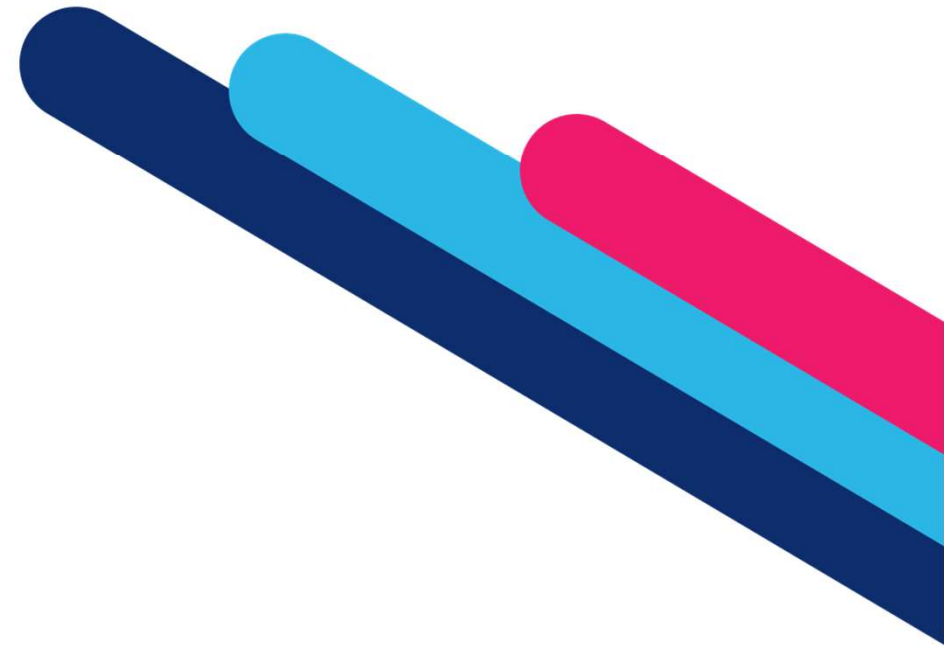
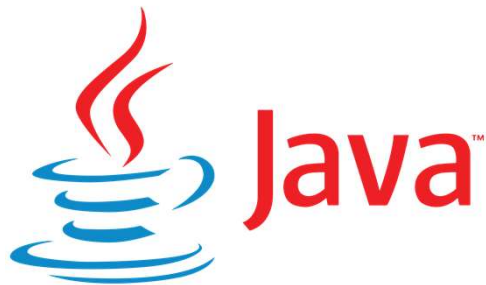




Microservicios con Java

Diseño y Desarrollo de Software

Mg. Jymmy Dextre Alarcón



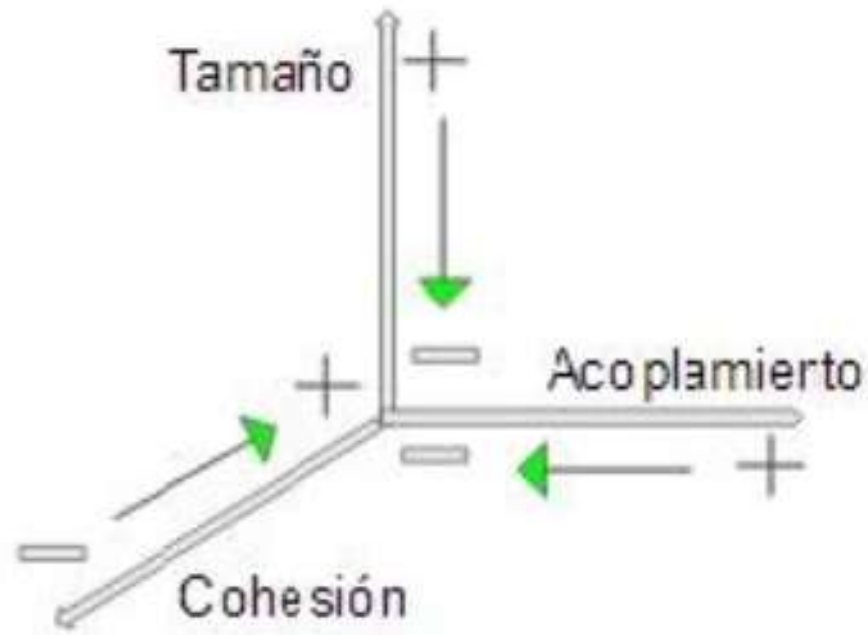


Objetivo de la sesión

- Implementar arquitectura basada en microservicios usando Java con Spring Boot, Eureka, Zuul y Maria DB considerando los fundamentos de programación.

Motivación

¿Qué debemos considerar a la hora de programar?



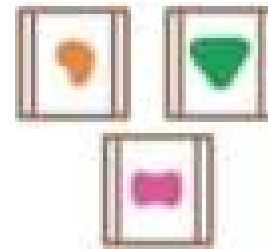


Bases de programación

- **Abstracción:**
 - Proceso mental de extracción de las características esenciales de algo, ignorando detalles superfluos.
- **Encapsulación:**
 - Proceso que se ocultan detalles del soporte de las características de una abstracción.
- **Modularización:**
 - Proceso de descomposición de un sistema en un conjunto de módulos poco acoplados y cohesivos.
- **Jerarquía:**
 - Proceso de estructuración que se produce una organización (jerarquía) de un conjunto de elementos en grados o niveles de responsabilidad.

Surgimiento de Microservicios

- **Arquitecturas monolíticas** -> una sola pieza, nivel de complejidad más alta. En caso de despliegue se hace toda la pieza. El fallo en una parte de la aplicación, provoca el fallo de toda la aplicación.
- **Microservicios** -> solución a las falencias de la arquitectura monolíticas.



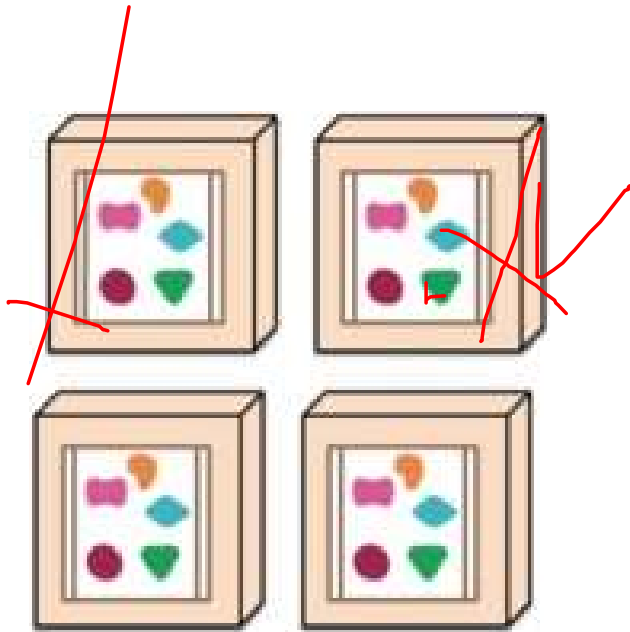


Surgimiento de Microservicios

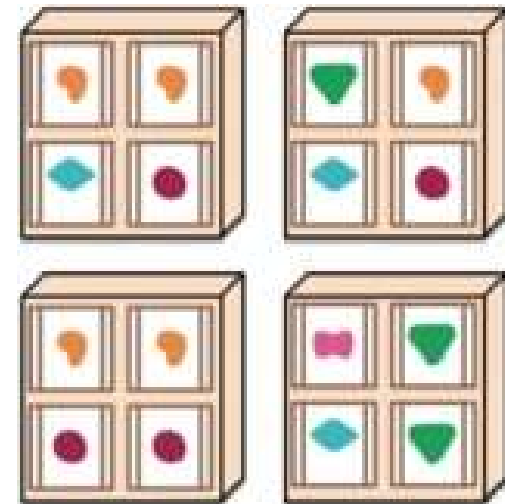
- **Definición:** *Servicios pequeños y autónomos que trabajan bien juntos modelado alrededor de un dominio de negocio.*
- Autónomo -> **característica importante!**
- Son procesos independientes.
- Se comunican, predominantemente, sobre API.
API = Application Programming Interface. Es una capa de abstracción.
- Alto nivel de autonomía.
- Son pequeños, se enfocan en hacer una sola cosa y la hacen bien!

Microservicios

Escalabilidad

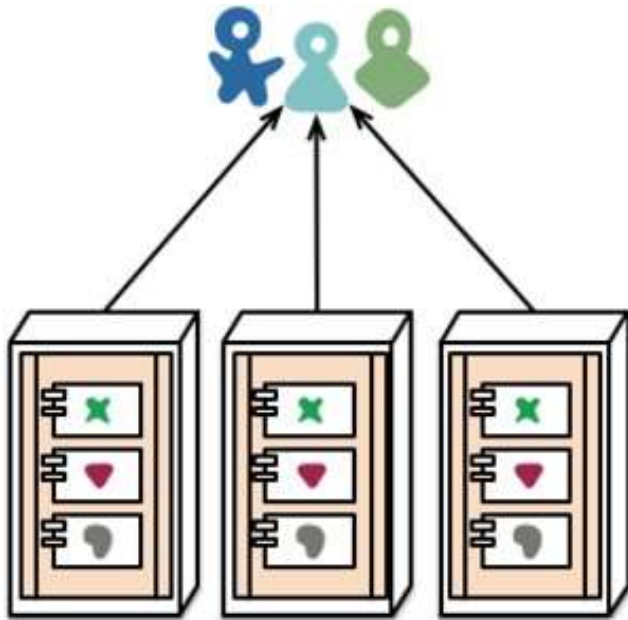


Monolítico

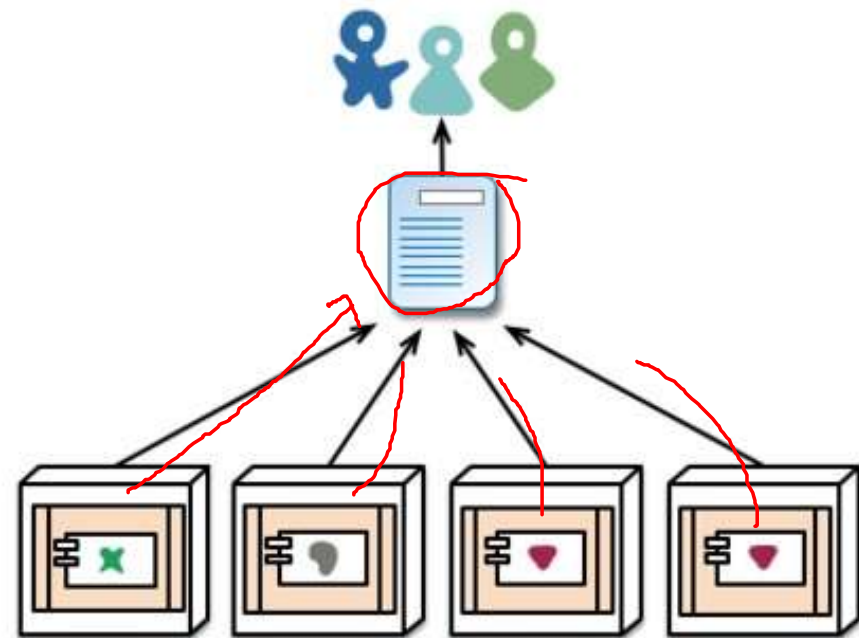


Microservicios

Microservicios

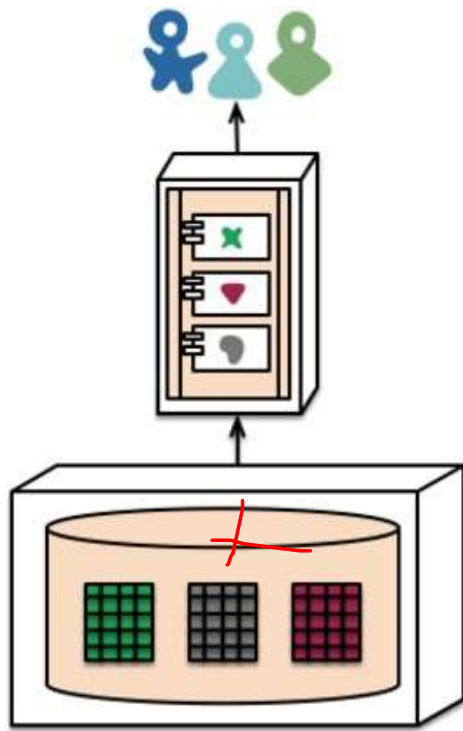


monolith - multiple modules in the same process

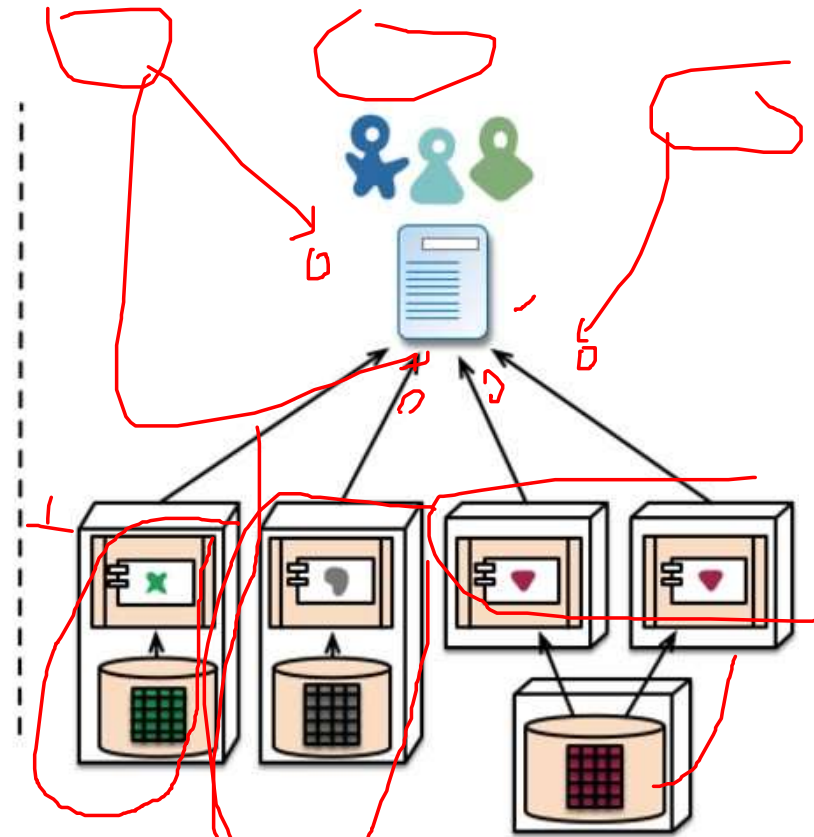


microservices - modules running in different processes

Microservicios



monolith - single database



microservices - application databases

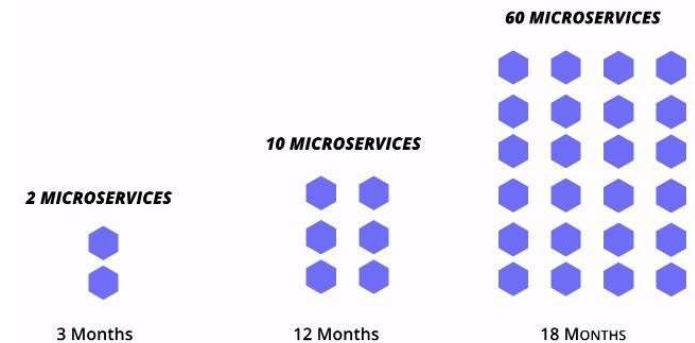


Ventajas de Microservicios

- El desarrollo se vuelve más autónomo e independiente, se asignan los microservicios a equipos específicos.
- Escalabilidad más objetiva, porque se puede crecer, escalar, los microservicios, etc.
- Cambios rápidos (despliegue autónomo).
- Adoptar tecnologías mas fácil y rápidamente.
- Adoptar la incertidumbre digital.

Desventajas de Microservicios

- Te da muchas opciones (podría ser muy bueno), pero en organizaciones grandes incrementa la complejidad, además si te encuentras en organizaciones burocráticas puede ser mas lento su implementación.
- Toma tiempo para obtener los beneficios.
- Las pruebas de integración podrían ser mas complejas.
- Monitoreo es mas complejo, ya que la palabra micro implica que tengamos una gran cantidad de servicios, en pequeña escala, y la cantidad a monitorear será amplia.
- La resiliencia NO es gratuita. Errores en cascada podrían ser muy peligrosos por ejemplo.
- Sistemas Distribuidos son más difíciles. Gran cantidad a distribuir, muchos nodos, complejo comparado con monolítico.



Manos a la obra

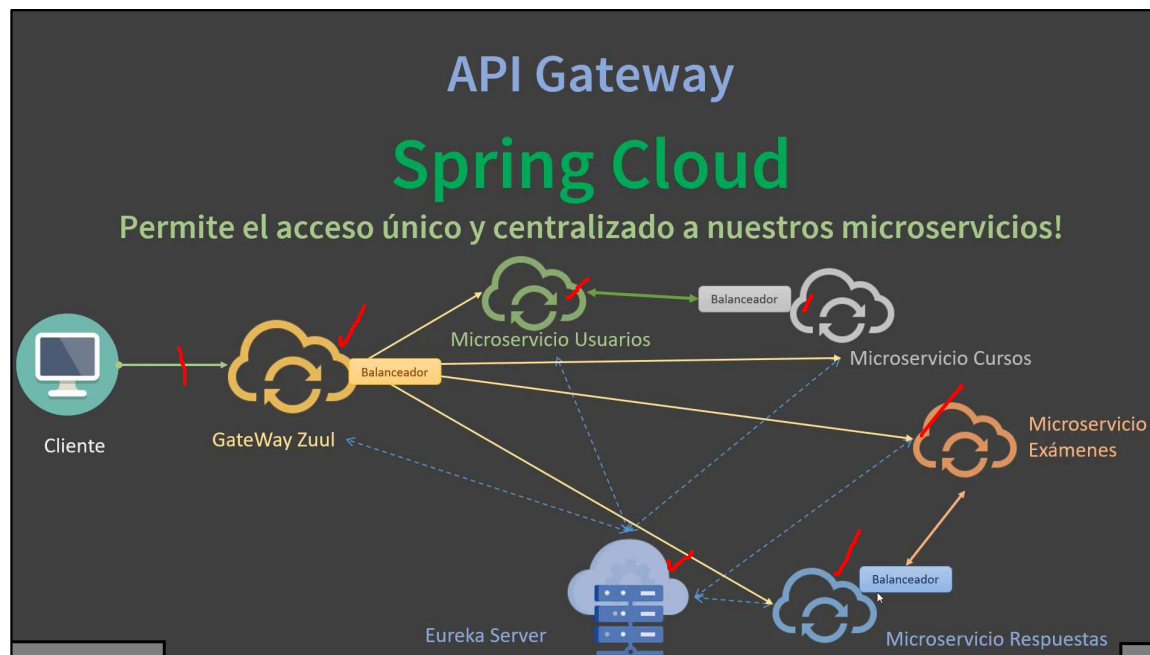
Algunos conceptos:

- Eureka: servidor de registro y descubrimiento de microservicios, open source desarrollado por Netflix.



Manos a la obra

- Zuul: es una librería generada por Netflix, para tener un solo punto de acceso (gateway) a todo los componentes que hacen parte de nuestro sistema. Por medio de este podemos gestionar el acceso a diferentes microservicios en vez de que cada uno tenga un punto de acceso único.






Manos a la obra ...


... mas conceptos


- En casi cualquier proyecto donde haya microservicios, es deseable que todas las comunicaciones entre esos microservicios pasen por un lugar común, de tal manera que se registren las entradas y salidas, se pueda implementar seguridad o se puedan redirigir las peticiones dependiendo de diversos parámetros.
- Maven: herramienta para gestionar y construir proyectos en Java.
- Postman: es una plataforma de colaboración líder para el desarrollo de API. Usada por más de 7 millones de desarrolladores y más de 300 000 empresas en todo el mundo. Postman permite a los usuarios diseñar, simular, depurar, probar, documentar, supervisar y publicar **API**.


Herramientas


- Instaladores:
 - Spring tool Suite, tanto **Git** como **Maven** son también agregados a la instalación.
 - JDK 8.
 - Maria DB.

 jdk-8u251-windows-x64.exe ✓

 mariadb-10.5.5-winx64.msi ✓

 Postman-win64-7.28.0-Setup.exe ✓

 SoapUI-x64-5.6.0.exe ✓

 spring-tool-suite-4-4.5.1.RELEASE-e4.14... ✓




Probando la Arquitectura

Ir a <http://localhost:8761>

Eureka

localhost:8761

HOMELAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2021-04-25T03:05:54 -0500
Data center	default	Uptime	00:18
		Lease expiration enabled	true
		Renews threshold	5
		Renews (last min)	7

DS Replicas

localhost

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
MICROSERVICIOS-USUARIOS	n/a (2)	(2)	UP (2) - microservicios-usuarios:816dc26509012ac67e6affed8134dd0f , microservicios-usuarios:de6e48a287e3cd3088d443aefa60b1cc

General Info

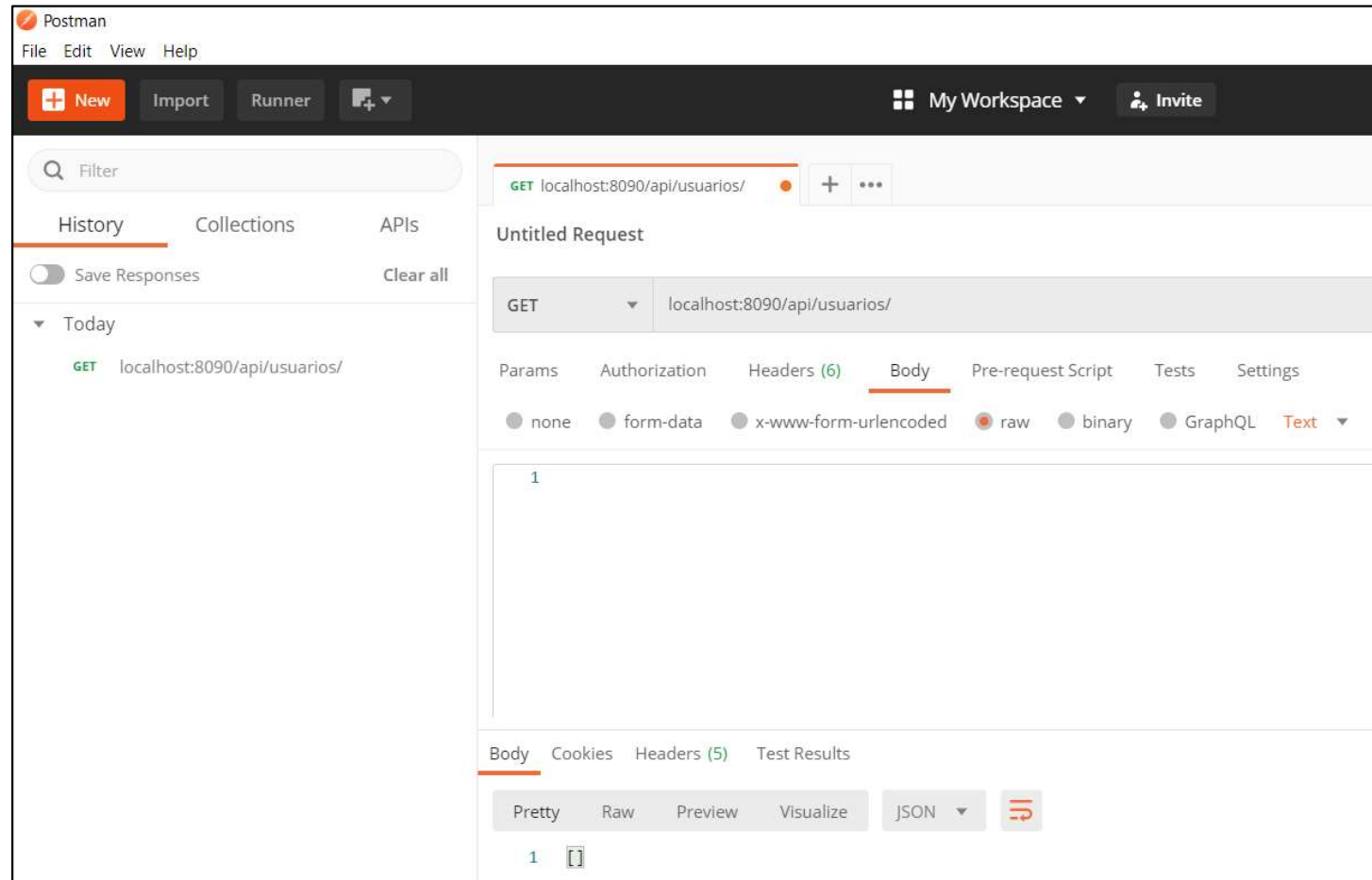
Name	Value
------	-------

CERTUS

Probando con Postman

Ir a localhost:8090/api/usuarios/

Listar



CERTUS

Probando con Postman

Ir a localhost:8090/api/usuarios/

Grabar

The screenshot shows the Postman interface for a POST request to `localhost:8090/api/usuarios/`. The request is configured with the following details:

- Method:** POST
- URL:** localhost:8090/api/usuarios/
- Body Type:** JSON
- Body Content:**

```
1 {
2   "id": 1,
3   "nombres": "Jimmy",
4   "apellidos": "Dextre",
5   "email": "jdextre@certus.edu.pe"
6 }
```

The bottom of the interface shows the response status: **Status: 201 Created**, **Time: 39 ms**, and **Size: 246 B**. The **Headers (5)** tab is currently selected.





Gracias

