

# Consumo API REST para registro Equipo

URL	Method
http://localhost:8080/ sdc/equipo/registro	POST

## 1. Argumentos de entrada:

Ninguno

## 2. Argumentos de salida:

- **Http Status:** HTTP 201 Created
- **Body:**

```
{
  "nombreEquipo": "Fibra Tóxica",
  "colorEquipo": "Azul",
  "mascota": "Telito"
}
```

The screenshot displays the Postman interface for a REST client. The top section shows the request configuration: a POST request to the URL `localhost:8080/sdc/equipo/registro`. The request body is a JSON object: `{ "nombreEquipo": "Fibra Tóxica", "colorEquipo": "Azul", "mascota": "Telito" }`. The bottom section shows the response details: the status is `201 Created`, the time taken is `41 ms`, and the size is `202 B`. The response body is a JSON object: `{ "estado": "creado" }`.

```
{
  "nombreEquipo": "Hormigon Armado",
  "colorEquipo": "Plomo",
  "mascota": "Hormiga"
}
```

The screenshot shows the Postman interface with a POST request to `localhost:8080/sdci/equipo/registro`. The request body is a JSON object: `{ "nombreEquipo": "Hormigon Armado", "colorEquipo": "Plomo", "mascota": "Hormiga" }`. The response status is 201 Created, with a time of 17 ms and a size of 202 B. The response body is `{ "estado": "creado" }`.

```
{
  "nombreEquipo": "Naranja Mecánica",
  "colorEquipo": "Naranja",
  "mascota": "Naranjito"
}
```

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/sdci/equipo/registro`. The request body is a JSON object: `{ "nombreEquipo": "Naranja Mecánica", "colorEquipo": "Naranja", "mascota": "Naranjito" }`. The response status is 201 Created, with a time of 280 ms and a size of 202 B. The response body is `{ "estado": "creado" }`.

### 3. Datos argumentos de salida:

Campo	Tipo de Dato	Observaciones
nombreEquipo	cadena	-
colorEquipo	cadena	-
mascota	cadena	-

# Consumo API REST para registro Participante

URL	Method
http://localhost:8080/sdci/participante/registro	POST

## 1. Argumentos de entrada:

Ninguno

## 2. Argumentos de salida:

- **Http Status:** HTTP 201 Created
- **Body:**

```
{
  "equipo": {
    "id": 1,
    "carrera": "Ingenieria Telecomunicaciones",
    "codigo": "20206404",
    "tipoParticipante": "estudiante"
  }
}
```

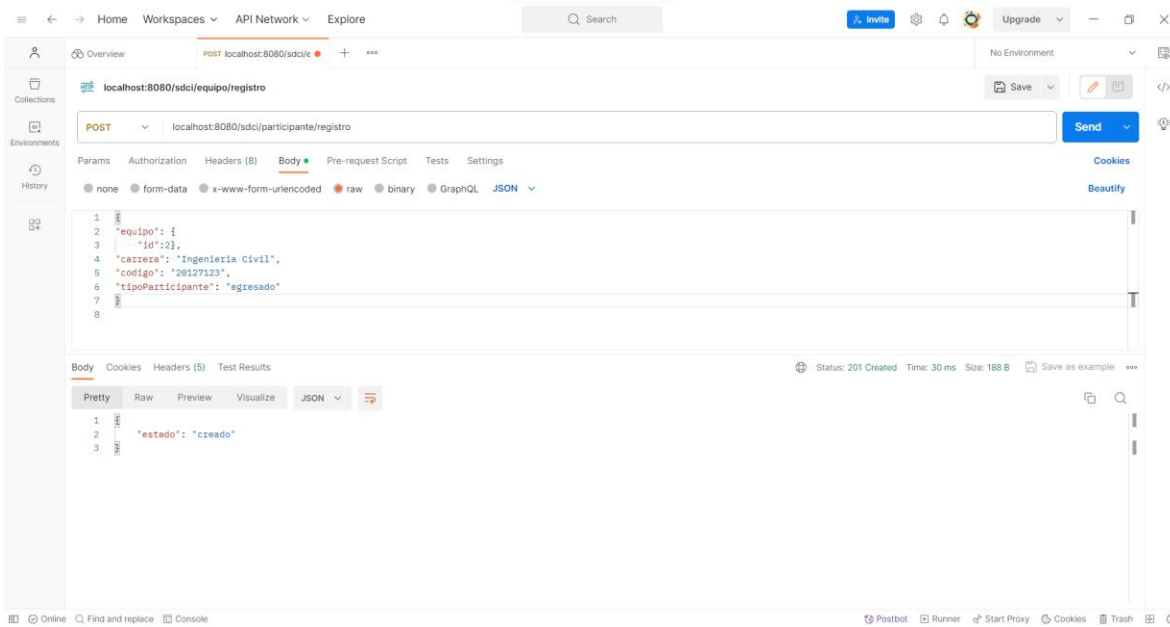
The screenshot displays the Postman interface for a REST client. The top section shows the request configuration: a POST request to the URL `localhost:8080/sdci/participante/registro`. The request body is set to JSON and contains the following data:

```
{
  "equipo": {
    "id": 1,
    "carrera": "Ingenieria Telecomunicaciones",
    "codigo": "20206404",
    "tipoParticipante": "estudiante"
  }
}
```

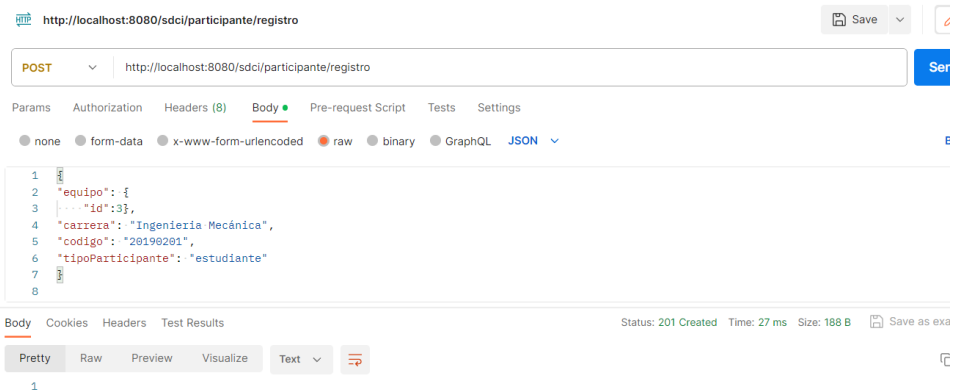
The bottom section shows the response details. The status is `201 Created`, the time taken is `64 ms`, and the size is `188 B`. The response body is:

```
{
  "estado": "creado"
}
```

```
{
  "equipo": {
    "id": 2,
  },
  "carrera": "Ingenieria Civil",
  "codigo": "20127123",
  "tipoParticipante": "egresado"
}
```



```
{
  "equipo": {
    "id": 3,
  },
  "carrera": "Ingenieria Mecánica",
  "codigo": "20190201",
  "tipoParticipante": "estudiante"
}
```



### 3. Datos argumentos de salida:

Campo	Tipo de Dato	Observaciones
equipo	Json	Null, si no hay equipo registrado
equipo.id	entero	-
carrera	Cadena	-
codigo	Cadena	-
tipoParticipante	cadena	-

# Consumo API REST para registro Deporte

URL	Method
http://localhost:8080/ sdc/deporte/registro	POST

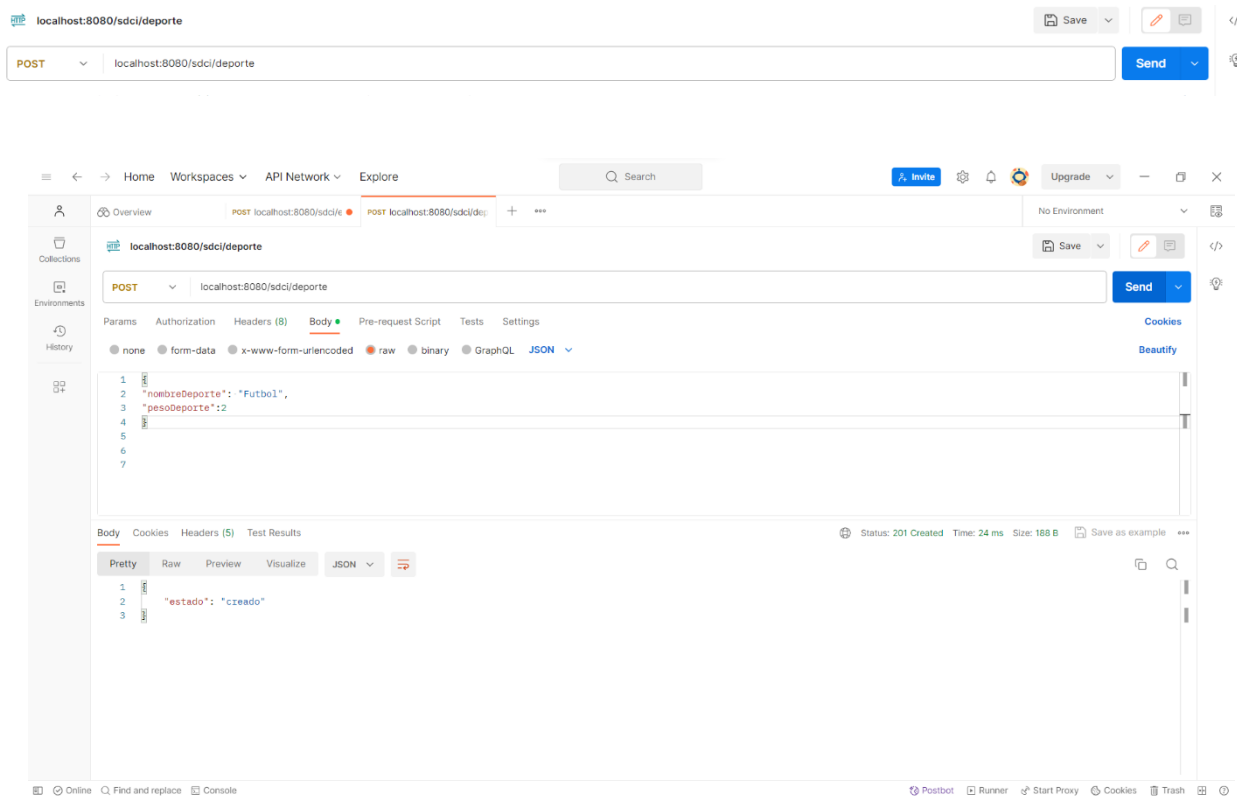
## 1. Argumentos de entrada:

Ninguno

## 2. Argumentos de salida:

- **Http Status:** HTTP 201 Created
- **Body:**
- 

```
{  
  "nombreDeporte": "Futbol",  
  "pesoDeporte": 2  
}
```



```
{
  "nombreDeporte": "Ping Pong",
  "pesoDeporte": 4
}
```

HTTP <http://localhost:8080/sdci/deporte/registro>

**POST** <http://localhost:8080/sdci/deporte/registro>

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ☐

```
1 {
2   "nombreDeporte": "Ping Pong",
3   "pesoDeporte": 4
4 }
5
6
```

Body Cookies Headers Test Results Status: 201 Created Time: 23 ms Size: 1

Pretty Raw Preview Visualize Text ☐

1

### 3. Datos argumentos de salida:

Campo	Tipo de Dato	Observaciones
nombreDeporte	cadena	-
pesoDeporte	entero	-



## Consumo API REST para registro Partido

URL	Method
http://localhost:8080/sdci/partido/registro?iddeporte={id}	POST

### 1. Argumentos de entrada:

Campo	Tipo de Dato	Observaciones
id	entero	-

### 2. Argumentos de salida:

**Http Status:** HTTP 201 Created

**Body:**

```
{
  "equipoA": {
    "id":1},
  "equipoB": {
    "id":2},
  "scoreA": 5,
  "scoreB":4
}
```



Overview | POST localhost:8080/sdci/c | POST localhost:8080/sdci/c | + ... | No Environment

localhost:8080/sdci/deporte

POST | localhost:8080/sdci/partido/registro?iddeporte=1

Params | Authorization | Headers (8) | **Body** | Pre-request Script | Tests | Settings

none | form-data | x-www-form-urlencoded | **raw** | binary | GraphQL | JSON

```
1 {
2   "equipoA": {
3     "id": 1,
4   "equipoB": {
5     "id": 2,
6   "scoreA": 5,
7   "scoreB": 4
8 }
9
10 }
```

Body | Cookies | Headers (5) | Test Results

Status: 201 Created | Time: 58 ms | Size: 188 B | Save as example

Pretty | Raw | Preview | Visualize | JSON

```
1 {
2   "estado": "creado"
3 }
```

Home | Workspaces | API Network | Explore

Overview | POST localhost:8080/sdci/c | POST localhost:8080/sdci/c | + ... | No Environment

localhost:8080/sdci/deporte

POST | localhost:8080/sdci/partido/registro?iddeporte=1

Params | Authorization | Headers (8) | **Body** | Pre-request Script | Tests | Settings

Query Params

Key	Value	Description
<input checked="" type="checkbox"/> iddeporte	1	
Key	Value	Description

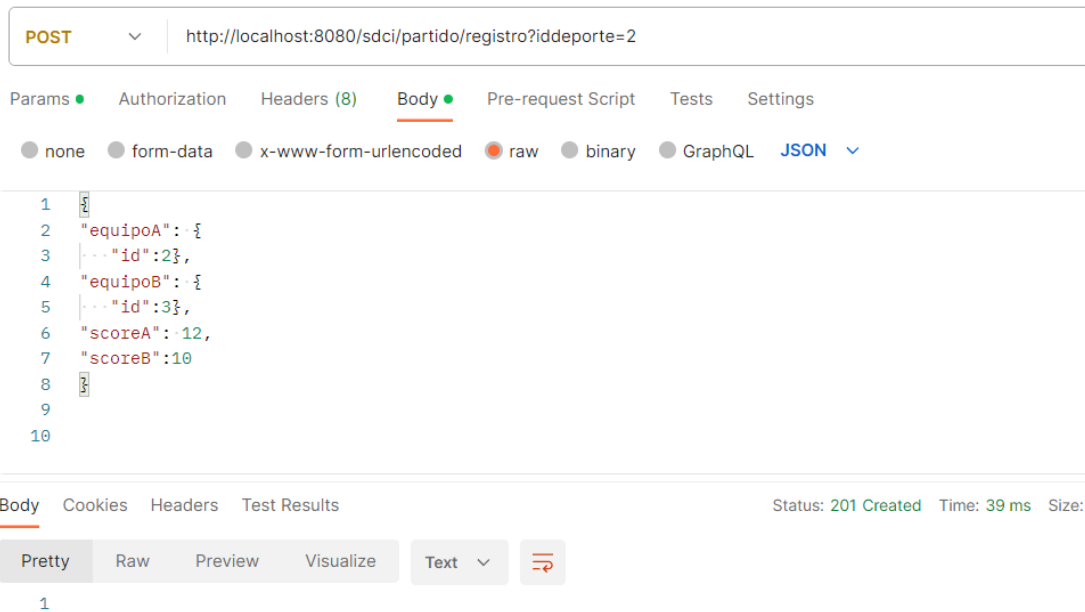
Body | Cookies | Headers (5) | Test Results

Status: 201 Created | Time: 58 ms | Size: 188 B | Save as example

Pretty | Raw | Preview | Visualize | JSON

```
1 {
2   "estado": "creado"
3 }
```

```
{
  "equipoA": {
    "id": 2,
  "equipoB": {
    "id": 3,
  "scoreA": 12,
  "scoreB": 10
  }
}
```



### 3. Datos argumentos de salida:

Campo	Tipo de Dato	Observaciones
equipoA	Json	Null, si no hay equipoB registrado
equipoA.id	entero	-
equipoB	Json	Null, si no hay equipoB registrado
equipoB.id	Entero	-
scoreA	Entero	-
scoreB	entero	-

En este caso mencionar que, en vista que cuando se va a registrar un partido, se debe añadir un registro a la tabla de historial de partidos, y en esta última tabla se requiere la primary key de la tabla de deporte (iddeporte), notamos que requerimos el iddeporte, por lo que, en la url, colocamos el id, ya que, de este modo, registramos el partido del mencionado, y a su vez, cumplimos con la condición de registrarlos en la tabla de historial de partidos.

## Consumo API REST para obtener la lista de participantes de un partido

URL	Method
http://localhost:8080/sdci/partido/getparticipantes/{id}	GET

### 1. Argumentos de entrada:

Campo	Tipo de Dato	Observaciones
id	entero	-

### 2. Argumentos de salida:

Opción 1: Solicitud correcta

Http Status: HTTP 200 OK

Body:

```
{
  "Participante": [
    {
      "id": 1,
      "equipo": {
        "id": 1,
        "nombreEquipo": "Fibra Tóxica",
        "colorEquipo": "Azul",
        "mascota": "Telito"
      },
      "carrera": "Ingenieria Telecomunicaciones",
      "codigo": 20206404,
      "tipoParticipante": "estudiante"
    },
    {
      "id": 2,
      "equipo": {
        "id": 2,
        "nombreEquipo": "Hormigon Armado",
        "colorEquipo": "Plomo",
        "mascota": "Hormiga"
      },
      "carrera": "Ingenieria Civil",
      "codigo": 20127123,
      "tipoParticipante": "egresado"
    }
  ]
}
```

Opción 2: Error en los parámetros de la solicitud  
Http Status: HTTP 400 Bad Request  
Body:

```
{
  "error": "No se encontro el partido indicado"
}
```

3. Datos argumentos de salida:

Campo	Tipo de Dato	Observaciones
id	entero	-
equipo	json	Null, si no tiene equipo
equipo.id	entero	-
equipo.nombreEquipo	Cadena	-
equipo.colorEquipo	Cadena	-
equipo.mascota	cadena	-
carrera	cadena	-
codigo	entero	-
tipoParticipante	cadena	-

4. Ejemplos  
Solicitud correcta:

localhost:8080/sdci/partido/getparticipantes/1

GET

localhost:8080/sdci/partido/getparticipantes/1

Ser

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 21 msSize: 551 BSave as exe

PrettyRawPreviewVisualizeJSON

```
1
2  "Participante": [
3    {
4      "id": 1,
5      "equipo": {
6        "id": 1,
7        "nombreEquipo": "Fibra Tóxica",
8        "colorEquipo": "Azul",
9        "mascota": "Telito"
10     },
11     "carrera": "Ingeniería Telecomunicaciones",
12     "codigo": 28286484,
13     "tipoParticipante": "estudiante"
14   },
15   {
16     "id": 2,
17     "equipo": {
18       "id": 2,
19       "nombreEquipo": "Hormigon Armado",
20       "colorEquipo": "Plomo",
21       "mascota": "Hormiga"
22     },
23     "carrera": "Ingeniería Civil",
24     "codigo": 28127123,
25     "tipoParticipante": "egresado"
26   }
27 ]
```

## Solicitud incorrecta:

The screenshot displays a REST client interface with a sidebar on the left containing icons for a user, Collections, Environments, History, and a grid icon. The main panel shows a collection named 'localhost:8080/sdci/deporte' with a single request. The request is a GET method to the URL 'localhost:8080/sdci/partido/getparticipantes/100'. The 'Body' tab is selected, showing a JSON response in 'Pretty' format: 

```
1 {
2   "error": "No se encontro el partido indicado"
3 }
```

 The top bar indicates the current request is a GET to 'localhost:8080/sdci/de...'.

## Consumo API REST para obtener la lista de participantes por partido de un determinado equipo

URL	Method
http://localhost:8080/sdci/partido/getparticipantes/idpartido?idequipo={id}	GET

### 1. Argumentos de entrada:

Campo	Tipo de Dato	Observaciones
id	entero	-

### 2. Argumentos de salida:

**Opción 1:** Solicitud correcta

**Http Status:** HTTP 200 OK

**Body:**

```
{
  "Participante": [
    {
      "id": 1,
      "equipo": {
        "id": 1,
        "nombreEquipo": "Fibra Tóxica",
        "colorEquipo": "Azul",
        "mascota": "Telito"
      },
      "carrera": "Ingenieria Telecomunicaciones",
      "codigo": 20206404,
      "tipoParticipante": "estudiante"
    }
  ]
}
```

**Opción 2:** Error en los parámetros de la solicitud

**Http Status:** HTTP 400 Bad Request

**Body:**

```
{
  "error": "No se encontro el equipo indicado"
}
```

### 3. Datos argumentos de salida:

Campo	Tipo de Dato	Observaciones
id	entero	-
equipo	json	Null, si no tiene equipo
equipo.id	entero	-
equipo.nombreEquipo	cadena	-
equipo.colorEquipo	cadena	-
equipo.mascota	cadena	-
carrera	cadena	-
codigo	entero	-
tipoParticipante	cadena	-

### 4. Ejemplos

#### Solicitud correcta:

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

- URL:** localhost:8080/sdci/deporte
- Method:** GET
- Query Params:** idequipo=1
- Status:** 200 OK, Time: 14 ms, Size: 372
- Response Body (JSON):**

```
{
  "Participante": [
    {
      "id": 1,
      "equipo": {
        "id": 1,
        "nombreEquipo": "Fibra Tóxica",
        "colorEquipo": "Azul",
        "mascota": "Telito"
      },
      "carrera": "Ingeniería Telecomunicaciones",
      "codigo": 20206404,
      "tipoParticipante": "estudiante"
    }
  ]
}
```

#### Solicitud incorrecta:



GET

localhost:8080/sdci/partido/getparticipantes/?idequipo=100

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params			
	Key	Value	Description
<input checked="" type="checkbox"/>	idequipo	100	
	Key	Value	Description

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 2

PrettyRawPreviewVisualizeJSON

```
1
2  "error": "No se encontro el equipo indicado"
3
```

# Consumo API REST para obtener la lista del historial de partidos jugados

URL	Method
http://localhost:8080/sdci/partido/gethistorialpartidos	GET

## 1. Argumentos de entrada:

Ninguno

## 2. Argumentos de salida:

**Opción 1:** Solicitud correcta

**Http Status:** HTTP 200 OK

**Body:**

```
{
  "Historial": [
    {
      "id": 1,
      "partidoIdpartido": {
        "id": 1,
        "equipoA": {
          "id": 1,
          "nombreEquipo": "Fibra Tóxica",
          "colorEquipo": "Azul",
          "mascota": "Telito"
        },
        "equipoB": {
          "id": 2,
          "nombreEquipo": "Hormigon Armado",
          "colorEquipo": "Plomo",
          "mascota": "Hormiga"
        },
        "scoreA": 5,
        "scoreB": 4
      },
      "deporteIddeporte": {
        "id": 1,
        "nombreDeporte": "Futbol",
        "pesoDeporte": 2
      },
      "horaFecha": "2023-10-22T22:45:41Z"
    }
  ]
}
```

### 3. Datos argumentos de salida:

Campo	Tipo de Dato	Observaciones
id	entero	-
partidoIdpartido	json	Null, si no hay partido registrado
partidoIdpartido.id	entero	-
partidoIdpartido.id.equipoA	json	Null, si no hay un equipob registrado
partidoIdpartido.id.equipoA.id	entero	-
partidoIdpartido.id.equipoA.nombreEquipo	cadena	-
partidoIdpartido.id.equipoA.colorEquipo	cadena	-
partidoIdpartido.id.equipoA.mascota	cadena	-
partidoIdpartido.id.equipoB	json	Null, si no hay un equipoB registrado
partidoIdpartido.id.equipoB.id	entero	-
partidoIdpartido.id.equipoB.nombreEquipo	cadena	-
partidoIdpartido.id.equipoB.colorEquipo	cadena	-
partidoIdpartido.id.equipoB.mascota	cadena	-
partidoIdpartido.scoreA	Entero	-
partidoIdpartido.scoreB	entero	-
deporteIddeporte	json	Null, si no hay deporte registrado
deporteIddeporte.id	entero	-
deporteIddeporte.nombreDeporte	cadena	-
deporteIddeporte.pesoDeporte	entero	-
horaFecha	date	-

## 4. Ejemplos

localhost:8080/sdci/partido/gethistorialpartidos

GETlocalhost:8080/sdci/partido/gethistorialpartidos

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

Query Params

Key	Value	Description	...
Key	Value	Description	

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 24 msSize: 526 BSave as

PrettyRawPreviewVisualizeJSON

```
1  {
2    "Historial": [
3      {
4        "id": 1,
5        "partidoIdpartido": {
6          "id": 1,
7          "equipoA": {
8            "id": 1,
9            "nombreEquipo": "Fibra Tóxica",
10           "colorEquipo": "Azul",
11           "mascota": "Telito"
12         },
13         "equipoB": {
14           "id": 2,
15           "nombreEquipo": "Hormigon Armado",
16           "colorEquipo": "Plomo",
17           "mascota": "Hormiga"
18         },
19         "scoreA": 5,
20         "scoreB": 4
21       },
22     ]
23   }
```

localhost:8080/sdci/partido/gethistorialpartidos

GETlocalhost:8080/sdci/partido/gethistorialpartidos

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

Query Params

Key	Value	Description	.
Key	Value	Description	

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 24 msSize: 526 BSav

PrettyRawPreviewVisualizeJSON

```
10     "colorEquipo": "Azul",
11     "mascota": "Telito"
12   },
13   "equipoB": {
14     "id": 2,
15     "nombreEquipo": "Hormigon Armado",
16     "colorEquipo": "Plomo",
17     "mascota": "Hormiga"
18   },
19   "scoreA": 5,
20   "scoreB": 4
21 },
22 "deporteIddeporte": 1,
23   "id": 1,
24   "nombreDeporte": "Futbol",
25   "pesoDeporte": 2
26 },
27 "horaFecha": "2023-10-22T22:45:41Z"
28 }
29 ]
30 }
```

Find and replaceConsolePostshotRunnerStart Prev

## Consumo API REST para obtener la lista del historial de partidos jugados de un determinado equipo.

URL	Method
http://localhost:8080/sdci/partido/gethistorialpartidos?idequipo={id}	GET

### 1. Argumentos de entrada:

Campo	Tipo de Dato	Observaciones
id	entero	-

### 2. Argumentos de salida:

**Opción 1:** Solicitud correcta

**Http Status:** HTTP 200 OK

**Body:**

```
{
  "Historial": [
    {
      "id": 2,
      "partidoIdpartido": {
        "id": 2,
        "equipoA": {
          "id": 2,
          "nombreEquipo": "Hormigon Armado",
          "colorEquipo": "Plomo",
          "mascota": "Hormiga"
        },
        "equipoB": {
          "id": 3,
          "nombreEquipo": "Naranja Mecánica",
          "colorEquipo": "Naranja",
          "mascota": "Naranjito"
        },
        "scoreA": 12,
        "scoreB": 10
      },
      "deporteIddeporte": {
        "id": 2,
        "nombreDeporte": "Ping Pong",
        "pesoDeporte": 4
      }
    }
  ]
}
```

```
        "horaFecha": "2023-10-23T01:31:41Z"
    }
}
}
```

**Opción 2:** Error en los parámetros de la solicitud  
**Http Status:** HTTP 400 Bad Request  
**Body:**

```
{
  "error": "No se encontro el equipo indicado"
}
```

### 3. Datos argumentos de salida:

Campo	Tipo de Dato	Observaciones
id	entero	-
partidoIdpartido	json	Null, si no hay partido registrado
partidoIdpartido.id	entero	-
partidoIdpartido.id.equipoA	json	Null, si no hay un equipob registrado
partidoIdpartido.id.equipoA.id	entero	-
partidoIdpartido.id.equipoA.nombreEquipo	cadena	-
partidoIdpartido.id.equipoA.colorEquipo	cadena	-
partidoIdpartido.id.equipoA.mascota	cadena	-
partidoIdpartido.id.equipoB	json	Null, si no hay un equipoB registrado
partidoIdpartido.id.equipoB.id	entero	-
partidoIdpartido.id.equipoB.nombreEquipo	cadena	-
partidoIdpartido.id.equipoB.colorEquipo	cadena	-
partidoIdpartido.id.equipoB.mascota	cadena	-
partidoIdpartido.scoreA	Entero	-
partidoIdpartido.scoreB	entero	-

deporteIddeporte	json	Null, si no hay deporte registrado
deporteIddeporte.id	entero	-
deporteIddeporte.nombreDeporte	cadena	-
deporteIddeporte.pesoDeporte	entero	-
horaFecha	date	-

#### 4. Ejemplos

Solicitud correcta:

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

- URL:** `http://localhost:8080/sdci/partido/gethistorialpartidos?idequipo=3`
- Method:** GET
- Query Params:** `idequipo=3`
- Status:** 200 OK, Time: 21 ms, Size: 541 B
- Response Body (JSON):**

```
{
  "Historial": [
    {
      "id": 2,
      "partidoIdpartido": {
        "id": 2,
        "equipoA": {
          "id": 2,
          "nombreEquipo": "Horzigon Azmado",
          "colorEquipo": "Plomo",
          "mascota": "Horziga"
        },
        "equipoB": {
          "id": 3,
          "nombreEquipo": "Naranja Mecánica",
          "colorEquipo": "Naranja",
          "mascota": "Naranjito"
        }
      }
    },
    {
      "id": 3,
      "partidoIdpartido": {
        "id": 3,
        "equipoA": {
          "id": 3,
          "nombreEquipo": "Naranja Mecánica",
          "colorEquipo": "Naranja",
          "mascota": "Naranjito"
        },
        "equipoB": {
          "id": 2,
          "nombreEquipo": "Horzigon Azmado",
          "colorEquipo": "Plomo",
          "mascota": "Horziga"
        }
      }
    }
  ]
}
```

localhost:8080/sdci/partido/gethistorialpartidos

Save

GEThttp://localhost:8080/sdci/partido/gethistorialpartidos?idequipo=3

ParamsAuthorizationHeaders (8)Body●Pre-request ScriptTestsSettings

Query Params

	Key	Value	Description
<input checked="" type="checkbox"/>	idequipo	3	

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 21 msSize: 541 B

PrettyRawPreviewVisualizeJSON

```
10  "id": 3,
11  "nombreEquipo": "Naranja Mecánica",
12  "colorEquipo": "Naranja",
13  "mascota": "Naranjito",
14  "scoreA": 12,
15  "scoreB": 10,
16  "deporteIddeporte": {
17    "id": 2,
18    "nombreDeporte": "Ping Pong",
19    "pesoDeporte": 4,
20  },
21  "horaFecha": "2023-10-23T01:31:41Z"
22 }
23 ]
24 }
```

## Solicitud incorrecta:

localhost:8080/sdci/partido/gethistorialpartidos

Save

GEThttp://localhost:8080/sdci/partido/gethistorialpartidos?idequipo=100

ParamsAuthorizationHeaders (8)Body●Pre-request ScriptTestsSettings

Query Params

	Key	Value	Description
<input checked="" type="checkbox"/>	idequipo	100	

BodyCookiesHeaders (4)Test Results

Status: 400 Bad RequestTime: 17 msSize: 189 B

PrettyRawPreviewVisualizeJSON

```
1  "error": "No se encontro el equipo indicado"
2
3  }
```