



ISWD613 APPLICACIONES WEB

NOMBRE ESTUDIANTE(S): Luis Andrés Guerrero Hinojosa
FECHA: 10/11/2025

TEMA: Proyecto 1

Introducción

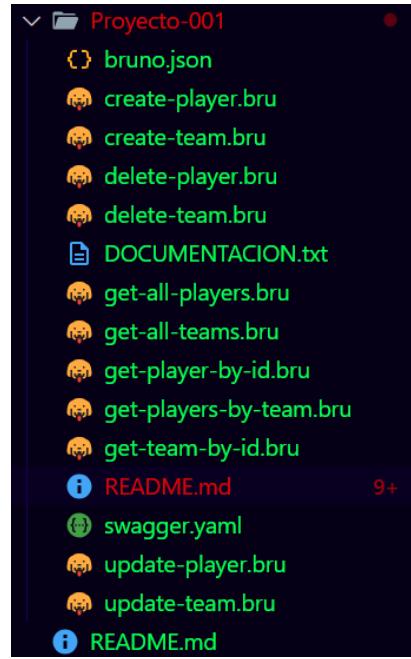
El proyecto “API Equipos y Jugadores” tiene como propósito diseñar y documentar una API RESTful completa que modele una relación uno a muchos entre las entidades Equipo (Team) y Jugador (Player). Este trabajo, desarrollado en el marco de la asignatura Construcción y Evolución de Aplicaciones Frontend durante noviembre de 2025, tiene como objetivo aplicar los principios fundamentales del diseño de APIs, la correcta implementación de métodos HTTP y la documentación técnica profesional.

La API busca demostrar una estructura robusta y coherente, capaz de manejar operaciones CRUD (crear, leer, actualizar y eliminar) tanto para equipos como para jugadores, garantizando la integridad referencial mediante el uso de claves foráneas. Cada equipo puede tener múltiples jugadores, y cada jugador pertenece exclusivamente a un equipo, estableciendo así una relación 1 a N clara y funcional. El modelo de datos está compuesto por las entidades Team, con atributos como id, name, country, foundedYear, stadium y coach; y Player, que incluye id, name, position, number, nationality, age y teamId como clave foránea que referencia al equipo correspondiente.

Desarrollo

La arquitectura del proyecto presenta una organización clara de carpetas, favoreciendo la mantenibilidad y escalabilidad del código. Los endpoints principales están estructurados bajo las rutas /teams y /players, con soporte para los métodos GET, POST, PUT y DELETE, además de un endpoint anidado /teams/{id}/players, que permite consultar todos los jugadores pertenecientes a un equipo específico. La API emplea el formato JSON para el intercambio de datos y respeta los principios RESTful, tales como la idempotencia, el uso de recursos y la ausencia de estado (statelessness). Asimismo, se han definido códigos de estado HTTP apropiados, como 200, 201, 204, 400 y 404, para reflejar de forma precisa las respuestas del servidor.

ISWD613 APLICACIONES WEB



El desarrollo del proyecto se complementó con el uso de herramientas profesionales como **Bruno API Client**, empleada para realizar las pruebas de los endpoints y validar el comportamiento de la API, y **Swagger/OpenAPI 3.0**, utilizada para documentar e interactuar con la API de manera visual e intuitiva. A través de Swagger UI se puede explorar cada ruta, enviar peticiones y observar las respuestas directamente desde el navegador.

Se procederá a poner las imágenes de bruno

```

1 [
2   {
3     "id": 1,
4     "name": "Leanne Graham",
5     "username": "Bret",
6     "email": "Sincere@april.biz",
7     "address": {
8       "street": "Kulas Light",
9       "suite": "Apt. 556",
10      "city": "Gwenborough",
11      "zipcode": "92998-3874",
12      "geo": {
13        "lat": "-37.3159",
14        "lng": "81.1496"
15      }
16    },
17    "phone": "1-770-736-8031 x56442"
  ]

```



ISWD613 APLICACIONES WEB

Bruno

Collection Edit View Window Help

bruno ...

Collections ↑

Q Search requests ...

> Sample API Collection
> JSONPlaceholder API...
v Clase 9 Proyecto - A...
 GET Obtener todos ...
 GET Obtener equip...
 POST Crear nuevo e...
 PUT Actualizar equi...
 DEL Eliminar equipo
 GET Obtener todos ...
 GET Obtener jugad...
 GET Obtener jugad...
 POST Crear nuevo ju...
 PUT Actualizar juga...
 DEL Eliminar jugad

GET https://jsonplaceholder.typicode.com/users/1

Params Body Headers Auth Vars Script Assert

Tests Docs Settings

Query

Name	Value
+ Add Param	

Path ②

Name	Value

Response Headers 24 Timeline Tests

200 OK 422ms 509B

```
1 {  
2   "id": 1,  
3   "name": "Leanne Graham",  
4   "username": "Bret",  
5   "email": "Sincere@april.biz",  
6   "address": {  
7     "street": "Kulas Light",  
8     "suite": "Apt. 556",  
9     "city": "Gwenborough",  
10    "zipcode": "92998-3874",  
11    "geo": {  
12      "lat": "-37.3159",  
13      "lng": "81.1496"  
14    },  
15    "phone": "1-770-736-8031 x56442",  
16    "website": "hildegard.org",  
17  }  
}
```

Q Search Cookies Dev Tools v2.14.1

ESP LAA 19:33 10/11/2025

Bruno

Collection Edit View Window Help

bruno ...

Collections ↑

Q Search requests ...

> Sample API Collection
> JSONPlaceholder API...
v Clase 9 Proyecto - A...
 GET Obtener todos ...
 GET Obtener equip...
 POST Crear nuevo e...
 PUT Actualizar equi...
 DEL Eliminar equipo
 GET Obtener todos ...
 GET Obtener jugad...
 GET Obtener jugad...
 POST Crear nuevo ju...
 PUT Actualizar juga...
 DEL Eliminar jugad

POST https://jsonplaceholder.typicode.com/users

Params Body * Headers¹ Auth Vars Script Assert

Tests Docs Settings JSON Prettify

Response Headers 25 Timeline Tests

201 Created 723ms 112B

```
1 {  
2   "name": "Barcelona FC",  
3   "country": "España",  
4   "foundedYear": 1899,  
5   "stadium": "Camp Nou",  
6   "id": 11  
7 }
```

Q Search Cookies Dev Tools v2.14.1

ESP LAA 19:34 10/11/2025



ISWD613 APLICACIONES WEB

Bruno

Collection Edit View Window Help

bruno ...

Collections ↑

Q Search requests ...

> Sample API Collection
> JSONPlaceholder API...
Clase 9 Proyecto - A...
 GET Obtener todos ...
 GET Obtener equip...
 POST Crear nuevo e...
 PUT Actualizar equi...
 DEL Eliminar equipo
 GET Obtener todos ...
 GET Obtener jugad...
 GET Obtener jugad...
 POST Crear nuevo ju...
 PUT Actualizar juga...
 DEL Eliminar jugad...
PUT https://jsonplaceholder.typicode.com/users/1

Params Body * Headers¹ Auth Vars Script Assert
Tests Docs Settings JSON Prettify

Response Headers²³ Timeline Tests
200 OK 537ms 141B

```
1+ {  
2   "name": "Barcelona FC",  
3   "country": "España",  
4   "foundedYear": 1899,  
5   "stadium": "Camp Nou",  
6   "coach": "Xavi Hernández",  
7   "id": 1  
8 }
```

Bruno

Collection Edit View Window Help

bruno ...

Collections ↑

Q Search requests ...

> Sample API Collection
> JSONPlaceholder API...
Clase 9 Proyecto - A...
 GET Obtener todos ...
 GET Obtener equip...
 POST Crear nuevo e...
 PUT Actualizar equi...
 DEL Eliminar equipo
 GET Obtener todos ...
 GET Obtener jugad...
 GET Obtener jugad...
 POST Crear nuevo ju...
 PUT Actualizar juga...
 DEL Eliminar jugad...
DELETE https://jsonplaceholder.typicode.com/users/1

Params Body Headers Auth Vars Script Assert
Tests Docs Settings

Query
Name Value
+ Add Param Bulk Edit
Path
Name Value

Response Headers²³ Timeline Tests
200 OK 722ms 2B

```
1 {}
```



ISWD613 APLICACIONES WEB

Bruno

Collection Edit View Window Help

bruno ...

Collections ↑↓

Q Search requests ...

> Sample API Collection
> JSONPlaceholder API...
Clase 9 Proyecto - A...
 GET Obtener todos ...
 GET Obtener equip...
 POST Crear nuevo e...
 PUT Actualizar equi...
 DEL Eliminar equipo
 GET Obtener todos ...
 GET Obtener jugad...
 GET Obtener jugad...
 POST Crear nuevo ju...
 PUT Actualizar juga...
 DEL Eliminar jugad...
+ Add Param Bulk Edit

Params Body Headers Auth Vars Script Assert

Tests Docs Settings

Query

Name Value

Path ⓘ

Name Value

GET https://jsonplaceholder.typicode.com/posts

Developer Mode | No environments

PUT Actualizar ... | DELETE Elimina... | GET Obtener t... | > +

Response Headers 24 Timeline Tests

200 OK 747ms 26.88KB

```
[  
  {  
    "userId": 1,  
    "id": 1,  
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",  
    "body": "quia et suscipit\\nrecusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est autem sunt rem eveniet architecto"  
  },  
  {  
    "userId": 2,  
    "id": 2,  
    "title": "qui est esse",  
    "body": "est rerum tempore vitae\\nsequi sint nihil\\n reprehenderit dolor beatae ea dolores neque\\nfugiat bla"  
  }]
```

Q Search Cookies Dev Tools v2.14.1

ESP LAA 19:35 10/11/2025

Bruno

Collection Edit View Window Help

bruno ...

Collections ↑↓

Q Search requests ...

> Sample API Collection
> JSONPlaceholder API...
Clase 9 Proyecto - A...
 GET Obtener todos ...
 GET Obtener equip...
 POST Crear nuevo e...
 PUT Actualizar equi...
 DEL Eliminar equipo
 GET Obtener todos ...
 GET Obtener jugad...
 GET Obtener jugad...
 POST Crear nuevo ju...
 PUT Actualizar juga...
 DEL Eliminar jugad...
+ Add Param Bulk Edit

Params Body Headers Auth Vars Script Assert

Tests Docs Settings

Query

Name Value

Path ⓘ

Name Value

GET https://jsonplaceholder.typicode.com/posts/1

Developer Mode | No environments

DELETE Elimina... | GET Obtener t... | > +

Response Headers 24 Timeline Tests

200 OK 988ms 292B

```
{  
  "userId": 1,  
  "id": 1,  
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",  
  "body": "quia et suscipit\\nrecusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est autem sunt rem eveniet architecto"  
}
```

Q Search Cookies Dev Tools v2.14.1

ESP LAA 19:35 10/11/2025



ISWD613 APLICACIONES WEB

Bruno

Collection Edit View Window Help

bruno ...

Collections ↑

Q Search requests ...

> Sample API Collection
> JSONPlaceholder API...
Clase 9 Proyecto - A...
GET Obtener todos ...
GET Obtener equipo...
POST Crear nuevo e...
PUT Actualizar equi...
DEL Eliminar equipo
GET Obtener todos ...
GET Obtener jugad...
GET Obtener jugad...
POST Crear nuevo ju...
PUT Actualizar juga...
DEL Eliminar jugad

Developer Mode | No environments

GET https://jsonplaceholder.typicode.com/users/1/posts

Params Body Headers Auth Vars Script Assert

Tests Docs Settings

Query

Name	Value
+ Add Param	

Bulk Edit

Path ⓘ

Name	Value

Response Headers 23 Timeline Tests

200 OK 754ms 2.66KB

```
[  
  {  
    "userId": 1,  
    "id": 1,  
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",  
    "body": "quia et suscipit\\nrecusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est autem sunt rem eveniet architecto"  
  },  
  {  
    "userId": 1,  
    "id": 2,  
    "title": "qui est esse",  
    "body": "est rerum tempore vitae\\nsequi sint nihil\\n reprehenderit dolor beatae ea dolores neque\\nfugiat blanditiis enim ipsum voluptatem\\n fugiat"  
  }]
```

Q Search Cookies Dev Tools v2.14.1

ESP LAA 19:35 10/11/2025

Bruno

Collection Edit View Window Help

bruno ...

Collections ↑

Q Search requests ...

> Sample API Collection
> JSONPlaceholder API...
Clase 9 Proyecto - A...
GET Obtener todos ...
GET Obtener equipo...
POST Crear nuevo e...
PUT Actualizar equi...
DEL Eliminar equipo
GET Obtener todos ...
GET Obtener jugad...
GET Obtener jugad...
POST Crear nuevo ju...
PUT Actualizar juga...
DEL Eliminar jugad

Developer Mode | No environments

POST https://jsonplaceholder.typicode.com/posts

Params Body * Headers¹ Auth Vars Script Assert

Tests Docs Settings

JSON Prettify

```
{  
  "name": "Lionel Messi",  
  "position": "Delantero",  
  "number": 10,  
  "nationality": "Argentina",  
  "teamId": 1  
}
```

Response Headers 25 Timeline Tests

201 Created 569ms 129B

```
{  
  "name": "Lionel Messi",  
  "position": "Delantero",  
  "number": 10,  
  "nationality": "Argentina",  
  "teamId": 1,  
  "id": 101  
}
```

Q Search Cookies Dev Tools v2.14.1

ESP LAA 19:36 10/11/2025



ISWD613 APLICACIONES WEB

The screenshot shows the Bruno API testing tool interface. On the left, there's a sidebar with collections like 'Sample API Collection' and 'Clase 9 Proyecto - A...'. The main area shows a 'PUT' request to 'https://jsonplaceholder.typicode.com/posts/1'. The 'Body' tab displays the following JSON payload:

```
1+ {
2   "name": "Lionel Messi",
3   "position": "Delantero",
4   "number": 10,
5   "nationality": "Argentina",
6   "teamId": 1,
7   "age": 36
8 }
```

The 'Response' tab shows the JSON response from the API:

```
1+ {
2   "name": "Lionel Messi",
3   "position": "Delantero",
4   "number": 10,
5   "nationality": "Argentina",
6   "teamId": 1,
7   "age": 36,
8   "id": 1
9 }
```

At the bottom, the status bar shows 'v2.14.1' and the date '10/11/2025'.

El proceso de aprendizaje permitió comprender a profundidad el diseño de relaciones entre entidades en APIs, la elaboración de documentación técnica mediante estándares de la industria, y la ejecución de pruebas funcionales en entornos de desarrollo controlados. Las evidencias del proyecto incluyen capturas de la colección completa en Bruno, las peticiones GET y POST, el funcionamiento del endpoint clave /teams/{id}/players, la visualización en Swagger UI y la estructura organizada del proyecto en VS Code.

En swagger



ISWD613 APLICACIONES WEB

The screenshot shows the Swagger Editor interface for a REST API. The left panel displays the API definition with code snippets for players and teams. The right panel shows operations like GET /teams/{id}, POST /players, and DELETE /players/{id}.

The screenshot shows the Swagger Editor interface for a REST API, identical to the one above but with a different timestamp.



ISWD613 APLICACIONES WEB

Conclusiones

Finalmente, este proyecto demuestra la capacidad de integrar los conceptos teóricos y prácticos del desarrollo de APIs RESTful, aplicando buenas prácticas de diseño, estructuración y documentación. Constituye una base sólida para la construcción de sistemas backend escalables, mantenibles y bien documentados, alineados con los estándares modernos de desarrollo web.

Repositorio GitHub:

<https://github.com/2025-b-sw-web-gr1/guerrero-2025-b-lagh-web-gr1/tree/main/Proyecto-001>

Autor: Guerrero 2025B — **Fecha:** Noviembre 2025

❖ PROMPT GENERADOR

PROYECTO 001

Quiero que generes un proyecto completo tipo RESTful API que documente una relación 1-a-muchos (por ejemplo:

- Un equipo de fútbol tiene muchos jugadores,
- Una receta tiene muchos ingredientes, etc.).

El proyecto debe incluir:

- Archivos Bruno (.bru) para todos los endpoints.
- Documentación Swagger (OpenAPI YAML).
- README.md con descripción, modelos, endpoints, estructura, y autores.
- DOCUMENTACION.txt con enlace a documentación ampliada.

ISWD613 APLICACIONES WEB

- Todo organizado en una carpeta llamada Proyecto-001.

🔗 OBJETIVO

Diseñar y documentar una API RESTful que cumpla los principios de diseño REST, mostrando una relación 1 → N entre dos entidades.

Ejemplo base:

- Team (1) tiene muchos Players (N)
- Player pertenece a un solo Team

📄 ESTRUCTURA DE CONTENIDOS

README.md debe incluir:

1. Descripción del proyecto
Breve explicación del propósito del API y su utilidad.
2. Introducción teórica
Qué es una API RESTful, métodos HTTP usados y concepto de stateless.
3. Modelo de datos (Team) y Player)
Explicación de entidades y atributos.
4. Diagrama ASCII de la relación 1-a-muchos
5. Team (1) —————< Player (N)
6. id id
7. name name
8. country position
9. teamId (FK)
10. Lista completa de endpoints
Incluye todos los métodos CRUD para ambas entidades y /teams/{id}/players.
11. Estructura del proyecto (árbol de archivos)
12. Proyecto-001/
 13. └── README.md
 14. └── DOCUMENTACION.txt
 15. └── swagger/
 16. └── football-api.yaml
 17. └── bruno/
 18. └── 01-get-teams.bru
 19. └── 02-get-team-by-id.bru
 20. └── 03-post-team.bru
 21. └── ...
 22. └── 11-get-players-by-team.bru
 23. Instrucciones de uso con Bruno
 - Cómo ejecutar cada archivo .bru.
 - Cómo probar las respuestas HTTP.
 24. Documentación Swagger (OpenAPI YAML)
Debe incluir todos los endpoints, parámetros, cuerpos de petición y respuestas.
 25. Conceptos RESTful aplicados
Recursos en plural, uso correcto de métodos, códigos HTTP coherentes.
 26. Información del autor o grupo
Nombres, correos institucionales, carrera y nombre del proyecto.

ISWD613 APLICACIONES WEB

📄 DOCUMENTACION.txt debe incluir:

- Enlace a un documento en Google Docs con la documentación extendida.
- Descripción breve del contenido (teoría, endpoints, ejemplos y conclusiones).

💾 EJEMPLOS DE DATOS

Team:

```
{  
  "name": "FC Barcelona",  
  "country": "España",  
  "founded": 1899,  
  "stadium": "Camp Nou",  
  "coach": "Xavi Hernández"
```

}

Player:

```
{  
  "name": "Lionel Messi",  
  "position": "Delantero",  
  "number": 10,  
  "nationality": "Argentina",  
  "teamId": 1  
}
```

🌐 BASE URL

<https://api.ejemplo.com>

⚡ CÓDIGOS HTTP

Código	Descripción
200 OK	GET exitoso
201 Created	POST exitoso
204 No Content	DELETE exitoso
400 Bad Request	Datos inválidos
404 Not Found	Recurso no encontrado

❖ PUNTOS CLAVE

- /teams/{id}/players demuestra la relación 1-a-muchos
- teamId es una clave foránea obligatoria en Player
- Se deben usar correctamente los métodos HTTP
- Swagger debe ser legible, estructurado y en formato OpenAPI 3.0
- Archivos Bruno numerados y organizados secuencialmente

📦 ESTRUCTURA DE LOS ARCHIVOS BRUNO

```
meta {  
  name: Nombre descriptivo del endpoint  
  type: http  
  seq: número secuencial (1–11)
```



ISWD613 APLICACIONES WEB

```
}
```

```
get|post|put|delete {
    url: https://api.ejemplo.com/ruta
    body: none|json
    auth: none|bearer_token|basic
}

headers {
    Content-Type: application/json
    Authorization: Bearer <token> (si aplica)
}

body:json {
    { datos de ejemplo }
}
```

🔗 ENTREGA FINAL

- Carpeta raíz: Proyecto-001/
- Commit:
"Proyecto-001: API Equipos y Jugadores (relación 1-a-muchos) con Bruno y Swagger"
- Subir a GitHub
-