

Universidad ORT Uruguay

Facultad de Ingeniería

Obligatorio 1

Diseño de Aplicaciones 2

Repositorio: <https://github.com/IngSoft-DA2-2023-2/242493-260956-281651>

Video detallando pruebas: <https://youtu.be/Mo39Tbi6L1s>

Romina Arour - 242493

Daiana Aysa - 281651

Ana Gutman - 260956

Tutores:

2023

Criterios Seguidos para cumplir con REST

Para garantizar la conformidad de nuestra API con los principios REST, se implementaron diversas medidas. En primer lugar, nos aseguramos de que la parte del servidor siguiera la arquitectura cliente-servidor, permitiendo así la escalabilidad de la plataforma sin depender de la interfaz de usuario.

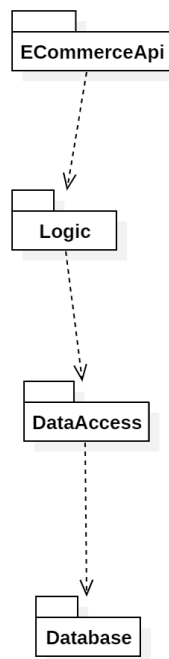
Además, nos cercioramos de que nuestra API fuera "stateless", lo que implica que cada solicitud debe incluir toda la información necesaria para llevar a cabo la operación deseada, prescindiendo de cualquier información almacenada en el servidor.

Asimismo, adoptamos una estructura apropiada para las URL de los recursos, haciendo uso de nombres en plural y la inclusión de identificadores únicos para cada recurso.

Para asegurar una Interfaz Uniforme, definimos diversas URIs para manejar las entidades del dominio utilizando los verbos HTTP como métodos, tales como GET, PUT, POST y DELETE, garantizando su uso coherente en toda la API.

Finalmente, en nuestro proyecto, implementamos un sistema de capas para la API, donde una interfaz lógica llama a la interfaz de base de datos sin depender directamente de la API en cuestión. Siguiendo estos criterios, logramos asegurar que nuestra API cumpliera de manera consistente y efectiva con los principios REST, contribuyendo así a una solución robusta y escalable.

Adjuntamos una imagen que ilustra nuestro sistema de capas.



Mecanismo de autenticación de requests

Para cumplir con REST, gestionamos tres tipos de usuarios, cada uno con diferentes niveles de permisos para acceder a recursos específicos. Nuestra estrategia se basa en el uso de tokens incluidos en los encabezados de las solicitudes que requieren autorización especial. Los usuarios son responsables de mantener sus tokens, ya que su pérdida implica la creación de uno nuevo. Cuando una solicitud llega a nuestra API, validamos el token asociado para determinar si el usuario tiene los permisos necesarios para acceder al recurso solicitado. Esto asegura un control preciso y seguro sobre quién puede acceder a qué recursos en nuestra aplicación, cumpliendo así con los principios REST.


Códigos de Error

La siguiente tabla muestra los códigos de error HTTP que utilizamos junto con sus breves descripciones.

200	La operación se realizó correctamente
201	El elemento fue creado correctamente
400	La request tiene parámetros erróneos
401	No se tienen permisos
403	Se tienen permisos pero no para el recurso
409	La contraseña no válida
500	Error del servidor

Descripción de los Resources de la API

El detalle de todos los endpoints creados se pueden ver en la siguiente tabla:

 Endpoints