PREGRADO

UNIDAD 1 | OVERVIEW

ASP.NET CORE & WEB SERVICES

SI730 | Aplicaciones Web

Al finalizar la unidad, el estudiante desarrolla y comunica una solución de web services aplicando los principios RESTful para una Arquitectura Orientada a Servicios en un ambiente de desarrollo ágil.

AGENDA

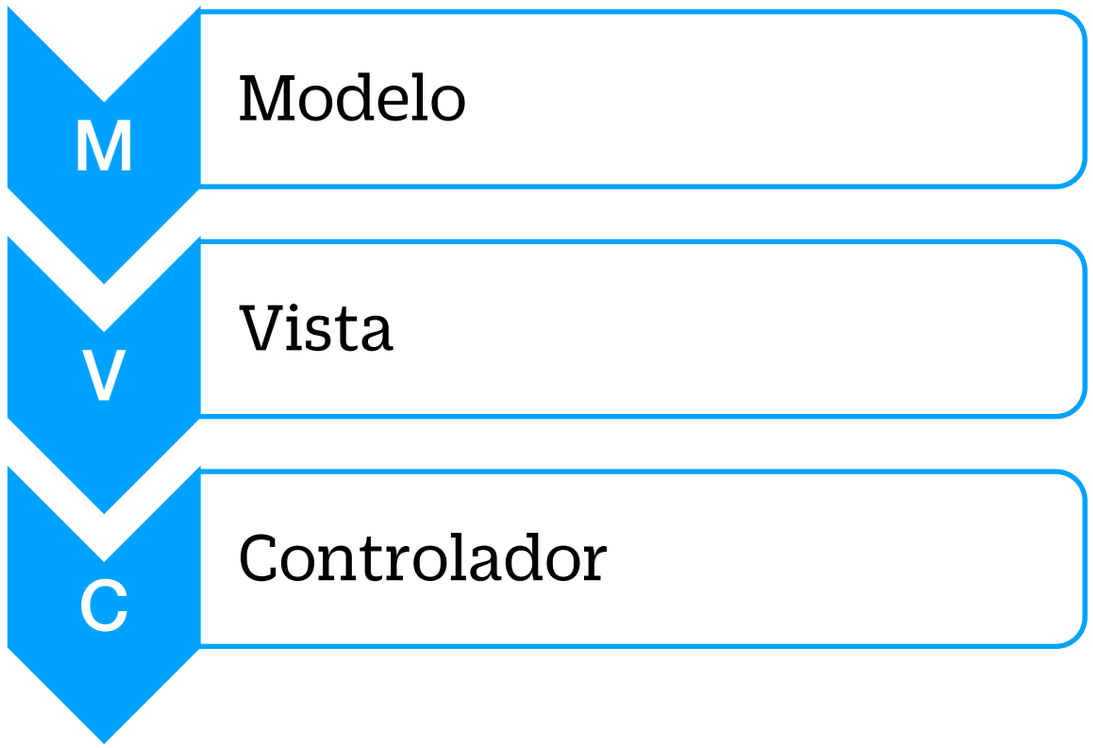
MVC PATTERN

ASP.NET CORE & MVC

DEPENDENCY INJECTION

RESTFUL WEB SERVICES

Definición



Es un patrón de desarrollo de software.

Un patrón de desarrollo de software es una forma reutilizable de resolver un problema común. Nos permiten desarrollar aplicaciones de manera mucho más sencilla con estructuras probadas y que funcionan.

Definición

- MVC es patrón de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: Modelo, Vista y Controlador.

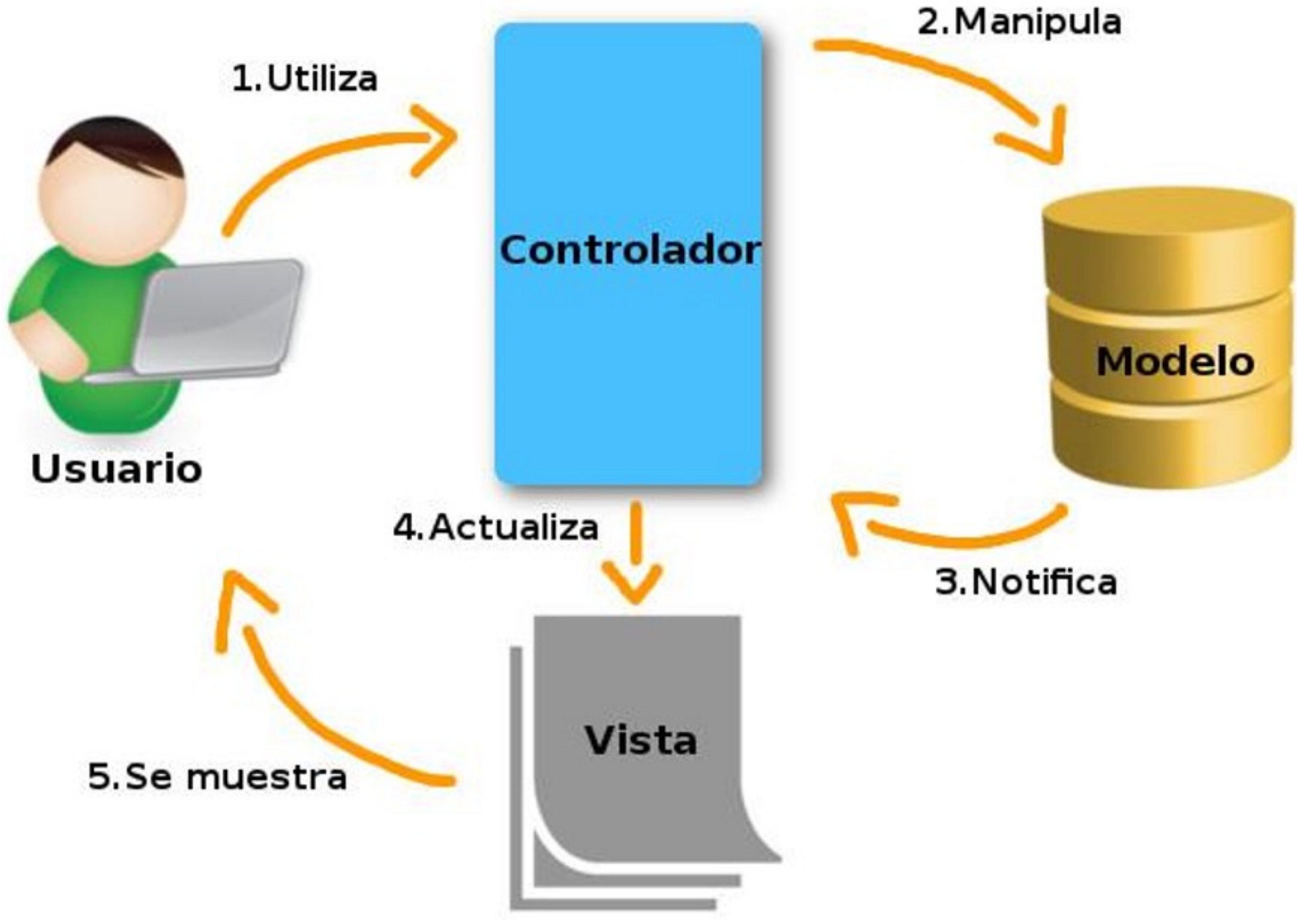
- Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

Definición

El Modelo contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

La Vista o interfaz de usuario, compone la información que se envía al cliente y los mecanismos interacción con éste.

El Controlador, es el cerebro y actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos.


1. Utiliza
2. Manipula
Usuario
Controlador
Modelo
4. Actualiza
3. Notifica
Vista
5. Se muestra

¿Por qué se utiliza MVC?

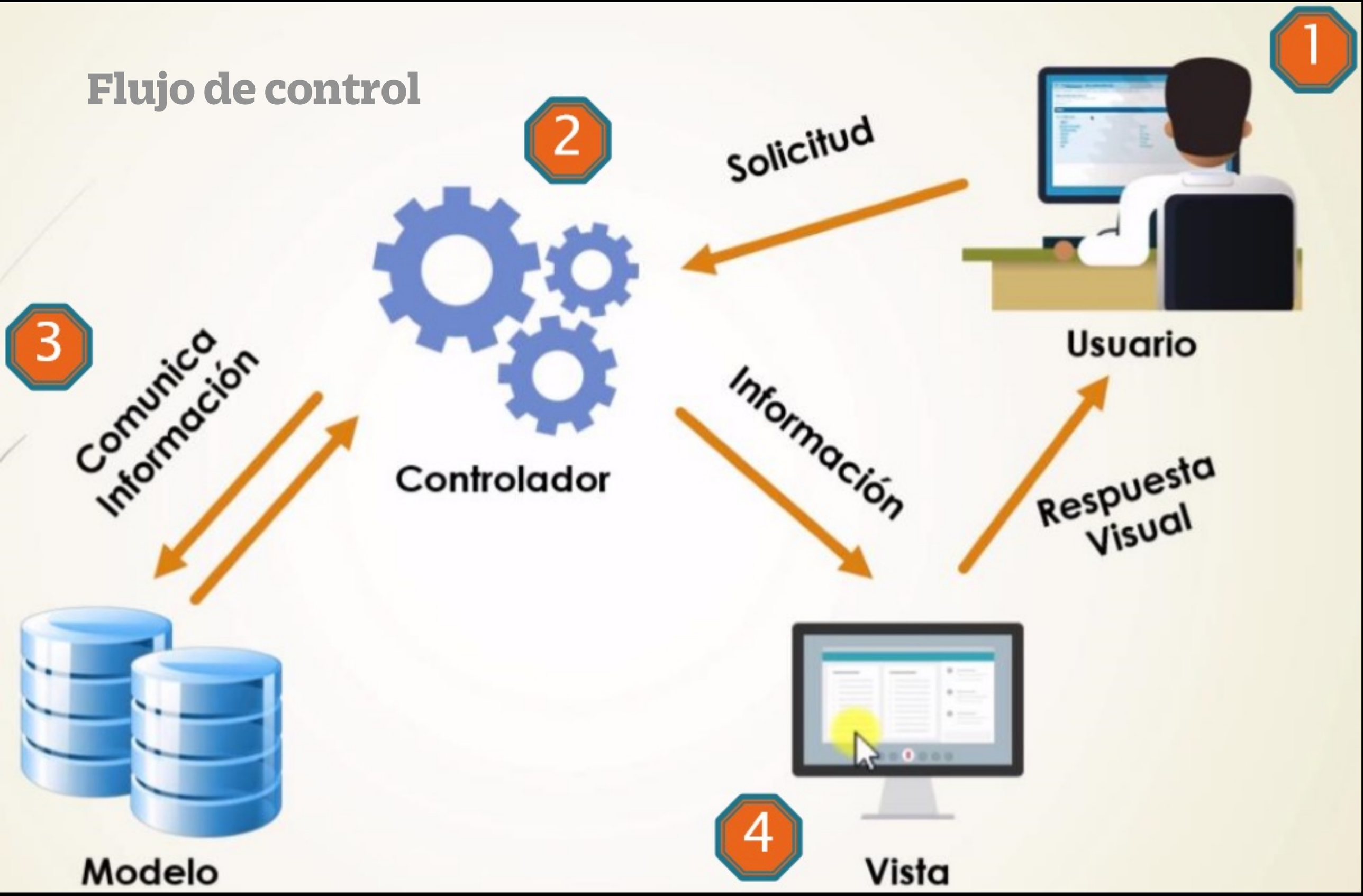
- Porque nos permite separar los componentes de nuestra aplicación dependiendo de la responsabilidad que tienen, esto significa que cuando hacemos un cambio en alguna parte de nuestro código, esto no afecte otra parte del mismo.
- Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la reutilización del código y la separación de conceptos.

Ejemplo

- Si modificamos nuestra BD, sólo deberíamos modificar el Modelo, que es quién se encarga de los datos y el resto de la aplicación debería permanecer intacta.

- Esto respeta el principio de responsabilidad única . Es decir, una parte del código no debe de saber qué es lo que hace toda la aplicación, sólo debe de tener una responsabilidad.

Flujo de control



Flujo de control

1. El usuario realiza una acción en la interfaz.

2. El controlador analiza el evento de entrada.

3. El controlador notifica al modelo la acción del usuario, lo que puede implicar un cambio del estado del modelo.

4. Se genera una nueva vista. La vista toma los datos del modelo a través del controlador.

   * *El modelo no tiene conocimiento directo de la vista.*

Funcionamiento del MVC en la web

- El usuario manda una petición a través del navegador.
- El controlador analiza la solicitud.
- El modelo, que se encarga de los datos de la app, consulta la base de datos y obtiene la información requerida.
- Luego, el modelo responde al controlador con los datos que solicitó.
- Finalmente, el controlador envía la información a la vista, pudiendo aplicar los estilos (CSS), organizar la información y construir la página que se visualiza en el navegador.

Ventajas del MVC

- Fácil organización del código en tres componentes diferentes.
- Crea independencia del funcionamiento.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de otras capas.
- Si trabaja con un equipo de programadores entonces les da una mayor facilidad para poder seguir el trabajo entre varios integrantes.
- Facilita el mantenimiento en caso de errores.
- Hacen que las aplicaciones sean fácilmente extensibles.
- Se adaptan a los frameworks de hoy en día.

Desventajas del MVC

- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.
- La curva de aprendizaje del patrón de diseño es más alta que usando otros modelos sencillos.

AGENDA

MVC PATTERN

ASP.NET CORE & MVC

DEPENDENCY INJECTION

RESTFUL WEB SERVICES

Introducción

Tecnología

Framework de trabajo

Concepto

- ASP.NET Core MVC es un framework de código abierto y multiplataforma para la creación de aplicaciones modernas conectadas a Internet, como aplicaciones web y web services.
- Se diseñó para proporcionar un framework de desarrollo optimizado para las aplicaciones que se implementan tanto en la nube como en servidores dedicados en las instalaciones del cliente.

Concepto

- ASP.NET Core MVC aprovecha el usa del patrón MVC, inyección de dependencias y una canalización de solicitudes formada por middleware.

- Es open source bajo la licencia Apache 2.0, lo que significa que está disponible de forma gratuita y la comunidad es animada para contribuir con corrección de errores y adición de nuevas características.

Ventajas

- Rapidez
- Ecosistema
- Seguridad

AGENDA

MVC PATTERN

ASP.NET CORE & MVC

DEPENDENCY INJECTION

RESTFUL WEB SERVICES

Concepto

- La inyección de dependencias (DI, por sus siglas en inglés) es un patrón usado en el diseño orientado a objetos que trata de solucionar las necesidades de creación de los objetos de una manera práctica, útil, escalable y con una alta versatilidad del código.

- Tiene como finalidad solucionar el problema de mantener los componentes o capas de una aplicación lo más desacopladas posible.

- Busca que sea más sencillo reemplazar la implementación de un componente por otro.

Concepto

- Para cumplir con dicho objetivo, DI nos permite inyectar comportamientos a componentes haciendo que nuestras piezas de software sean independientes y se comuniquen únicamente a través de una interface.

AGENDA

MVC PATTERN

ASP.NET CORE & MVC

DEPENDENCY INJECTION

RESTFUL WEB SERVICES

REST

REpresentational State Transfer

RESTful

Web Services basados en arquitectura REST.

Implementan interacción basada en métodos HTTP

Aplican URI (Uniform Resource Identifier)

Representación de información en formato JSON

RESTful API

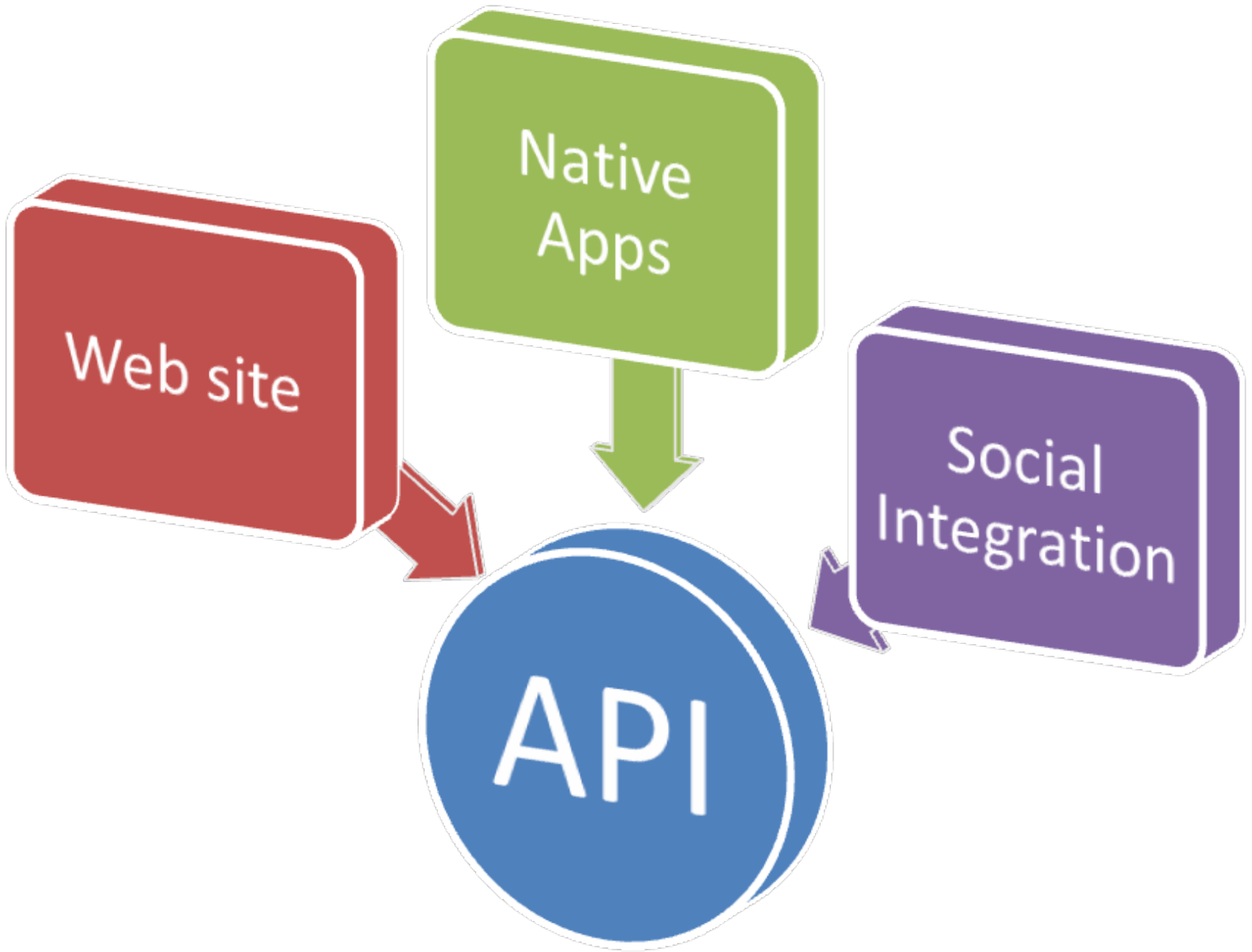
Application Programming Interface

Descompone una transacción en pequeños módulos

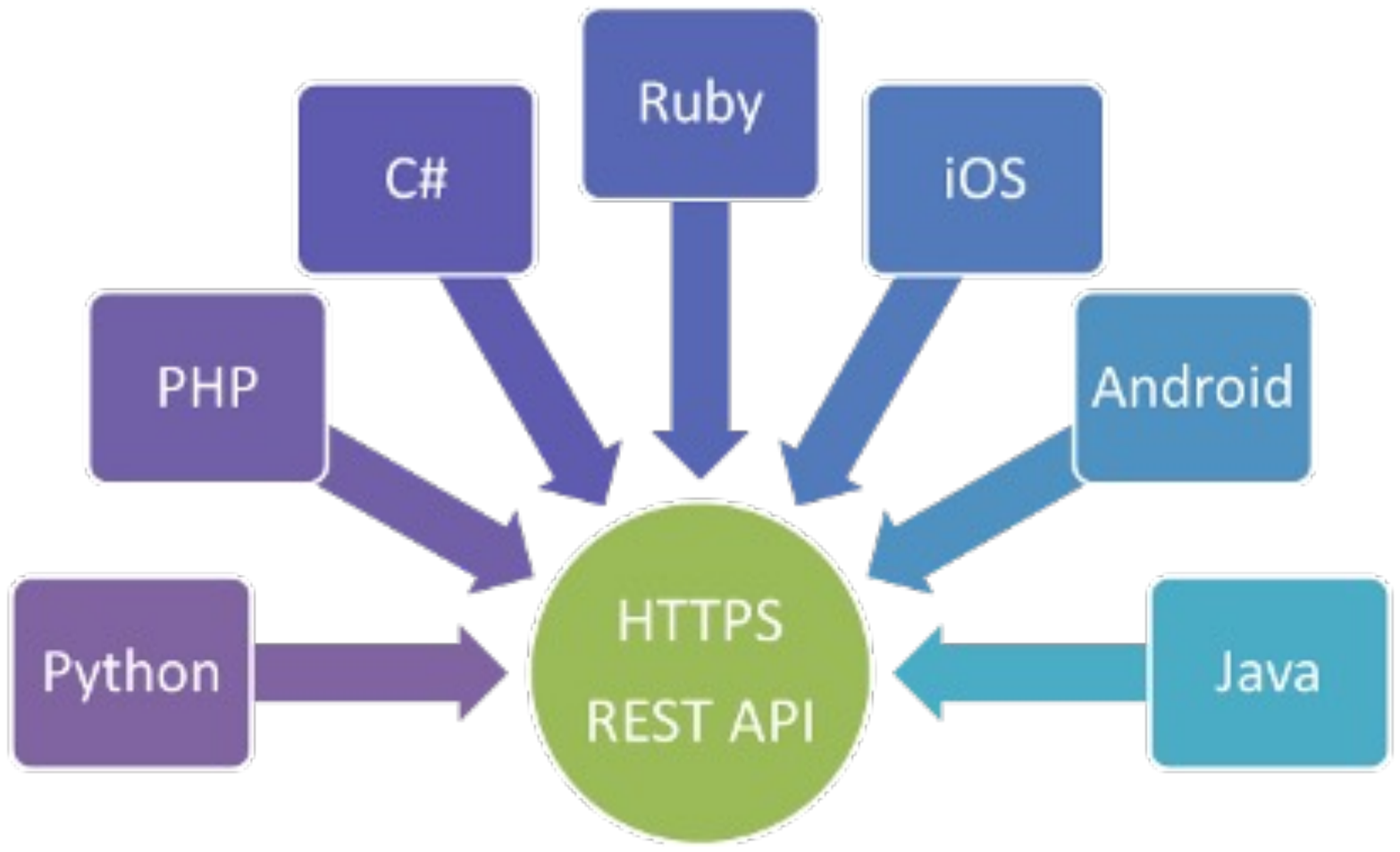
Cada módulo gestiona una parte de la transacción.

Aplica métodos basados en HTTP (RFC 2616)

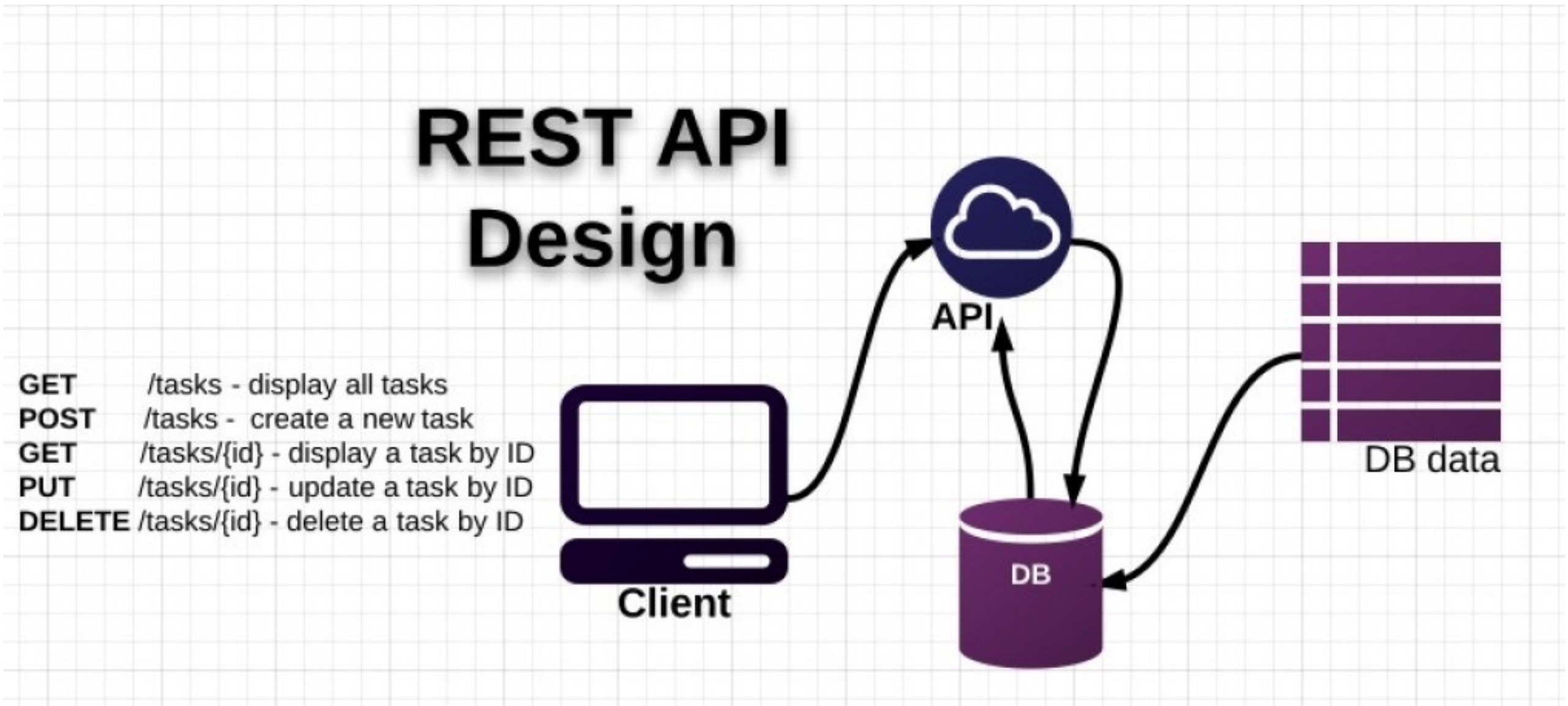
RESTful API



RESTful API



RESTful API


REST API
Design
GET /tasks - display all tasks
POST /tasks - create a new task
GET /tasks/{id} - display a task by ID
PUT /tasks/{id} - update a task by ID
DELETE /tasks/{id} - delete a task by ID
Client
API
DB
DB data

RESTful API

HTTP Methods

HTTP Verb	CRUD	Entire Collection (e.g. /customers)	Specific Item (e.g. /customers/{id})
POST	Create To create new subordinate resource	201 (Created), 'Location' header with link to /customers/{id} containing new ID.	404 (Not Found), 409 (Conflict) if resource already exists..
GET	Read To retrieve resource representation/information only	200 (OK), list of customers. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single customer. 404 (Not Found), if ID not found or invalid.
PUT	Update/Replace To update existing resource	405 (Method Not Allowed), unless you want to update/replace every resource in the entire collection.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
PATCH	Update/Modify To make partial update on a resource	405 (Method Not Allowed), unless you want to modify the collection itself.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
DELETE	Delete To delete resources	405 (Method Not Allowed), unless you want to delete the whole collection—not often desirable.	200 (OK). 404 (Not Found), if ID not found or invalid.

JSON

JavaScript Object Notation

JSON

Colección de pares nombre / valor

Lista ordenada de valores

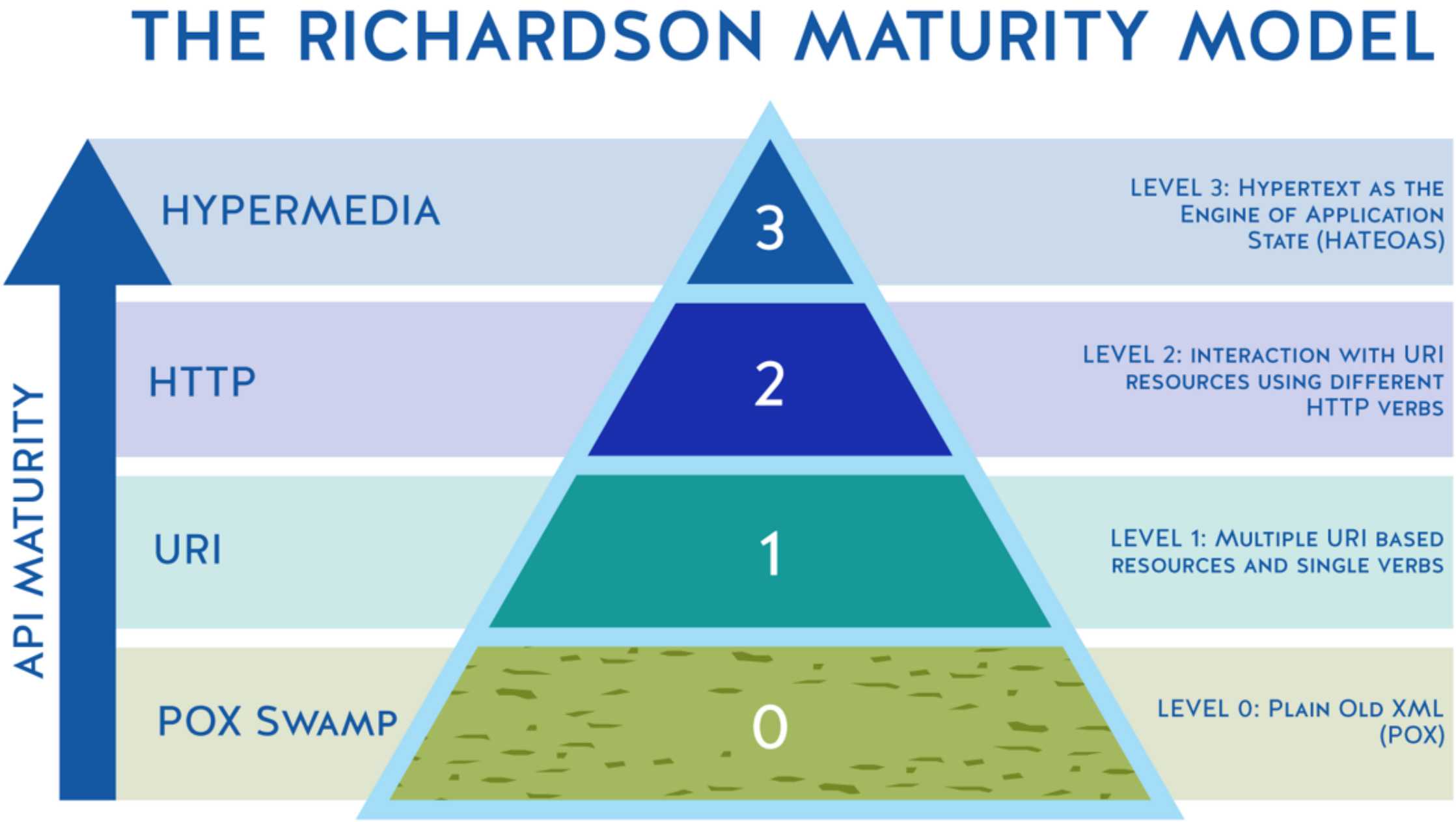
```
{"customers":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
]}
```

JSON Files

Extensión .JSON

MIME Type “application/json”

Richardson Maturity Model



https://www.crummy.com/writing/speaking/2008-QCon/act3.html

HATEOAS Driven REST APIs

HATEOAS (Hypermedia as the Engine of Application State)

Es una restricción de la arquitectura de la aplicación.

Hypermedia se refiere que contiene enlaces a otras formas de media como images, movies, text.

Por ejemplo: `HTTP GET http://api.domain.com/management/departments/10`

```
{
    "departmentId": 10,
    "departmentName": "Administration",
    "locationId": 1700,
    "managerId": 200,
    "links": [
        {
            "href": "10/employees",
            "rel": "employees",
            "type" : "GET"
        }
    ]
}
```

https://restfulapi.net/hateoas/

REST Resource Naming

REST Resource Naming Guide

https://restfulapi.net/resource-naming/

REST API Versioning

URI Versioning

https://example.com/api/v1
https://api.example.com/v1
https://apiv1.example.com

Otros tipos

https://restfulapi.net/versioning/

RESUMEN

Recordemos

El patrón de desarrollo MVC, el cual nos permite descomponer nuestra aplicación en capas o componentes que facilitan el manejo del proyecto.

En la actualidad las aplicaciones web requieren ser seguras, flexibles y escalables, es por eso que Microsoft lanzó una nueva tecnología llamada .NET Core. Esta tecnología se complementa con el uso del framework ASP .NET Core.

RESUMEN

Recordemos

Una alternativa para manejar eficientemente la creación de objetos en nuestro proyecto es utilizar el patrón de inyección de dependencias.

RESTful viene de REpresentational State Transfer

A nivel de representación de información de web services, uno de las principales combinaciones es RESTful API + JSON

REFERENCIAS

Para profundizar

https://www.neosoft.es/blog/que-es-una-aplicacion-web/

https://si.ua.es/es/documentacion/asp-net-mvc-2/modelo-vista-controlador.html

https://msdn.microsoft.com/es-es/magazine/mt694084.aspx

https://github.com/jahbenjah/little-aspnetcore-book/blob/spanish/ElPeque%C3%B1oLibroDeASPNETCore.pdf

https://www.youtube.com/watch?v=Gj_EbIN4P5w

PREGRADO

Ingeniería de Software

Escuela de Ingeniería de Sistemas y Computación | Facultad de Ingeniería

UPC
Universidad Peruana
de Ciencias Aplicadas

Prolongación Primavera 2390,
Monterrico, Santiago de Surco
Lima 33 - Perú
T 511 313 3333
https://www.upc.edu.pe

exígete, innova