

Programación Orientada a Objetos

2025

Práctica 3

HTTP

- ¿Qué es el protocolo HTTP?
- Mencionar los distintos métodos y verbos de HTTP
- ¿Los request de tipo POST tienen un body? ¿Y los de tipo GET?
- Códigos de estado de response HTTP: Describa los códigos: 200, 201, 400, 401, 403 y 500.
- ¿Qué posibles valores de Media Type puede tomar el header Content Type de HTTP?

REST

- ¿Qué es REST?
- Explique brevemente los 6 principios de REST
- ¿Qué es un Resource?
- En base a los lineamientos sugeridos, que verbos de HTTP se deben utilizar para cada tipo de llamada al servidor para implementar la operaciones CRUD (create, retrieve, update, delete)
- Describa brevemente REST resource naming strategy
- ¿cuál la diferencia entre REST y RESTful?

Spring.io

- RESTful Web Services
 - ¿Con qué anotaciones se declaran los distintos métodos HTTP?
 - Describa el propósito de las anotaciones
 - `@RestController`
 - `@RequestMapping`
 - `@PathVariable`
 - `@RequestParam`
 - `@RequestBody`
 - `@ResponseStatus`

- `@RequestHeader`
- Explique el uso de la clase `ResponseEntity`
- Spring Web:
 - ¿Cuál es el propósito de la anotación `@Service`?
 - Para que se usa la anotación `@Autowired`

JSON

- Defina brevemente **JavaScript Object Notation**

DTO

- Explique brevemente el patrón de diseño **Data Transfer Object**.
- Investigue el uso de la librería ModelMapper
<http://modelmapper.org/getting-started/>

Proyecto a implementar: Sistema de Inscripción a Torneos Deportivos

Sobre el proyecto creado en el trabajo práctico anterior, vamos a comenzar a implementar el backend de un sistema que permita realizar la inscripción a las distintas competencias en torneos mediante un web service Restful que implementa la arquitectura Rest.

El sistema tendrá dos tipos de usuarios: Administrador (puede crear torneos y competencias) y Participante (puede registrarse en la plataforma e inscribirse en competencias de un torneo).

Cada competencia tiene un precio base (por ejemplo \$20.000). Si un participante se inscribe en más de una competencia dentro del mismo torneo, la primera se paga completa y las siguientes al 50%.

Del participante el sistema llevará registro de su nombre, apellido, tipo y número de documento (no se pueden repetir). El participante deberá registrarse en el sistema para poder inscribirse en las distintas competencias del torneo, para eso además de los datos personales el sistema registrará como datos para su autenticación una dirección email (que deberá ser única el sistema) y la contraseña.

Del usuario administrador el sistema registrará un email y contraseña. Un usuario no puede registrarse como administrador. Un usuario administrador sólo se puede crear por otro administrador desde el backoffice del mismo sistema. Además un administrador podrá crear torneos y competencias.

Del torneo se registrará el nombre, una descripción y las fechas de inicio y fin del mismo. Para que los distintos participantes puedan inscribirse en las competencias el torneo debe

ser publicado. Los usuarios administradores podrán editar los datos de un torneo o eliminar un torneo mientras el mismo no se encuentre publicado.

De las competencias el sistema registrará el nombre, el precio y cupo (es decir, la cantidad de participantes que pueden inscribirse). Los usuarios administradores podrán agregar competencias al torneo mientras el mismo no se encuentre publicado, al igual que editar o eliminar.

Los participantes se podrán inscribir una única vez por competencia siempre que el torneo de la misma esté publicado, la fecha en la que se inscriba sea anterior a la fecha de inicio del torneo y la competencia no haya alcanzado el límite del cupo. Una vez inscripto el participante no podrá eliminar su suscripción.

La inscripción es la entidad que relaciona al participante con la competencia, además registrará el precio de esa inscripción (tener en cuenta el descuento que puede aplicarse sobre el precio base de la competencia) y la fecha en la que el participante se inscribió.

Desde el backoffice del sistema los usuarios administradores deberán poder acceder a un listado de participantes inscriptos por competencia y el monto total recaudado por el sistema.

Finalmente los participantes podrán acceder al listado de todos los torneos que participaron y al listado de competencias dentro del mismo. El listado de torneos deberá estar ordenado de manera cronológica descendente.

Se pide:

- Diseñar el modelo de clases (UML): Torneo, Competencia, Participante, Inscripción
- Implementar las clases (identificando atributos y sus tipos y las relaciones entre las mismas) en java dentro del proyecto creado en el trabajo práctico anterior. Las clases deberán estar dentro del package ar.edu.unnoba.poo2025.torneos.model.
- Crear el modelo relacional de la base de datos (DER).
- Respetar las convenciones de nombrado propias de cada uno de los puntos anteriores.

Entrega

La entrega se realiza sobre la tarea creada en el curso de la plataforma de educación digital. No se puede entregar archivo con url al repositorio git. La entrega debe incluir el código fuente del proyecto. Además agregar al mismo una carpeta que contenga el diagrama UML y otra carpeta que contenga el script para poder crear el esquema de la base de datos.