

Desarrollo de Web Api's con .NET

Repaso + trabajo integrador I



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
ENCUENTRO REGIONAL BUENOS AIRES

**Centro de
e-Learning**



¿En qué
consiste la
arquitectura
MVC?

Arquitectura **Model-View-Controller**

Se utiliza para desacoplar la interfaz de usuario (vista), los datos (modelo) y la lógica de la aplicación (controlador).

Separa una aplicación en tres grupos de componentes principales: **modelos**, **vistas** y **controladores**.

Permite lograr la separación de intereses.



¿Qué es una
API?

Application Programming Interface

Es una interfaz que permite la comunicación entre 2 aplicaciones.

Podemos decir que es un sitio que en vez de responder algo visual, como HTML con CSS, nos responde información.

Los usuarios probablemente no van a ingresar a este tipo de sitios, pero sí una aplicación que necesite abastecerse de cierta información.

Por ejemplo, una aplicación que necesite un listado completo de todos los países del mundo podría consumir una API que le otorgue esa información.



¿Y una API
REST?

REpresentational State Transfer

Es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP.

Proporciona un fuerte aislamiento con quien la consume.

Los objetos REST son manipulados a través de una URI (Uniform Resource Identifier).

Esta URI (endpoint) hace de identificador único de cada recurso del sistema REST, por lo que no puede ser compartida por más de un recurso.

Representational State Transfer

La estructura básica de una URI es la siguiente:

{protocolo}://{hostname}:{puerto}/{ruta del recurso}?{parámetros de filtrado (opcional)}

El nombre de la URI no debe contener palabras que impliquen acciones, por lo que deben evitarse los verbos en su construcción.

Además, las URI siguen una jerarquía lógica de capas que permite ordenar los recursos y englobar las distintas funcionalidades entre sí.

Métodos HTTP

Los métodos son usados para manipular los diferentes recursos que conforman una API.

Los principales métodos soportados por HTTP son:

- POST: crear un recurso nuevo.
- PUT: modificar un recurso existente.
- GET: consultar información de un recurso.
- DELETE: eliminar un recurso determinado.
- PATCH: modificar solamente un atributo de un recurso.



¿Qué es una
prueba
unitaria?

Pruebas unitarias

Una prueba unitaria es una prueba que ejercita componentes o métodos de software individuales, también conocidos como "unidad de trabajo".

Estas solo deberían probar código que pueda controlar el desarrollador. No se usan para comprobar problemas con la infraestructura.

Estos problemas incluyen la interacción con bases de datos, sistemas de archivos y recursos de red.



¿Porqué
hacer
pruebas
unitaria?

¿Porqué hacer pruebas?

- Protección frente a regresión
- Documentación ejecutable
- Calidad de código



¿Cuáles son
los posibles
resultados
de una
prueba?

¿Porqué hacer pruebas?

- Verde
- Falso negativo
- Rojo
- Falso positivo



¿En qué se
diferencian
las pruebas
de caja
negra y caja
blanca?

Pruebas de caja negra vs caja blanca

Las de **caja negra** verifican si se obtiene la salida deseada a una entrada dada.

Es el método en el cual el elemento es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno.

Las de **caja blanca** verifican cómo el sistema se ejecuta internamente, cómo está construido y sus soluciones estructurales garantizando que todos los **caminos** independientes dentro de un componente se han ejercitado al menos una vez.



¿Qué son las
clases de
equivalencia?

Clases de equivalencia

La división en particiones de clases de equivalencia es una técnica de pruebas de **caja negra** en la que las entradas del programa se clasifican en clases.

Considera como perteneciente al dominio de entrada a todas las entradas que puedan tener efecto en la salida de la prueba, positivo o negativo.

Cada clase se llama clase de equivalencia porque todos elementos en la clase prueban la misma cosa.

Si uno de los elementos de una clase provoca un error, todos los de esa clase probablemente lo harán.

La idea principal de esta técnica es identificar los casos de prueba usando un elemento de cada clase de equivalencia.

¡¡A trabajar!!

