

TPE - Parte 3

Consigna

Para la tercera entrega, se debe continuar la idea del trabajo original. El objetivo es agregar una API REST pública para brindar integración con otros sistemas. Esta API se debe entregar en un repositorio nuevo. Lo único que comparten con el TP anterior es la base de datos.

Los servicios brindados por la API deben cumplir unos **requerimientos mínimos*** pero idealmente se debe pensar alguna idea original que se adapte a su contexto.

Ejemplos:

Comercial (Productos y Categorías)

API Rest para listar todos los productos en oferta filtrando por categoría. De esta manera otro sitio de terceros podría mostrar una publicidad con los productos en oferta.

Noticias (Noticias y Sección)

API Rest para listar noticias y realizar comentarios anónimos en las noticias. De esta manera se podría generar una app mobile para leer y comentar noticias.

***Requerimientos Funcionales Mínimos**

1. La API Rest debe ser RESTful
2. Debe tener al menos un servicio que liste (GET) una colección entera de entidades.
3. El servicio que lista una colección entera debe poder ordenarse opcionalmente por al menos un campo de la tabla, de manera *ascendente* o *descendente*.
4. Debe tener al menos un servicio que obtenga (GET) una entidad determinada por su ID.
5. Debe tener al menos un servicio para agregar y otro para modificar datos (POST y PUT)
6. La API Rest debe manejar de manera adecuada al menos los siguientes códigos de error (200, 201, 400 y 404)

Requerimientos Funcionales Optativos

Los opcionales suman un **punto cada uno**. Se debe completar al **menos uno para acceder a la promoción ya que el resto del trabajo sin opcionales suma 6 puntos**.

7. El servicio que obtiene una colección entera debe poder paginar.
8. El servicio que obtiene una colección entera debe poder filtrarse por alguno de sus campos.
9. El servicio que obtiene una colección entera debe poder ordenarse por cualquiera de los campos de la tabla de manera ascendente o descendente. (A diferencia del obligatorio que es solo por un campo a elección).
10. El servicio debe requerir un token para realizar modificaciones (PUT, POST)



Requerimientos NO Funcionales OBLIGATORIOS

- Se debe adjuntar documentación de los endpoints generados en el README.md del repositorio. Es decir, una descripción de cada endpoint, cómo se usan y ejemplos. Entender que esta documentación la va a leer otro desarrollador para entender cómo se consume la API.

Aclaraciones

Respecto al modelo de datos: Se puede (y se alienta) agregar columnas/tablas para que el servicio que agreguen tenga más sentido en su contexto. Ejemplo: Agregar una tabla “reseñas” para almacenar las reseñas via API REST de una película.

NOTA: La base de datos es compartida por los dos proyectos. Tener cuidado de no “romper” la primera entrega con los cambios en la DB de la segunda.

Generales:

- La API REST se debe crear en otro repositorio distinto al trabajo original.
- La base de datos es la misma para los dos (incluirla de todos modos en el nuevo repositorio).
- No es necesario crear un frontend que consuma la API. Queda a total elección del alumno/a si quiere agregar un frontend. Todas las pruebas se harán utilizando POSTMAN (o similar)



Criterios de corrección

Se evaluará la correcta división de responsabilidades en las clases, no repetición de código, identificadores (nombres de clases, variables, etc) descriptivos, etc.

Los trabajos deben implementar la totalidad de la funcionalidad (ambas entregas) funcionando correctamente.

El trabajo debe realizarse en equipos de 2, donde a cada miembro se le evaluarán distintas responsabilidades de la siguiente forma:

Miembro A:

- Listado ordenado
- PUT
- Ordenado por cualquier campo (opcional)

Miembro B:

- Obtener un elemento por ID
- POST
- Filtado (opcional)

Opcionales que suman a ambos:

- Paginado
- Autenticación Token

Entrega



LUNES 13 DE NOVIEMBRE



La entrega se realizará en un repositorio público de GIT a elección. La URL debe estar subida a la planilla de trabajos en la fecha de entrega pautada