

Trabajo Práctico Nro. 1 APIs con SpringBoot

Ejercicio N° 1 – Conversión de Unidades

Un turista norteamericano decidió recorrer en auto diferentes países de América Latina. A la hora de cargar combustible, para su sorpresa, se encontró con que en la mayor parte de los países no se utiliza la medida “galones” sino que se utiliza “litros” a la hora de reabastecerse.

Para poder solicitar correctamente la cantidad de litros que necesita, requiere el desarrollo de una API que, al enviarle la cantidad de galones otorgue como respuesta el equivalente en litros.

Para el desarrollo tener en cuenta que: 1 galón es equivalente a 3,78541 litros.

Ejercicio N° 2 – Promedio de Notas

Un alumno de secundaria desea poder contar con una API mediante la cual, al enviar las 3 notas de una materia obtenga como respuesta el promedio de la misma. Para el pasaje de notas como parámetro utilizar la annotation `@RequestParam`.

Ejercicio N° 3 – Restaurante

Un restaurante posee un total de 5 platos en su menú. Cada uno de ellos tiene un número de plato, nombre, precio y breve descripción.

El restaurante desea una aplicación que permita que los clientes sean capaces consultar los datos de los platos del menú a partir de proporcionar el número de plato. Para ello, se necesita el desarrollo de una API que a partir de la recepción del núm. de plato en una solicitud GET, devuelva el plato correspondiente con todos sus datos.

Nota: Tener en cuenta que, como aún no se está trabajando con base de datos, los platos deben ser almacenados en una **collection**, por ejemplo una **ArrayList** o una **LinkedList**.

Ejercicio N° 4 – Odontólogo

Un odontólogo cuenta con una lista de pacientes. De cada uno de ellos cuenta con los datos: id, DNI, nombre, apellido y fecha de nacimiento.

- a. El odontólogo necesita de una API que sea capaz de brindarle el listado completo de sus pacientes. Para esto, desarrollar un end-point que permita devolver la lista completa de pacientes mediante el método GET.
- b. Además, el odontólogo necesita saber sus pacientes que son menores de edad para contemplar que los mismos asistan a sus citas acompañados por un mayor. Para ello, necesita que la API posea un end-point que al realizar la solicitud GET, devuelva únicamente a los pacientes que sean menores de edad. *Para realizar el cálculo de edad tener en cuenta la fecha de nacimiento de cada paciente y la fecha «actual» (día en que se esté haciendo la consulta).*

Tips para fechas

- 1- Utilizar el tipo de dato **LocalDate**.
- 2- Para obtener la fecha de hoy se utiliza el método **LocalDate.now()**
- 3- Para calcular períodos entre dos fechas se utiliza la clase **Period** y el método **between**. Por ejemplo: **Period.between (fecha_hoy, otra_fecha)**
- 4- Si utilizamos el método **getYears** en el objeto **Period** que tengamos, obtendremos la cantidad de años de diferencia.

Ejercicio N° 5 – Estaturas en Básquet

Un director técnico de un equipo de básquet desea poder contar con una API que, al enviar los datos de 5 de sus jugadores en el body de una solicitud http POST, los mismos sean dados de alta en una bases de datos lógica representada por una lista de tipo ArrayList y que, al mismo tiempo, luego del alta correspondiente, se pueda obtener inmediatamente como respuesta el promedio de estatura de todos sus jugadores ingresados.

Los datos que se ingresan para cada jugador son los siguientes: id, dni, nombre, apellido, edad, peso y estatura.

Tener en cuenta para la realización de este ejercicio la manipulación/creación de archivos JSON, el uso de Postman como herramienta para generar la solicitud POST, la annotation @RequestBody y el manejo de ArrayLists.

Para todos los ejercicios, generar y adjuntar la coleccion.json