

RELEVAMIENTO DE API

CanchaYa

Preparado Para

Corti Matias

08/04/2025

Versión 1.0

Preparado Por

Antar, Mauro

Benitez, Ignacio

Cayo, Ariel

Dieguez, Matias

Mendez, Alexis

Perez, Thiago

UTN FRLP

Pantallas del frontend

HomePage

Endpoints

1. **Endpoint 1: Obtener lista de deportes**
 - a. **Método HTTP:** GET
 - b. **URL:** /api/deportes
 - c. **Regla de Negocio:** Devuelve una lista de todos los deportes disponibles en la plataforma para construir el componente **FiltroDeporte**
2. **Endpoint 2: Obtener canchas populares**
 - a. **Método HTTP:** GET
 - b. **URL:** /api/canchas/populares
 - c. **Regla de Negocio:** Devuelve una lista con mejor puntaje o más reservadas. El backen puede terminar la lógica exacta, por ejemplo, devolver la cancha con la calificación mas alta de cada deporte.

Cuales necesitan autenticación

No se necesita ninguna autenticación ya que la información de la página principal es pública y debe ser accesible para todos los visitantes, hayan iniciado sesión o no.

Interfaces TypeScript para request/response

1. Endpoint 1:

```
// Response para el Endpoint 1: api/deportes
interface DeporteResponse {
  id: number;
  nombre: string;
  // Interfaz para una cancha en una lista
  interface CanchaListItem {
    id: number;
    nroCancha: number;
    puntaje: number;
    precioDesde: number;
    imageUrl: string;
    complejo: {
```

```

        id: number;
        nombre: string;
        ubicacion: string;
    };
    turnusDisponibles: { hora: string }[];
}

```

2. Endpoint 2:

```

// Response para el Endpoint 2: /api/canchas/populares
// Devuelve un array de CanchaListItem
type PopularesResponse = canchaListItem[];

```

Perfil Usuario

Endpoints

3. Endpoint 1: Mostrar Datos Del Usuario

- a. Método HTTP: GET
- b. URL: /api/usuario
- c. Regla de Negocio: Permite al usuario ver sus datos personales (teléfono, mail, domicilio, nombre, apellido y un apodo).

4. Endpoint 2: Modificar Datos Personales

- a. Método HTTP: PUT
- b. URL: /api/usuario
- c. Regla De Negocio: El usuario podrá modificar sus datos personales en el momento que desee.

5. Endpoint 3: Calificar Reservar

- a. Método HTTP: POST
- b. URL: /api/reserva/id:/calificar
- c. Regla De Negocio: El usuario podrá calificar y dejar su opinión en aquellas reservas cuyo estado sean finalizadas

Cuales necesitan autenticación

Para empezar todos los endpoints necesitan que el usuario haya iniciado sesión

Endpoint 3: Para poder calificar la reserva el estado de las misma debe ser finalizada

Interfaces TypeScript para request/response

1. Endpoint 1:

```
// Response
interface UsuarioResponse {
  id: number;
  nombre: string;
  apellido: string;
  apodo: string;
  mail: string;
  telefono: number;
  domicilio: string;
  reservas: reservaResumen[];
}

// Response del historial de reservas del usuario
interface reservaResumen {
  id: number; // id de la reserva
  estado: 'pendiente' | 'cancelada' | 'confirmada' | 'finalizada'
  fecha: datetime;
  hora: datetime;

  complejnombre: string;
  canchanombre: string;
}
```

2. Endpoint 2:

```
// Request
interface ModificarUsuarioRequest {
  nombre: string;
  apellido: string;
  apodo: string;
  mail: string;
  telefono: number;
  domicilio: string;
}

// Response
interface UsuarioActualizarResponse {
  success: boolean;
  usuario: UsuarioResponse;
}
```

3. Endpoint 3:

```
// Request
interface CalificarReservaRequest {
```

```

    puntaje: number | null; // // 0.5 a 5 o 0 si eligió "?"
    comentario?: string;
}
// Response
interface CalificarReservaResponse {
    id: number;
    reservaId: number;
    canchaId: number;
    puntaje: number;
    comentario: string;
    fecha: datetime;
}

```

Perfil Complejo

Endpoints

6. **Endpoint 1: Mostrar datos del complejo**
 - a. Método HTTP: GET
 - b. URL: /api/miComplejo/id:
7. **Endpoint 2: Modificar datos del complejo**
 - a. Método HTTP: PUT
 - b. URL: /api/miComplejo/id/
8. **Endpoint 3: Crear una cancha del complejo**
 - a. Método HTTP: POST
 - b. URL: /api/miComplejo/id:/canchas/
9. **Endpoint 4: Eliminar una cancha del complejo**
 - a. Método HTTP: DELETE
 - b. URL: /api/miComplejo/id:/canchas/id:

Cuales necesitan autenticación

Todos los endpoints requieren que el usuario tenga perfil de dueño y sea dueño del complejo.

Interfaces TypeScript para request/response

```
// Response para un GET de Complejo
interface ComplejoRes {
  id : number;
  altura : number;
  calle : string;
  cuit : string;
  descripcion : string;
  nombre : string;
  porcentajeReembolso : number;
  puntaje : number;
  localidad : Localidad;
  canchas : Cancha[] | null;
}

// Request para hacer PUT de un Complejo
interface ComplejoReq {
  id : number;
  nombre : string;
  descripcion : string;
  porcentajeReembolso: number;
  localidad : Localidad;
}

// Request para hacer POST de una cancha
interface CanchaReq {
  nroCancha : number;
  descripcion : string;
  deporte : Deporte;
}
```

Administrador

Endpoints

10. Endpoint 1:

- a. Método HTTP: GET
- b. URL: /api/admin

- c. Regla de Negocio: Devuelve el entorno del perfil administrador, en donde se pueden visualizar complejos ya aprobados para incluir en la página y las nuevas solicitudes de nuevos complejos.

Cuales necesitan autenticación

El usuario debe haber iniciado sesión con perfil de administrador para poder visualizar y manejar las solicitudes.

Interfaces TypeScript para request/response

```
// response de solicitudes de nuevos complejos
interface solicitud {
  imageUrl: string
  calle: string;
  altura: int;
  porcReembolso: float;
  horario: time;
  cuit: int;
  descripcion: string;
}

// response de complejos aprobados
type ComplejosAprobados = complejosAprobadosLista[]
```

Reserva Cancha

Endpoints

11. Endpoint 1: Obtener cancha

- a. Método HTTP: GET
- b. URL /api/canchas/id:
- c. Regla de Negocio: Devuelve los datos públicos de una cancha: nombre, dirección, imágenes, puntaje, descripción, turnos disponibles, reseñas.

12. Endpoint 2: Hacer reservacion de turno

- a. Método HTTP: POST
- b. URL: /api/canchas/id:/reserva

- c. Regla De Negocios: Permite hacer una reserva

13. Endpoint 3: Dejar reseña

- a. Método HTTP: POST
- b. URL: /api/canchas/id:/resena
- c. Regla De Negocios: Permite dejar una reseña con la opiniones y un puntaje para que otros usuarios puedan verlo

Cuales necesitan autenticación

Endpoint 1: No se necesita ninguna autenticación ya que la información de la página principal es pública y debe ser accesible para todos los visitantes, hayan iniciado sesión o no.

Endpoint 2: El usuario debe haber iniciado sesión en la página para poder realizar esta operación.

Endpoint 3: El usuario debe haber iniciado sesión en la página y debe tener una reserva activa para poder realizar esta operación.

Interfaces TypeScript para request/response

1. Endpoint 1:

```
interface CanchaDetalleResponse {  
  id: number;  
  nroCancha: number;  
  puntaje: number;  
  precioDesde: number;  
  imageUrl: string;  
  // No se como se pone la ubicación de la cancha  
  ubicacion: string;  
  Descripcion: string;  
  
  complejo: {  
    id: number;  
    nombre: string;  
    ubicacion: string;  
  };  
  turnosDisponibles: TurnosDisponibles[];  
}
```

2. Endpoint 2:


```
// Request
interface CrearReservaRequest {
    turnoId: number;
}

// Response
interface ReservaResponse {
    id: number;
    usuarioid: number;
    canchaid: number;
    turno: {
        id: number;
        fecha: datetime;
        hora: datetime;
    };
    estado: 'confirmada' | 'cancelada';
}
```

3. Endpoint 3:

```
// Request
interface CrearReservaRequest {
    turnoId: number;
    puntaje: number;
}

// Response
interface ReservaResponse {
    id: number;
    usuarioid: number;
    canchaid: number;
    puntaje: number;
    comentario: string;
    fecha: datetime;
}
```

Resultado filtrado

Endpoints

14. Endpoint 1: Buscar canchas por criterios

- a. **Método HTTP:** GET
- b. **URL:** /api/canchas/buscar?ciudad=
/api/canchas/buscar?deporte=
/api/canchas/buscar?fecha=
Ej: /api/canchas/buscar?ciudad=LaPlata&deporte=Tenis&fecha=2025-07-30
/api/canchas/buscar?deporte=Tenis&page=2&limit=12
- c. **Regla de Negocio:** Filtrar todas las canchas de la base de datos según los parámetros opcionales de consulta(ciudad, deporte, fecha, hora). El backend debe ser capaz de manejar combinaciones de los filtros. Debe incluir la paginación (page, limit) para manejar grandes cantidades de resultados de manera eficiente.

Cuales necesitan autenticación

No necesita autenticación ya que la funcionalidad de búsqueda es pública. Un usuario no necesita iniciar sesión para ver qué canchas están disponibles.

Interfaces TypeScript para request/response

```
// Respuesta para el Endpoint 1: /api/canchas/buscar
interface BusquedaResponse {
  resultados: CanchaListItem[];
  totalResultados: number;
  totalPaginas: number;
  paginaActual: number;
}
```