

ALEJANDRO SAINZ SAINZ PADEL-NODE

```
Modelo Pista cargado correctamente
^C
alex@PikAlex:~/ProyectosProgramacion/Github/Daw2-26-26/club-padel-api-practica main$ alex@PikAlex:~/ProyectosProgramacion/Github/Daw2-26-26/club-padel-api-practica main$ npx ts-node src, [dotenv@17.2.3] injecting env (5) from .env -- tip: ⚡ suppress all logs with { quiet: true }
Servidor escuchando en http://localhost:3000
Conexión a la base de datos establecida correctamente
Modelo Pista cargado correctamente
```

Esta es la aplicación cargada desde la terminal. Se me olvido incluir un mensaje desde la carga del Modelo Reserva. Pero se ha cargado, lo se porque ya no da fallos.

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [club_padel_db]> show tables;
+-----+
| Tables_in_club_padel_db |
+-----+
| pistas
| reservas
+-----+
2 rows in set (0,000 sec)

MariaDB [club_padel_db]>
```

Captura desde la terminal de la base de datos club_padel_db y las tablas en esa bd.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help'. Below that is a toolbar with icons for back, forward, search, and settings, along with an 'Upgrade' button. The main area is titled 'Workspaces' with a dropdown menu, and there are buttons for 'Overview', 'pistasReservas' (which is highlighted with a red underline), and 'No environment'. On the left, there are several small icons. The central workspace titled 'pistasReservas' contains a table of variables:

Variable	Value ↑
raiz	/api
pista	/pistas
res	/reservas
baseUrl	http://localhost:3000
Add variable	

Después de crear el workspace en postman defino las variables del enviroment.

```
pistas.routes.ts  reservas.routes.ts
alex@PikAlex:~/ProyectosProgramacion/Github/Daw2-26-26/club-padel-api-practica/src/routes
import {Router} from 'express';
import {listarPistas, obtenerPistaId, crearPista, modificarNombrePista, eliminarPista} from '../controllers/pistasController';

const router: Router = Router();

router.get('/pistas', listarPistas);
router.get('/pistas/:id', obtenerPistaId);
router.post('/pistas',crearPista);
router.put('/pistas/:id',modificarNombrePista);
router.delete('/pistas/:id', eliminarPista);

export = router;
```

Estos son mis endpoints para las pistas, sobre los que voy a hacer las pruebas, o por lo menos las que se piden para la práctica.

The screenshot shows the Postman application interface. At the top, there are navigation buttons for Workspaces and More, along with search, filter, settings, and upgrade options. The workspace is set to 'pistasReservas'. Below the header, the URL is 'HTTP Padel Club / GetPistas' with a 'Save' button and a 'Share' button. The method is set to 'POST' with parameters: {{baseUrl}}, {{raiz}}, and {{pista}}. A large blue 'Send' button is visible. Below the method section, tabs for Docs, Params, Auth, Headers (8), Body (selected), Scripts, Settings, and Cookies are present. The Body tab has two dropdowns: 'raw' (selected) and 'JSON'. The JSON content is:

```
1 {  
2   "nombre": "Pista Central",  
3   "tipo": "INDOOR",  
4   "precioHora": 11.45  
5 }
```

At the bottom, the response is shown with a status of '201 Created', a duration of '64 ms', and a size of '308 B'. The response body is identical to the request body:

```
1 {  
2   "id": 3,  
3   "nombre": "Pista Central",  
4   "tipo": "INDOOR",  
5   "precioHora": 11.45  
6 }
```

Primero comienzo haciendo un post. Correcto. Como ya había hecho alguna prueba primero, por esos tiene la id 3.

The screenshot shows the Postman application interface. At the top, there is a navigation bar with 'File', 'Edit', 'View', and 'Help' options. Below the navigation bar, there are buttons for 'Workspaces' and 'More'. On the right side of the top bar, there are icons for search, filter, settings, notifications, and a profile picture, followed by a red 'Upgrade' button.

The main workspace is titled 'pistasReservas'. The left sidebar contains a tree view with nodes like 'pistas' and 'Padel'. A 'POST GetF' item is also visible. The main content area shows a request configuration for an 'HTTP' request to 'Padel Club / Get all'. The method is set to 'GET', and the URL template is {{baseUrl}}/{{raiz}}/{{pista}}. There are buttons for 'Save', 'Share', and 'Send'. Below the request configuration, there are tabs for 'Docs', 'Params' (which is selected), 'Auth', 'Headers (8)', 'Body', 'Scripts', 'Settings', and 'Cookies'. The 'Params' tab shows a table for 'Query Params' with columns for 'Key', 'Value', 'Descri...', and 'Bulk Edit'. The table currently has one row with 'Key' and 'Value' columns.

At the bottom of the interface, there is a 'Body' section showing a JSON response. The response is a single object with an array of objects. One object in the array is displayed in detail:

```
1 [  
2 {  
3   "id": 2,  
4   "nombre": "Pista Este",  
5   "tipo": "INDOOR",  
6   "precioHora": "0.00"  
7 },  
8 {  
9   "id": 3
```

Después hacemos un get all. Es la misma url pero con método get.

The screenshot shows a REST client interface with the following details:

- Header Bar:** File, Edit, View, Help.
- Toolbar:** Back, Forward, Workspaces, More, Search, Refresh, Settings, Bell, Upgrade button.
- Request URL:** {{baseUrl}}/{{raiz}}/{{pista}}
- Method:** POST
- Body:** JSON (shown in raw and JSON dropdowns)

```
1 {  
2   "nombre": "Pista Central",  
3   "tipo": "DESCUBIERTA",  
4   "precioHora": 11.45  
5 }
```
- Response:** 409 Conflict (11 ms, 287 B)

```
{ } JSON ▾
```

Body: { "error": "Ya existe una pista con ese nombre" }
- Bottom Navigation:** Cloud View, Console, Terminal, Runner, Vault, etc.

Intento crear una pista con un tipo que no es el indicado. Hice validación en el controller y por eso nos devuelve el estatus 409 y el mensaje de error correspondiente. Aunque bueno, he intentado insertarla con un nombre que ya existe y por eso me da el error del nombre. Vamos a ver si lo hago bien ahora.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Help', and an 'Upgrade' button. Below the navigation is a toolbar with icons for search, refresh, settings, and notifications.

The main workspace shows a collection named 'pistas' with a sub-collection 'Padel'. A specific POST request is selected, labeled 'GetPistas'. The request details show it's a POST method to 'Padel Club / GetPistas'. The body is set to 'raw' and 'JSON'. The JSON payload is:

```
1 {
2     "nombre": "Pista Maginifica",
3     "tipo": "DESCUBIERTA",
4     "precioHora": 0
5 }
```

Below the request details, the response section shows a red '400 Bad Request' status. The response body is:

```
1 {
2     "error": "El par\u00e1metro tipo proporcionado no corresponde con los permitidos"
3 }
```

Ahora sí. Y devuelve el error que se corresponde al tipo. Solo puede ser INDOOR o OUTDOOR.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help'. Below it is a toolbar with icons for back, forward, search, and settings, along with an 'Upgrade' button. The main workspace has tabs for 'pistas' and 'Padel', with 'POST Get' selected. The URL is 'pistasReservas'. On the left, there's a sidebar with various icons. The main area shows a POST request to 'Padel Club / GetPistas'. The 'Body' tab is active, showing a JSON payload:

```
1 {  
2   "nombre": "Pista Maginifica",  
3   "tipo": "OUTDOOR",  
4   "precioHora": 0  
5 }
```

Below the request, the response is shown: '201 Created' with a duration of '17 ms' and a size of '308 B'. The response body is identical to the request body.

Hago lo mismo pero con precio cero, ya que todo está preparado para que el precio tenga un valor por default si no se le indica uno válido.

```
12   "precioHora": "11.45"  
13 },  
14 {  
15   "id": 4,  
16   "nombre": "Pista Maginifica",  
17   "tipo": "OUTDOOR",  
18   "precioHora": "0.00"  
19 }  
20 ]
```

Si hago un get all de nuevo, vemos que en la BD está con ese precio por defecto.

The screenshot shows a REST client interface with the following details:

- Header Bar:** File, Edit, View, Help, Upgrade.
- Toolbar:** Back, Forward, Workspaces (Padel selected), More, Search, Settings, Notifications, AI icon, Upgrade button.
- Search Bar:** pistasReservas.
- Request Details:** HTTP Padel Club / GetPistas, Method: POST, URL: {{baseUrl}}/{{raiz}}/{{pista}}, Save, Share, Send button.
- Request Headers:** Docs, Params, Auth, Headers (8), Body (selected), Scripts, Settings, Cookies.
- Request Body:** raw, JSON (selected), Schema, Beautify. Content:

```
1 {  
2   "nombre": "Pista Negativa",  
3   "tipo": "OUTDOOR",  
4   "precioHora": -12  
5 }
```
- Response:** Body, 400 Bad Request, 10 ms, 303 B, Debug with AI, Preview, AI icon.
- Response Content:**

```
1 {  
2   "error": "El precio por hora debe ser un número positivo"  
3 }
```

Intento insertarla con un precio negativo, pero también se habían incluido validaciones para este caso.

File Edit View Help

← → Workspaces More 🔎 ⚙️ 🔔 Upgrade

HTTP Padel Club / GetById

GET {{baseUrl}} {{raiz}} {{pista}} /3

Send

Params

Query Params

Key	Value	Description
Key	Value	Description

Body 200 OK 6 ms 305 B

{ } JSON ▾ Preview Visualize

```
1 {  
2   "id": 3,  
3   "nombre": "Pista Central",  
4   "tipo": "INDOOR",  
5   "precioHora": "11.45"  
6 }
```

Cloud View Console Terminal Runner Vault AI

The screenshot shows a REST API testing interface. At the top, there's a navigation bar with File, Edit, View, Help, Workspaces, and More. On the right of the bar are icons for search, refresh, settings, notifications, and an upgrade button. Below the bar is a breadcrumb trail: POST Get, GET Get a, and GET GetById. To the right of the trail is a dropdown menu set to 'pistasReservas'. The main area has tabs for Docs, Params (which is selected), Auth, Headers (8), Body, Scripts, Settings, and Cookies. Under Params, there's a section for Query Params with a table. The table has columns for Key, Value, and Description. There are two rows: one with 'Key' and 'Value' empty, and another with 'Key' empty and 'Description' as 'Description'. Below this is a 'Body' section showing a successful 200 OK response with 6 ms and 305 B. It includes tabs for JSON, Preview, and Visualize, and a code editor showing the JSON response body. The response body is a single object with fields id, nombre, tipo, and precioHora. At the bottom, there are links for Cloud View, Console, Terminal, Runner, Vault, and AI.

Este es un Get By Id de una pista. Resultado correcto.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help'. Below the navigation is a toolbar with icons for back, forward, search, and settings, along with an 'Upgrade' button. The main workspace shows a collection named 'pistasReservas'. A specific request titled 'Padel Club / Modificar Nombre' is selected, showing a GET method with the URL template {{baseUrl}}/{{raiz}}/{{pista}}/3. The 'Body' tab is active, displaying a JSON payload:

```
1 {  
2   "nombre": "Pista Cambiada"  
3 }
```

Below the body, the response section shows a '200 OK' status with a response time of 9 ms and a size of 305 B. The response body is displayed as:

```
{  
  "id": 3,  
  "nombre": "Pista Central",  
  "tipo": "INDOOR",  
  "precioHora": "11.45"  
}
```

Probamos a cambiar el nombre de una pista. Operación correcta. Nos devuelve los datos anteriores de la pista. Vamos a hacer un get by id ahora para comprobar el cambio.

File Edit View Help

← → Workspaces More 🔎 ⚙️ 🔔 🔍 Upgrade

HTTP Padel Club / Modificar Nombre

PUT {{baseUrl}}/{{raiz}}/{{pista}}/3

Send

Docs Params Auth Headers (8) Body Scripts Settings Cookies

raw JSON Schema Beautify

```
1 {
2   "nombre": "Pista Cambiada"
3 }
```

Body 200 OK 20 ms 306 B

```
{ } JSON Preview Visualize
```

```
1 {
2   "id": 3,
3   "nombre": "Pista Cambiada",
4   "tipo": "INDOOR",
5   "precioHora": "11.45"
6 }
```

Cloud View Console Terminal Runner Vault AI

This screenshot shows a REST API testing interface. At the top, there's a navigation bar with File, Edit, View, Help, and a workspace dropdown. On the right of the bar are icons for search, add, settings, notifications, and an upgrade button. Below the bar is a header for 'pistasReservas' with navigation arrows and a dropdown menu. The main area contains an 'HTTP' request builder with the URL 'Padel Club / Modificar Nombre' and a 'PUT' method. The body is set to '{{baseUrl}}/{{raiz}}/{{pista}}/3'. A large text area shows the JSON payload: '1 { 2 "nombre": "Pista Cambiada" 3 }'. Below this, a status bar shows '200 OK' with response times and sizes. At the bottom, there are tabs for Cloud View, Console, Terminal, and various runner and vault options.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help'. Below the navigation is a toolbar with icons for back, forward, search, and settings, along with an 'Upgrade' button. The main workspace shows a collection named 'pistasReservas'. A specific request is selected: a 'DELETE' request to the URL 'Padel Club / Delete'. The request details show the method as 'DELETE' and the URL as '{{baseUrl}} {{raiz}} {{pista}} /3'. Below this, tabs for 'Docs', 'Params' (which is selected), 'Auth', 'Headers (6)', 'Body', 'Scripts', 'Settings', and 'Cookies' are visible. A 'Query Params' table is present but empty. The response section shows a status of '200 OK' with a response time of '14 ms' and a size of '306 B'. The response body is displayed as JSON:

```
1 {  
2   "id": 3,  
3   "nombre": "Pista Cambiada",  
4   "tipo": "INDOOR",  
5   "precioHora": "11.45"  
6 }
```

At the bottom, there are links for 'Cloud View', 'Console', 'Terminal', 'Runner', 'Vault', and other application tools.

Eliminamos ahora una pista.

Hacemos ahora de nuevo un gel all.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help'. Below the navigation is a toolbar with icons for search, refresh, settings, and notifications, along with an 'Upgrade' button. The main workspace shows a collection named 'pistasReservas'. A specific request is selected: 'HTTP Padel Club / Get all'. The method is set to 'GET', and the URL template is {{baseUrl}}/{{raiz}}/{{pista}}. Below the URL, there are buttons for 'Save', 'Share', and 'Send'. The 'Send' button is highlighted in blue. To the right of the URL, there are tabs for 'Docs', 'Params', 'Auth', 'Headers (8)', 'Body', 'Scripts', 'Settings', and 'Cookies'. The 'Body' tab is active, showing a table for 'Query Params'. Under the 'Body' tab, the status is '200 OK' with a response time of '6 ms' and a size of '378 B'. The response body is displayed as JSON:

```
1 [  
2 {  
3     "id": 2,  
4     "nombre": "Pista Este",  
5     "tipo": "INDOOR",  
6     "precioHora": "0.00"  
7 },  
8 {  
9     "id": 4,  
10    "nombre": "Pista Maginifica",  
11    "tipo": "OUTDOOR",  
12    "precioHora": "0.00"  
13 }  
14 ]
```

Se ha eliminado y solo nos quedan dos pistas.

The screenshot shows a REST client interface with the following details:

- Header Bar:** File, Edit, View, Help.
- Toolbar:** Back, Forward, Workspaces, More, Search, Refresh, Settings, Bell, Upgrade button.
- Request URL:** {{baseUrl}} {{raiz}} /reservas
- Method:** POST
- Body:** Raw JSON input:

```
1 {  
2   "pistaId":2,  
3   "fechaString":"2026-01-15",  
4   "horaInicio":"17:00",  
5   "horaFin":"18:00"  
6 }
```
- Response:** 201 Created, 21 ms, 320 B, Body content:

```
1 {  
2   "id": 1,  
3   "pistaId": 2,  
4   "fecha": "2026-01-15",  
5   "horaInicio": "17:00",  
6   "horaFin": "18:00"  
7 }
```
- Bottom Navigation:** Cloud View, Console, Terminal, Runner, Vault, etc.

Hago una reserva. El campo fechaString en el body es para hacer un req.body en el controller y luego transformarlo a un tipo Date que ya se llamará fecha, que es lo que espera TS. Por alguna razón la variable res no funcionaba en la url y tuve que poner reservas a mano, aunque en el environment parece que está bien.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help'. Below the navigation is a toolbar with icons for 'Workspaces', 'More', a search bar, and a 'Upgrade' button. The main workspace is titled 'pistasReservas'. A sub-workspace titled 'Padel Club / Get Reservas' is currently selected. The request section shows a 'GET' request with the URL template {{baseUrl}}/{{raiz}}/reservas. The 'Params' tab is selected, showing a table for 'Query Params' with columns for Key, Value, Description, and Bulk Edit. The table is empty. Below the request section, the response status is '200 OK' with a response time of 6 ms and a size of 411 B. The response body is displayed in JSON format:

```
7   "horaFin": "18:00:00"
8   },
9   {
10  "id": 2,
11  "pistaId": 2,
12  "fecha": "2026-01-15",
13  "horaInicio": "18:30:00",
14  "horaFin": "19:30:00"
15 }
16 ]
```

At the bottom, there are buttons for 'Cloud View', 'Console', 'Terminal', 'Runner', 'Vault', and other application tools.

Una reserva adicional para esa pista. Para meter datos para lo que viene despues.

The screenshot shows the Postman application interface. At the top, the menu bar includes File, Edit, View, Help, and an Upgrade button. The main workspace shows a collection named "pistasReservas". A POST request is selected, targeting the URL `HTTP Padel Club / Reserva`. The request body is defined as `POST {{baseUrl}}/{{raiz}}/reservas`, with the raw JSON content being:

```
1 {
2   "pistaId":2,
3   "fechaString":"2026-01-15",
4   "horaInicio":"17:30",
5   "horaFin":"19:30"
6 }
```

The response status is 409 Conflict, with a response time of 11 ms and a response size of 309 B. The response body is a JSON object with an error message:

```
1 {
2   "error": "La pista solicitada esta ocupada en esa franja de tiempo"
3 }
```

Intento solapar una reserva. Da fallo.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help'. Below the navigation bar, there are icons for 'Workspaces' (with a dropdown menu), 'More' (with a dropdown menu), a search bar, and other application settings. An 'Upgrade' button is visible in the top right corner.

The main workspace shows a collection named 'pistasReservas'. A specific request named 'GetFecha' is selected. The request details show it's a 'GET' method against the URL `HTTP Padel Club / {{baseUri}}/{{raiz}}/reservas/2026-01-15`. There are buttons for 'Save', 'Share', and 'Send'. Below the method selection, tabs for 'Docs', 'Params', 'Auth', 'Headers (8)', 'Body', 'Scripts', 'Settings', and 'Cookies' are present. The 'Params' tab is currently active.

The 'Query Params' section contains a table with columns 'Key', 'Value', and 'Descri...'. It lists two parameters: 'id' with value '1' and 'pistaId' with value '2'.

The 'Body' section shows the response status as '200 OK' with a response time of '7 ms' and a size of '411 B'. The response body is displayed as JSON:

```
1 [  
2 {  
3   "id": 1,  
4   "pistaId": 2,  
5   "fecha": "2026-01-15",  
6   "horaInicio": "17:00:00",  
7   "horaFin": "18:00:00"  
8 },  
9 {  
10  "id": 2,  
11  "pistaId": 2,  
12  "fecha": "2026-01-15",  
13  "horaInicio": "18:30:00",  
14  "horaFin": "19:30:00"  
15 }  
16 ]
```

At the bottom, there are various navigation and utility icons: Cloud View, Console, Terminal, Runner, Vault, and others.

Aunque era opcional filtro las reservas por fecha. Voy a hacerlo ahora por id de reserva, pero va a dar el mismo resultado, ya que las dos reservas introducidas son para la misma pista.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with File, Edit, View, Help, and a workspace dropdown set to Workspaces. To the right of the workspace dropdown are icons for search, filter, settings, notifications, and a profile picture, followed by an Upgrade button.

The main workspace area has tabs for Overview, pistas, and GET GetIdPista (which is currently selected). Below the tabs are buttons for Save, Share, and a copy icon. The URL bar shows the endpoint {{baseUrl}}/{{raiz}}/reservas/?idPista=2.

The request details section shows a GET method. Below it, the Params tab is selected, displaying a table with one row where idPista is set to 2. Other tabs like Docs, Headers, Body, Scripts, Settings, and Cookies are also visible.

The response section shows a 200 OK status with a response time of 10 ms and a body size of 411 B. The response body is displayed as JSON:

```
1 [  
2 {  
3     "id": 1,  
4     "pistaId": 2,  
5     "fecha": "2026-01-15",  
6     "horaInicio": "17:00:00",  
7     "horaFin": "18:00:00"  
8 },  
9 {  
10    "id": 2,  
11    "pistaId": 2,  
12    "fecha": "2026-01-15",  
13    "horaInicio": "18:30:00",  
14    "horaFin": "19:30:00"  
]
```

At the bottom, there are links for Cloud View, Console, Terminal, Runner, Vault, and other workspace management options.

Obtenemos un listado por idPista

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Help', and an 'Upgrade' button. Below the navigation is a toolbar with icons for search, refresh, settings, and notifications.

The main workspace shows a collection named 'pistasReservas'. A specific API endpoint named 'EliminarRes' is selected, which is a DELETE method. The URL is set to `HTTP Padel Club / EliminarRes`. The request body is defined as `DELETE {{baseUrl}} {{raiz}} /reservas/2`.

Below the request details, there are tabs for 'Docs', 'Params' (which is currently selected), 'Auth', 'Headers (8)', 'Body', 'Scripts', 'Settings', and 'Cookies'. The 'Params' tab shows a table for 'Query Params' with two rows:

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

At the bottom of the interface, there's a preview section for the 'Body' tab, showing a JSON response with the following data:

```
1 {  
2   "id": 2,  
3   "pistaId": 2,  
4   "fecha": "2026-01-15",  
5   "horaInicio": "18:30:00",  
6   "horaFin": "19:30:00"  
7 }
```

The preview also includes tabs for 'JSON', 'Preview', 'Visualize', and other options like 'Copy', 'Share', and 'Edit'.

At the very bottom, there are navigation links for 'Cloud View', 'Console', 'Terminal', 'Runner', 'Vault', and 'AI'.

Por último anulamos una reserva.