

SÉPTIMO LABORATORIO
CURSO: GESTION DE SERVICIOS DE TICS [TEL137]
SEMESTRE: 2023-II

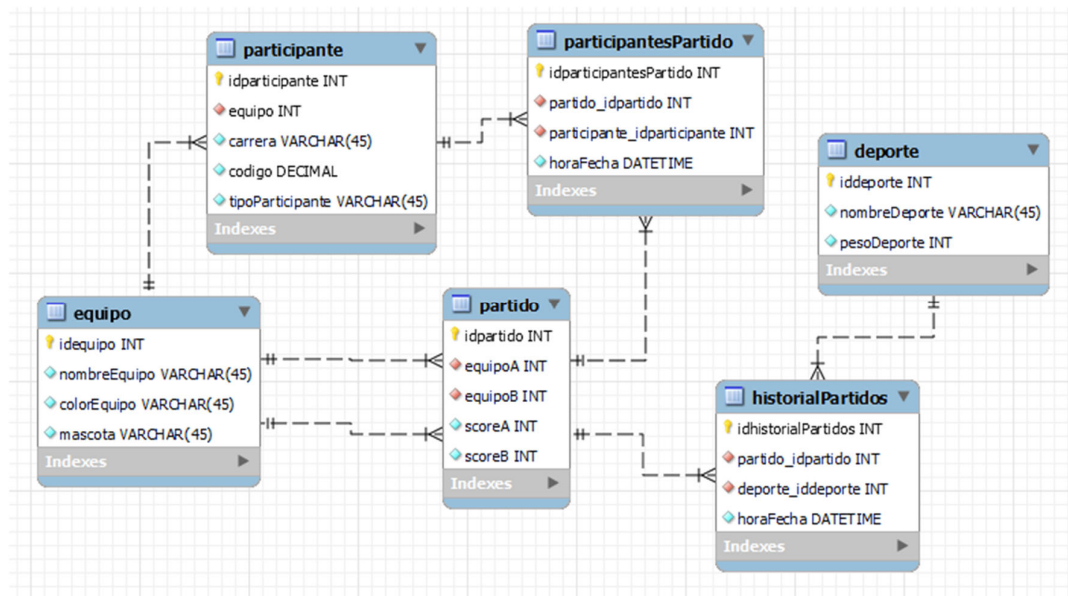
Tema: Creación de Web Services REST con Spring boot

- Duración: 1 hora y 30 minutos
- El laboratorio se realizará en parejas.

Recomendación: “Leer todo el documento antes de empezar”

Enunciado:

En la facultad de Ciencias e Ingeniería, desde hace más de 40 años, se celebra un torneo deportivo llamado Semana de Ciencias e Ingeniería (SDCI). El centro estudiantil de dicha facultad necesita crear un Web Service que les ayude a registrar el resultado de las actividades deportivas colectivas que hay en dicho evento. En ese sentido tenemos el siguiente modelo de DB tentativo:



Por ello, se les pide generar el Web Service del tipo REST que permita al centro estudiantil poder registrar los resultados de los partidos de este evento.

OBS: El entregable de este laboratorio deberá incluir, aparte del código en GitHub, la documentación reglamentaria del Web Service generado. DEBE INCLUIR capturas de pantalla de la aplicación POSTMAN (se debe poder apreciar **la ventana completa** de la aplicación en donde se aprecien los resultados obtenidos después de haber ejecutado el request correspondiente).



PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA
INGENIERÍA DE LAS TELECOMUNICACIONES

Ejemplos de capturas válidas:

- Se debe visualizar claramente el endpoint usado y la respuesta obtenida por parte del servidor.

lab7_unica / guardar-solicitudes

POST localhost:8080/solicitudes/registro

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1 {
2   "id": "3",
3   "solicitud_producto": "credito_especial",
4   "solicitud_monto": "100",
5   "solicitud_fecha": "2022-10-25",
6   "usuarios_id": {
7     "id": 1,
8     "nombre": "Lewis",
9     "apellido": "Rengifo",
10    "correo": "l.rengifo@pucp.edu.pe",
11    "username": "lrengifo",
12    "password": "root",
13    "estado_logico": "activo",
14    "rol_id": {
15      "id": 1,
16      "nombre_rol": "socio"
17    },
18    "fecha_registro": "2022-10-25T05:00:00.000+00:00"
19  },
20   "solicitud_estado": ""
21 }
```

Body Cookies Headers (5) Test Results

Status: 201 Created Time: 82 ms Size: 206 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "Monto solicitado": "100.0",
3   "id": "3"
4 }
```

- En caso de enviar algún campo vía el menú de parámetros, se debe poder apreciar los parámetros enviados, a parte de la respuesta por parte del servidor.

lab7_unica / aprobar-solicitud

PUT localhost:8080/solicitudes/aprobarSolicitud?idSolicitud=3

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

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> idSolicitud	3	
Key	Value	Description

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 191 ms Size: 184 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "id solicitud": "3"
3 }
```

Ejercicio 1:

Se necesita poder registrar lo siguiente:

- a) Equipos participantes de la SDCl. **(2 puntos)**
 - `localhost:8080/sdci/equipo/registro` [POST]
- b) Participantes de la SDCl. **(2 puntos)**
 - `localhost:8080/sdci/participante/registro` [POST]
- c) Deportes colectivos de la SDCl. **(2 puntos)**
 - `localhost:8080/sdci/deporte/registro` [POST]

Para ello, se necesita crear los endpoints correspondientes en el Web Service para poder llenar las tablas referenciadas.

El server debe responder con un 200 OK en caso de creación exitosa. De lo contrario, responder con el HTTP Response correspondiente.

Ejercicio 2:

Registro de partidos:

- a) Se necesita poder registrar los partidos ocurridos durante la SDCl. Al registrar un partido se debe añadir, también, un registro a la tabla de historial de partidos.
 - `localhost:8080/sdci/partido/registro` [POST] **(2 puntos)**
- b) Se necesita poder obtener la lista de participantes de un partido y se debe poder obtener dicha lista filtrada si es que indica el equipo deseado (el equipo debe ser uno de los 2 equipos involucrados en el partido en cuestión)
 - `localhost:8080/sdci/partido/getparticipantes` [GET] **(2 puntos)**
 - `localhost:8080/sdci/partido/getparticipantes?idequipo=2` [GET] **(5 puntos)**
- c) Se necesita poder obtener la lista del historial de partidos jugados durante SDCl
 - `localhost:8080/sdci/partido/gethistorialpartidos` [GET] **(2 puntos)**

- `localhost:8080/sdci/partido/gethistorialpartidos?=idequi
po=3` [GET] (3 puntos)

Para ello, se necesita crear los endpoints correspondientes en el Web Service para poder llenar las tablas referenciadas.

El server debe responder con un 200 OK en caso de creación exitosa. De lo contrario, responder con el HTTP Response correspondiente.

OBS: La estructura de la base de datos puede ser modificada en caso de ser absolutamente necesario. Si lo hace, necesita sustentar en la documentación entregada la modificación realizada.