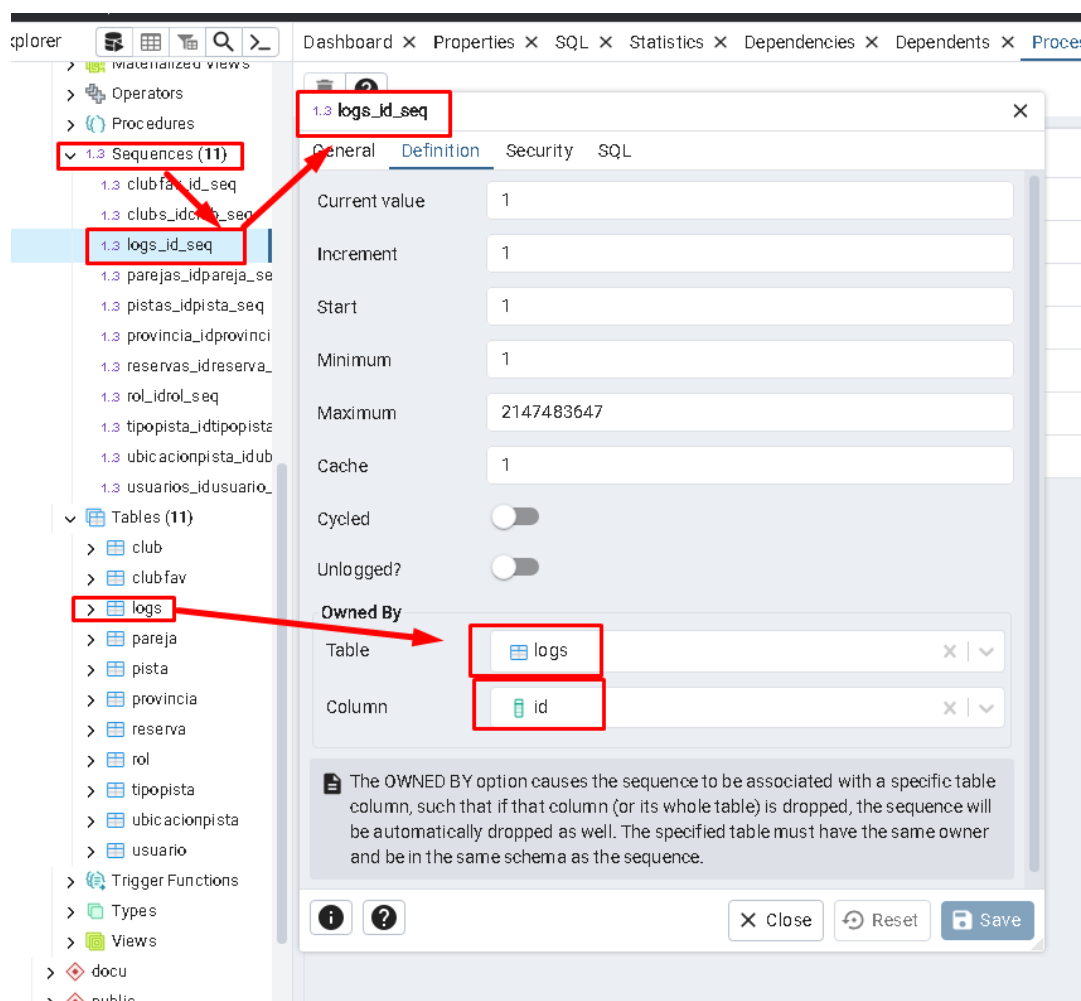
    -- En caso de error, rollback a la transacción

    ROLLBACK TRANSACTION;
```

```
-- Volver alanzar el error para que el cliente lo vea  
  
THROW;  
  
END CATCH;  
  
END;
```

4.2.2. Secuencias

La gran mayoría de secuencias creadas en las diferentes tablas están ahí porque el id principal de la tabla está creado como “serial”. El cual se queda como integer pero automáticamente se genera una secuencia y se le asigna para que sea incremental.



4.3. DML

El DML se utiliza para manipular los datos almacenados en la base de datos. Incluye operaciones que permiten insertar, actualizar, eliminar y seleccionar datos.

Aquí un par de ejemplos:

```
INSERT INTO app.rol (idrol, nombre) VALUES (DEFAULT, 'Admin');

INSERT INTO app.tipopista (idtipopista, nombre) VALUES (DEFAULT,
'Cristal_Cesped');

INSERT INTO app.ubicacionpista (idubicacionpista, nombre) VALUES (DEFAULT,
'Indoor');

INSERT INTO app.usuarios (idusuario, nombre, contraseña, mail, nivel, foto,
victorias, derrotas, idrol, idpareja) VALUES (DEFAULT, 'Andreu',
123123'andreuba13@gmail.com', 0, 'https://cdn-icons-
png.flaticon.com/512/6773/6773382.png', 0, 0, 3);
```

En la carpeta BD del repositorio el archivo insertsBD.txt contendrá todos los restantes.

También mencionar que existe un método que desde el catch lanza inserciones a la base de datos con el detalle del error, el pathmedodo el tipo de error y el mensaje del error.

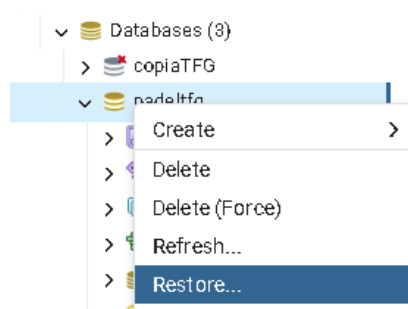
```
public enum TipoLog {
    ERROR,
    WARN,
    INFO,
    DEBUG
}
```

Ejemplo de la llamada:

```
BaseDAL.InsertarLogs(TipoLog.ERROR, $"error al buscar el mail: error =
{ex.Message}", "GetEmailExistente", 0);
```


4.3.1. Backup

He creado un Backup de la base de datos del proyecto con el cual teniendo la última versión de PgAdmin4 solo tendrás que hacer el create de la base de datos con el nombre de padeltfg (con este nombre te aseguras que todo este igual y evitas problemas de refactorizar código) y cuando ya lo has asignado haz un restore y seleccionar el archivo “padeltfg_backup.sql” y así podrás tener todos los schemas, tablas y datos por defecto con los cuales poder probar la aplicación.



4.4. API PADELMATCH

Es el proyecto de ASP.NET Core v7 donde se han desarrollado todos los Endpoints.

En él, se encuentra la carpeta Controllers la cual contiene las rutas principales,

Por el momento no se utiliza ningún tipo de servicio externo, tan solo la carpeta de servicios creada pero sin contenido.

Este es el controlador principal, ya que no hay ningún equipo, por lo que están todas las rutas declaradas aquí. Como si de un global se tratase, esto en un futuro con más tiempo se reorganizaría y estructuraría para que sea más fácil de leer.

Cuando se reestructure en un futuro el proyecto, se asignarán carpetas como Routes.

Esta carpeta contendrá RoutesPistas.cs, RoutesUsuarios.cs ... y así poder tener todo más estructurado las rutas heredarían de BaseRoute.cs y esta solo heredaría de ControllerBase.

Así todas las rutas empezarán por “/api/”

```
namespace ConectorPortugalAtena.Routes;

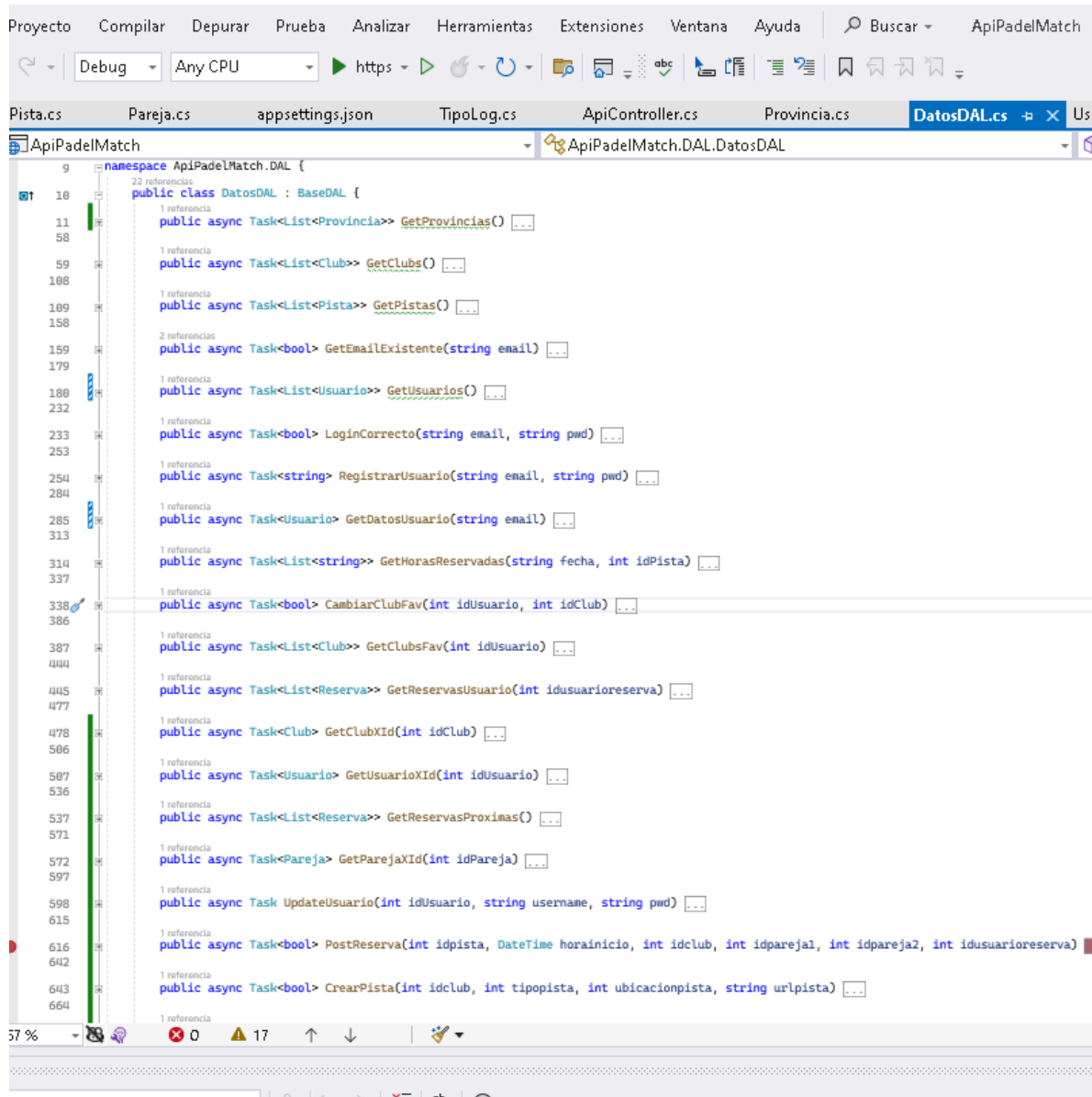
[Route("/api/")]
[ApiController]
public class BaseRoute : ControllerBase {
}
```

Las routes:

```
[Route("/api/Pistas")]
[ApiController]
public class PistasRoutes: BaseRoute
{

    [Produces("application/json")]
    [ProducesResponseType(StatusCodes.Status200OK)]
    [ProducesResponseType(StatusCodes.Status400BadRequest)]
    [ProducesResponseType(StatusCodes.Status500InternalServerError)]
    [HttpPost("CrearPista")]
    public async Task<IActionResult> CrearPista ([FromHeader]int tipopista,
    [FromHeader] int idclub....)
    {
```

Ejemplo de los métodos dentro del Data Access Layer (DAL)



Y aquí un solo método para que podáis ver la parte de (DAL):

```
public async Task<bool> LoginCorrecto(string email, string pwd) {
    try {
        var existe = false;
        using (var connection = new
NpgsqlConnection(Config.GetConnectionString("tfqBD"))) {
            var query = "SELECT COUNT(*) FROM app.usuario WHERE email =
@Email AND password = @Pwd";
            await connection.OpenAsync();
```

```

        using (var command = new NpgsqlCommand(query, connection)) {
            command.Parameters.AddWithValue("@Email", email);
            command.Parameters.AddWithValue("@Pwd", pwd);
            var count = (long)await command.ExecuteScalarAsync(); // Usa
            ExecuteScalarAsync para obtener el valor de COUNT(*)
            existe = count > 0;
        }
    }
    return existe;

} catch (Exception ex) {
    BaseDAL.InsertarLogs(TipoLog.ERROR, $"error al buscar el mail: error =
{ex.Message}", "GetPasswordCorrecta", 0);
    return false;
}
}

```

```

public static IConfiguration Conf {
    get {
        return _conf ?? (_conf = new ConfigurationBuilder()
            .AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)
            .Build());
    }
}

```

En el program.cs tendremos que añadir

```

// Register appsettings

builder.Configuration.AddJsonFile("appsettings.json", optional: false,
reloadOnChange: true);

```

En el appsettings:
 declarar los posibles connection string y así podremos recuperar la conexión sin
 tener que hacer Hardcoding.

```

"ConnectionStrings": {

    "tfgBD": "Server=127.0.0.1;User
ID=postgres;Password=dreuan12;Database=padeltfg;Port=5432;CommandTimeout=
300;IncludeErrorDetail=true;"

},

```

También está la carpeta Modelos la cual contiene todos los modelos de la api PadelMatch: usuario, reserva, pista, provincia, club y pareja.

Ejemplo de uno de ellos:

```
public class Pista {  
    public int idpista { get; set; }  
    public int idclub { get; set; }  
    public int tipopista { get; set; }  
    public int ubicacionpista { get; set; }  
    public string urlimagenes { get; set; }  
}
```

El único Enum que no está declarado en ninguna base de datos es el de TipoLog este se utiliza para clasificar el tipo de log sobre la tabla Logs.

```
public enum TipoLog {  
    ERROR,  
    WARN,  
    INFO,  
    DEBUG  
}
```

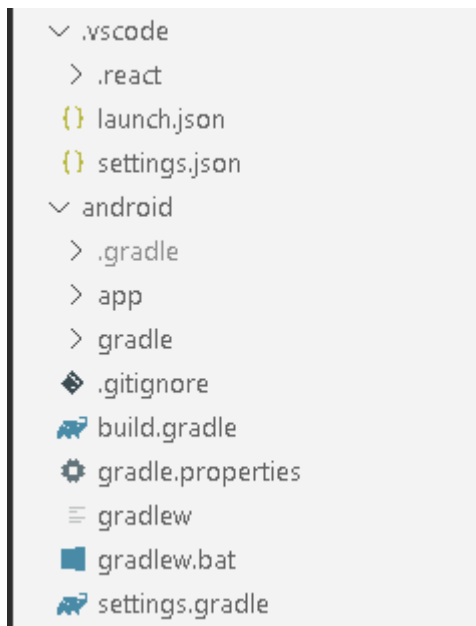
4.5. APLICACIÓN MÓVIL

La aplicación móvil PadelMatch está diseñada para facilitar la reserva de pistas de pádel, permitiendo a los usuarios gestionar sus reservas, ver información sobre clubs y organizar partidos.

Desarrollada en React Native y lanzada mediante Expo, la aplicación se estructura en diversas clases, modelos y Endpoints para proporcionar una experiencia de usuario fluida y eficiente.

4.5.1. CARPETAS

- **Assets:**
Contiene la carpeta images donde se guarda el gif, vibora.mp4, imágenes logo etc.
Las cuales se utilizan luego en diferentes pantallas de la App.
- **Navigation:**
Esta carpeta contiene la carpeta screens a la cual se han añadido las pantallas de app, LoginScreen, HomeScreen y todos los demás que podrás encontrar en el repositorio de Github.
- **Componentes:**
Aquí se guarda el AuthContext, Header, ImageViewer, PistaComponent, SearchResult y styles (al final opte por dejar ahí algún style general para el principio pero con el desarrollo del proyecto me atrapaba mucho con todos los styles juntos y preferí dividir la gran mayoría en individualmente por cada una de las screens).
- **Vscode:**
Tiene toda la parte de android, No se ha podido integrar IOS aún por falta de tiempo pero se eligió este lenguaje, por sus ventajas a la hora de querer tener la aplicación también en IOS.



La principal ventaja de React Native es que permite a los desarrolladores crear aplicaciones móviles utilizando un único conjunto de código base en lugar de tener que escribir códigos separados para iOS y Android (utilizando componentes de interfaz de usuario que se traducen a elementos nativos de cada plataforma).

4.5.2. ENDPOINTS

Las rutas utilizadas para programar la App han sido en local. Se ha utilizado el debug para comprobar que llegaban bien los datos a la Api y si algo de parte del cliente no tenía sentido ya sabía de donde podía venir el error.

Ejemplo de ruta en desarrollo:

```
const urlreservas =  
'http://192.168.1.139:5098/Api/GetReservasUsuario';
```

Ejemplo de ruta en producción:

```
const urlreservas=  
http://api.padelmatch.elitepadelformation.com/Api/GetReservasUsuario';
```

5. TECNOLOGÍAS UTILIZADAS

Frontend:

- Framework: React Native
- Lenguaje: JavaScript
- Entorno de Desarrollo: Expo
- Editor de Código: Visual Studio Code
- Emulador: Android Studio

Backend:

- Framework: ASP.NET Core
- Lenguaje: C#
- Entorno de Desarrollo: Visual Studio
- Gestión de Dependencias: Paquetes NuGet

Almacenamiento y gestión de información:

- Base de datos: [PostgreSQL](#)
- Lenguaje: [SQL](#)
- Entorno desarrollo: [MySQL Workbench](#) + [PGAdmin4](#) + [DataGrip 2023](#)

Documentación externa:

- Marco de gestión y creación de documentación técnica: [Mkdocs](#)
- Lenguaje: [Markdown](#), [aspx](#), [html](#), [css](#)
- Theme utilizado: [Material](#)

6. RECURSOS

6.1. Software

- Se ha utilizado Visual Studio para el desarrollo de la API REST en ASP.NET Core v7 donde se han declarado todos los creado todo el código necesario para su correcto funcionamiento.
Paquetes Nuget instalados:
 1. [Accelergreat PostgreSQL \(Npsql\) Entity Framework](#).
 2. [Newtonsoft.Json](#)
 3. [Swashbuckle.AspNetCore](#)
 4. [System.Data.SqlClient](#)
- Se ha utilizado Android Studio para poder tener un emulador móvil Pixel 6 API 24 con el que se han realizado todas las pruebas.
- Visual Studio Code desarrollo de toda la aplicación de PadelMatch utilizando expo para poder hacer más viable el desarrollo de la app.
- También se ha utilizado Visual Studio Code para el desarrollo de la Documentación API para desarrolladores externos, la cual está hecha en Markdown con MkDocs y el tema Material.

6.2. Hardware

- Para poder desarrollar PadelMatch se ha utilizado un ordenador sobremesa.
- Necesidad de un dispositivo móvil para utilizar la aplicación.
- Un servidor donde poder publicar todos los servicios.

6.3. Sistemas Operativos

- Ordenador sobremesa para el desarrollo: [Windows 10 pro](#)
- Servidor: [Windows Server 2021](#)

6.4. Equipo

- En el desarrollo de PadelMatch solo he estado yo implicado, realizando todo el desarrollo tanto de back como de front.
- A futuro sería mucho más sencillo tener programadores con diferentes tareas para poder desarrollar PadelMatch de una forma más rápida y eficaz. Aplicando Agile, reuniones diarias y poniendo puntos de vista en común.
- También a futuro necesitaremos un equipo de marketing y soporte para poder resolver todos los problemas rápidamente y afrontar todo tipo de situaciones.

7. PUBLICACIÓN

Se ha redactado un readme en el repositorio de Github donde se explica paso a paso el cómo poder tanto volver a crear este proyecto como publicarlo para tener acceso externo.

8. CONCLUSIONES

- Este proyecto se haría más asequible con un par de personas, así se podría repartir el trabajo entre los integrantes y poder avanzar de una forma mucho más rápida.
- Lo bueno es que he aprendido a utilizar React Native ya que nunca había trabajado con él, también aprender JavaScript que lo tenía pendiente, el

desarrollo con expo al principio me fue incluso complicada solo la parte de poder crear el proyecto y entender como lanzarlo.
Pero ahora ya lo tengo como un aprendizaje más.

- Tenía bastantes dudas en si abrir puertos en mi casa llamando a la operadora pero empecé un proyecto personal y ahora tengo un Windows server 2021 en el cual tengo publicada una web en el dominio(elitepadelformation.com) comprado en GoDaddy y este servidor lo he utilizado para publicar en el IIS las apis que he desarrollado.
- También mencionar que he mejorado a la hora de administrar el DNS y configurar las rutas para poder acceder a los Endpoints de las apis.

8.1. GRADO DE CONSECUCCIÓN DE OBJETIVOS

Analizando el proyecto, **el aprendizaje** en JavaScript, React Native, Expo, Aspx, Markdown y los conceptos aprendidos a la hora de asignar los dominios y configurar el IIS, pongo en valor el desarrollo del proyecto, ya que he aprendido muchas habilidades nuevas.

Esta parte para mi tiene un **90%** de efectividad.

En el **apartado del desarrollo** de solo la **App PadelMatch** en mi opinión, creo que está bastante bien con un diseño agradable y lineal.

Lo único, es que algunas funcionalidades de la App no las he podido completar y solo asignaría un **70%** de requisitos conseguidos accesibles desde la UI.

Las **apis y servidores** también me han hecho sentir muy realizado y asignaría fácil un **80%**.

Mi grado de consecución de objetivos sumando y dividiendo los porcentajes es de un **80%**.

Este proyecto aún tiene mucho por delante y con un equipo se conseguirán los objetivos de cooperar con clubs y facturar de las reservas que se hagan a través de la App.

8.2. PROBLEMAS ENCONTRADOS

A medida que desarrollaba PadelMatch encontraba diferentes errores con los cuales pivotaba en la orientación del proyecto, tanto programando en JavaScript como en la base de datos y también a la hora de poner en marcha el marco de trabajo con expo.

Navegación en la app

El primer problema que aún no he conseguido resolver es a la hora de cerrar sesión o intentar volver solo a la pestaña de Login creo que hay algún concepto de JavaScript que desconozco o no lo implemento todo lo bien que se debería hacer.

Este fallo aparece cuando el usuario inicia sesión e intenta volver hacia atrás con los botones del dispositivo la pantalla se queda en blanco devolviendo:

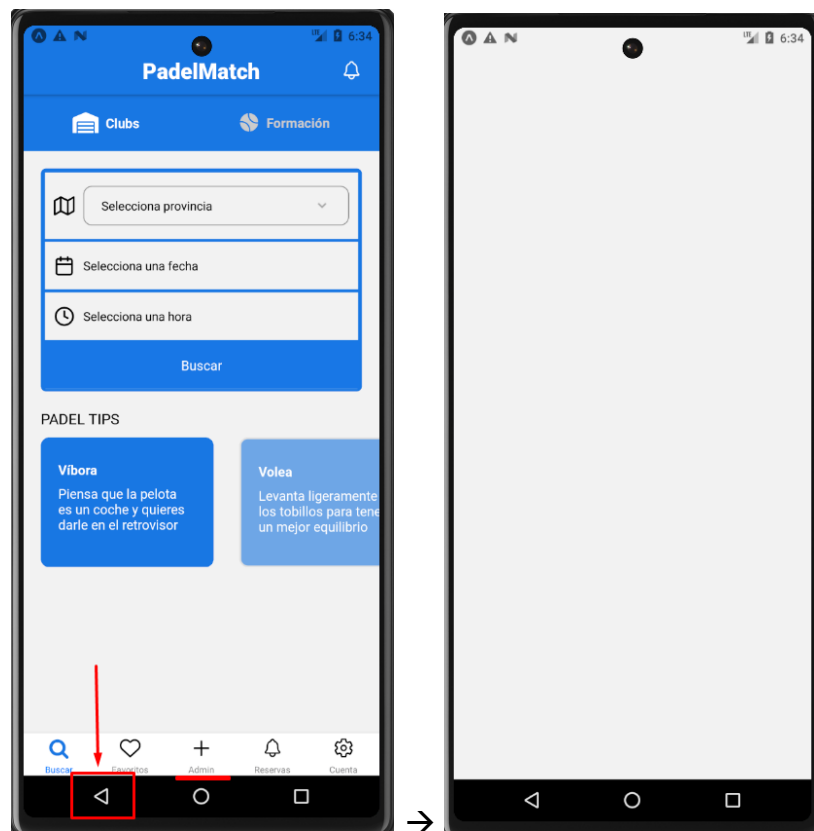
Require cycles are allowed, but can result in uninitialized values. Consider refactoring to remove the need for a cycle.

WARN Require cycle: navigation\screens\LoginScreen.js -> StackNavigator.js -> navigation\screens\LoginScreen.js

Supuestamente está dentro de todo el NavigationContainer.

Primera de las correcciones más graves que resolveré en un futuro.

```
<AuthProvider>
  <NavigationContainer>
    <Stack.Navigator>
      <Stack.Screen name="Login" component={LoginScreen} options={{ headerShown: false }} />
      <Stack.Screen name="Main" component={BottomTabs} options={{ headerShown: false }} />
      <Stack.Screen name="Search" component={SearchScreen} options={{ title: `Clubs` }} />
      <Stack.Screen name="PistasClub" component={PistasClubScreen} />
      <Stack.Screen name="HorasPista" component={HorasDisponiblesPista} />
      <Stack.Screen name="PagarReserva" component={PagarReserva} />
      <Stack.Screen name="Favoritos" component={FavoritosScreen} />
      <Stack.Screen name="Profile" component={ProfileScreen} />
      <Stack.Screen name="Formación" component={FormacionScreen} />
    </Stack.Navigator>
  </NavigationContainer>
</AuthProvider>
```



Parámetros entre pantallas

Cuando comencé a poder pasar parámetros entre screens vi que necesitaba guardar demasiados parámetros a parte de los que eran necesarios para la lógica base, cuando intente pasar hasta datos del usuario entre pantallas vi que no era una buena práctica y conseguí encontrar una forma de guardármelo con context.

Acabe utilizando el AuthContext.js para poder hacer el Inicio de sesión(email), logout(), y recuperación del usuario me ha sido muy útil y sencillo implementarlo y ahora es posible recuperar desde cualquier pantalla la información del usuario.

Debug en expo

No he conseguido poder debugear la parte de la App con puntos de interrupción, busqué y probé de implementar React Native Doctor pero sin resultado alguno.

Entonces como ya estaba en una etapa muy avanzada seguí resolviendo los problemas que me daba la App implementando console.logs justo encima de las líneas que creía que no funcionaban correctamente.

En un futuro, esto tendrá que ser solucionado para un desarrollo fácil, rápido y conveniente.

Montar expo y dependencias

Como nunca había utilizado React Native ni JavaScript ni Expo los primeros meses, tuve que volver a empezar el proyecto en repetidas ocasiones, hasta el punto de tener tantas carpetas para el proyecto que ya era imposible hacer que funcionará y tener que crear unas 10 veces fácilmente el sistema de la instalación desde cero con Expo.

React-Native-Community

He tenido bastantes problemas con diferentes tipos de integraciones, con ciertos paquetes de instalación o modales desarrollados por la comunidad de React Native y también muchas soluciones, pero se ha perdido un tiempo primordial el cual no se puede recuperar por no comprobar reseñas finales de los paquetes por si aún estaban en desarrollo etc.

Backup BD

Intenté realizar un Backup de la base de datos una vez terminada y PgAdmin4 siempre me devolvía error(Failed (exit code: 3221225781)), las versiones antiguas tienen muchos fallos, instalé la última y todo perfecto, Backup subida en el repositorio por si se quiere hacer un restore.

8.3. MEJORAS

En un futuro con un buen equipo para llevar a cabo el proyecto pienso llevar un progreso controlado con SCRUM o Agile el desarrollo de PadelMatch.

- Habrá mejoras de funcionalidad y sin errores visuales.
- Implementar el guardado en base de datos de las imágenes que se suban a la App tanto de usuario como pistas como clubs, hacer un CDN donde la ruta sea accesible desde cualquier lugar y, la ruta concluya con la extensión de la imagen.
- Muy probable que se realice una web para no limitarnos a solo acceso a reservas si tienes la App descargada y también lanzar App en IOS.
- Crear el proceso de reserva más sencillo y rápido.
- Agregar en HomeScreen un botón de asignar Hoy o Mañana que rellene el DatePicker.

9. ANEXOS Y DOCUMENTACIÓN COMPLEMENTARIOS

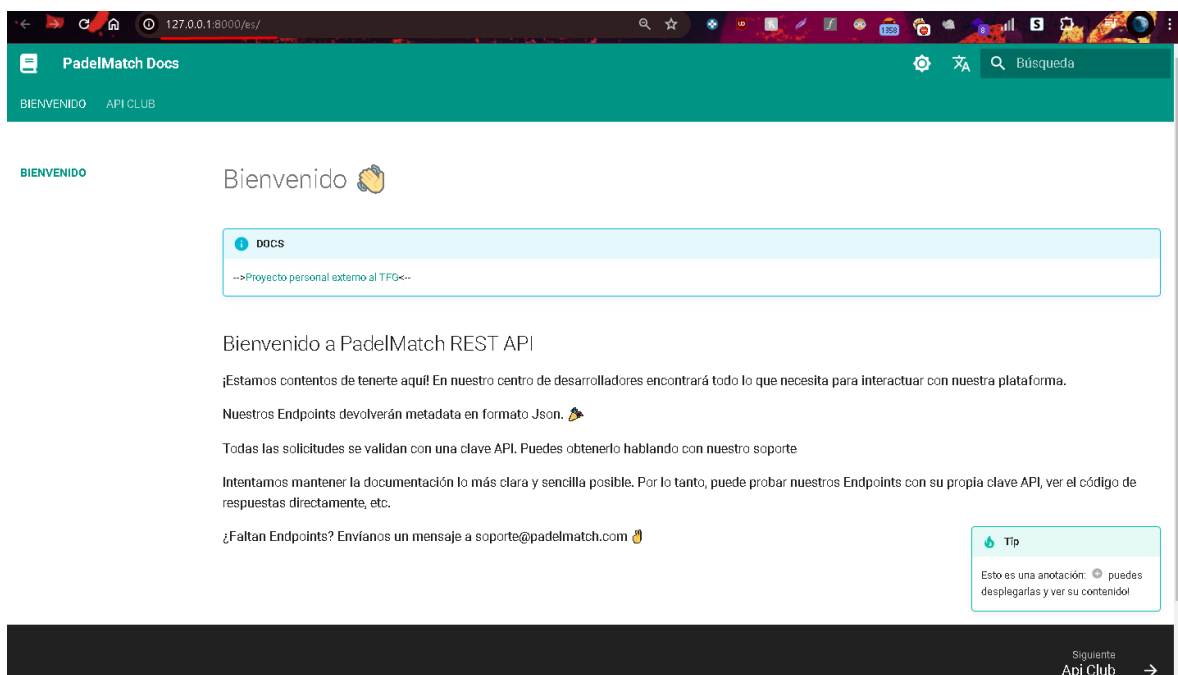
He preparado una Documentación Api externa con login y acceso restringido controlado por tokens en las cookies del navegador para solo usuarios que se les haya concedido acceso previo.

Username: Tribunal

Password: tribunalvereda10=D

Documentación Api PadelMatch

<http://docs.padelmatch.elitepadelformation.com/login.aspx>



Para publicar la documentación y poder controlar el acceso a las rutas con un usuario que haya hecho login se ha creado un .exe con c# el cual sobrescribe el código de mkdocs al hacer el build.

Este .exe (ExeDocumentación) itera sobre la ruta que hace referencia al site.

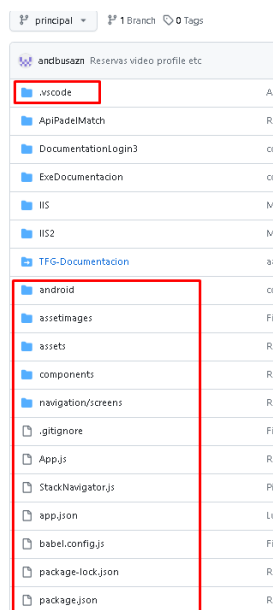
```
string carpetaSite =  
@"C:\github\23_24_DAM_ANDREU_BUSTOS_AZNAR_PISTAS_PADEL\TFG-  
Documentacion\site";
```

Injecta código en todos los archivos .html creados después del build los cuales hacen ejecutan Endpoints de la Api Documentacion Login.

Actualmente a día de hacer la memoria aún esta publicada la documentación en el servidor pero en un par de días estará operativa y la ruta será pública, no local.

10. REFERENCIAS BIBLIOGRAFÍA

1. Documentación MKDOCS IIS: https://github.com/ieslavereda-projects/23_24_DAM_ANDREU_BUSTOS_AZNAR_PISTAS_PADEL/tree/principal/IIS
2. Api Login Documentación: https://github.com/ieslavereda-projects/23_24_DAM_ANDREU_BUSTOS_AZNAR_PISTAS_PADEL/tree/principal/IIS2
3. App PadelMatch:



https://github.com/ieslavereda-projects/23_24_DAM_ANDREU_BUSTOS_AZNAR_PISTAS_PADEL/tree/principal

4. Que es React Native: 3androides.com
5. Dropdown Button: <https://www.npmjs.com/package/react-native-dropdown-select-list>
6. Lottie Animación prelogin: <https://www.youtube.com/watch?v=mZXCOdIFg6Q>

7. IIS error 404.17: <https://stackoverflow.com/questions/7083533/http-error-404-17-not-found>
8. IIS error: <https://learn.microsoft.com/es-es/troubleshoot/developer/webapps/iis/health-diagnostic-performance/error-message-you-visit-web-site>
9. IIS error: <https://community.lansweeper.com/t5/installation/install-iis-internet-information-services/ta-p/64422>
10. Date picker: <https://www.npmjs.com/package/react-native-modal-datetime-picker?activeTab=readme>
11. También se puso en desarrollo un scheduler el cuál acabé descartando: <https://www.npmjs.com/package/react-native-calendar-timetable>
12. Colores obtenidos de: <https://color.adobe.com/es/create/color-wheel>
13. Concepto base de reservas y navegación con : <https://galaxies.dev/> y <https://www.youtube.com/watch?v=iWzUZiVoiR0>
14. Llamada a endpoints con axios: <https://axios-http.com/es/docs/intro>

11. ÍNDICE DE IMÁGENES

- Logotipo PadelMatch creado en canva basado en el idpegatina: EAEgIPLGZek
<https://www.canva.com/>
- Animación logotipo en la app con canva
<https://www.canva.com/>
- Imagen IES LA VEREDA
<https://portal.edu.gva.es/lavereda/es/centro-es/>
- Diseño presentación diapositivas: Basado en Plantilla Olmos Carlos
<https://www.canva.com/>
- Video interno app: Grabado por mí y con autorización de Javier Navarro.
<https://elitepadelformation.com/>
- Imágenes públicas en internet para rellenar las pistas y usuarios de PadelMatch:
 - <https://padelmagic.es/wp-content/uploads/2023/06/padellounge.jpg>

- https://www.elneverazo.com/wp-content/uploads/2023/03/334161222_754282159565211_7642507603463632578_n.jpg
- <https://www.padelfip.com/wp-content/uploads/2023/02/Javier-Navarro-Perez.jpg>
<https://padelstar.es/wp-content/uploads/2020/03/Club-de-Padel-Madrid-Ciudad-Raqueta.jpg>
- <https://padelmagic.es/wp-content/uploads/2023/06/padellounge.jpg>
<https://i.ytimg.com/vi/d8ySIJqM830/hqdefault.jpg>
- <https://padelmagic.es/wp-content/uploads/2023/06/padellounge.jpg>
- https://sportcityvalencia.com/wp-content/uploads/2020/06/IMG_8919web.jpg
- https://assets-global.website-files.com/639778ab6f2e51ed2139df9a/63de42f33a1d59699804906e_20210527_084626-2.jpg
- <https://cpimg.tistatic.com/06860496/b/4/Tennis-Padel-Courts.jpg>
https://unbuendiaenmadrid.com/wp-content/uploads/2022/03/Z62_9970-2.jpg
- <https://sc04.alicdn.com/kf/H62d022f118aa47749ca5b24f0ac3b7225.jpg>
- <https://sportsclubtour.com/wp-content/uploads/2021/08/Vita10-Club-Padel-1.jpg>
- <https://zynkdesign.com/wp-content/uploads/2023/12/Screenshot-2023-12-20-at-11.27.13-1024x723.png>
- https://worldpadeltour.com/assets/themes/worldpadeltour.com/img/posts/1690449903_Publicidad%20Mondo_Cervezas%20Victorias%20Malaga%20Open%202023-Dieciseisavos_MPR01006%20copia.thigh.jpg
- <https://www.pistas-padel.es/wp-content/uploads/2021/11/Pista-de-padel-indoor.jpg>
- https://padelquatro.com/wp-content/uploads/2021/12/Pista_central_b.jpeg
- <https://www.empadelados.com/wp-content/uploads/padel-indoor-696x385.jpg>
- Todas las imágenes restantes son capturas de pantalla hechas por mí a la hora de redactar la memoria.