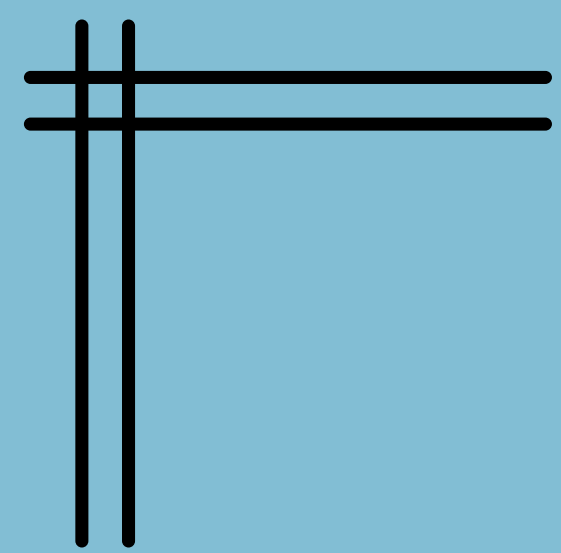


# TALLER 2

Presented by Gabriel De Souza y  
Miguel Torres



# /Agenda

## 1. **Stack**

- a. Lenguajes
- b. Framework
- c. Tecnologías
- d. Patrón

## 2. **Diagramas**

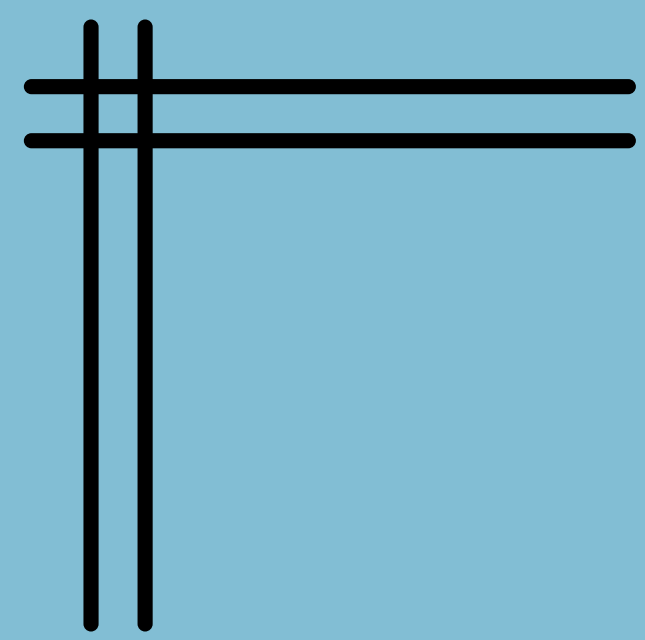
- a. Alto Nivel
- b. Bajo Nivel

## 3. **Repositorio**

## 4. **Demonstración**

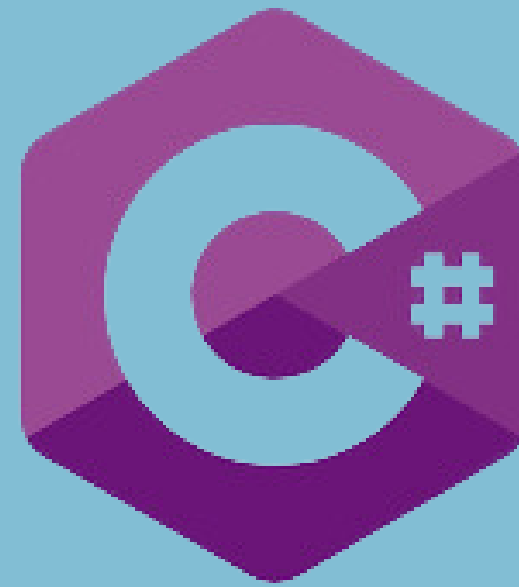
## 5. **Conclusiones**

## 6. **Lecciones Aprendidas**



## /Stack/Lenguajes

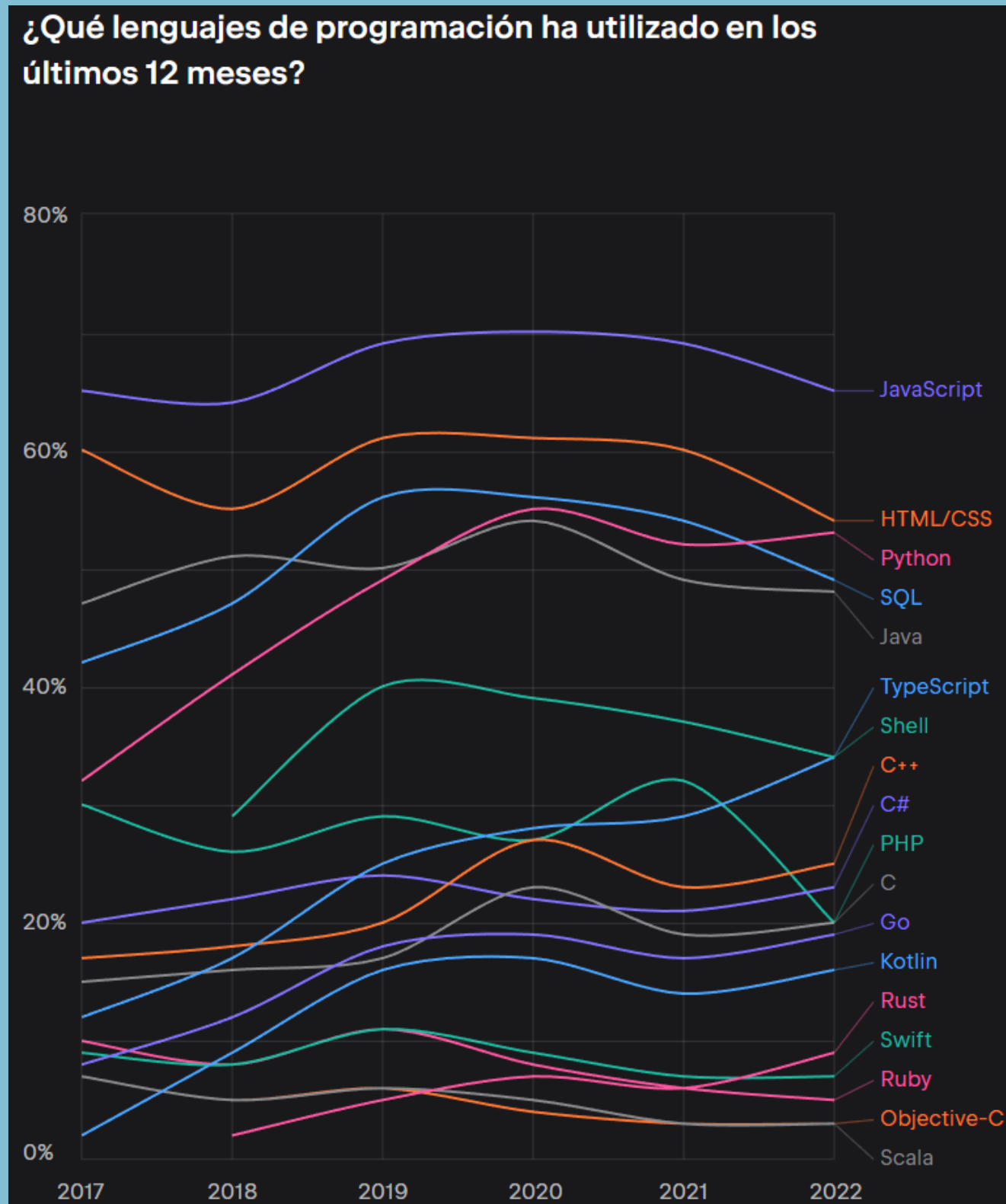
1



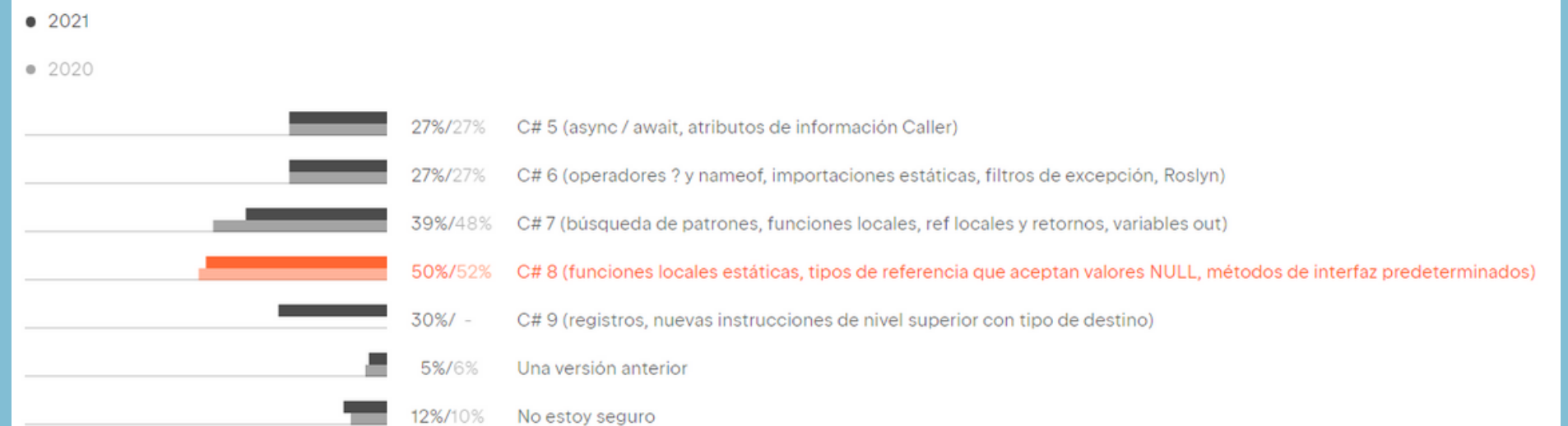
1

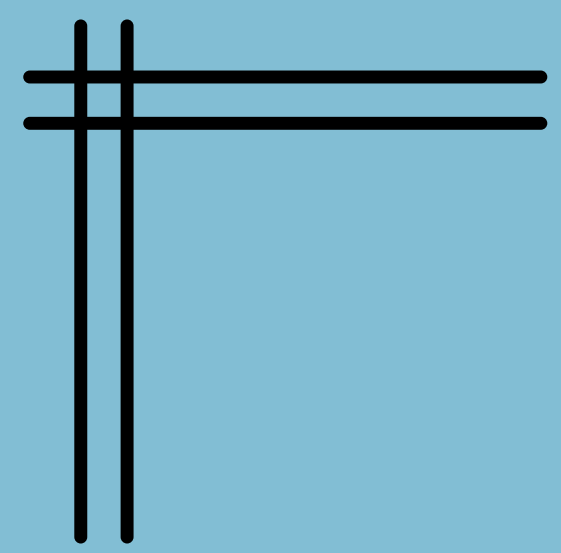


# /Stack/Lenguajes

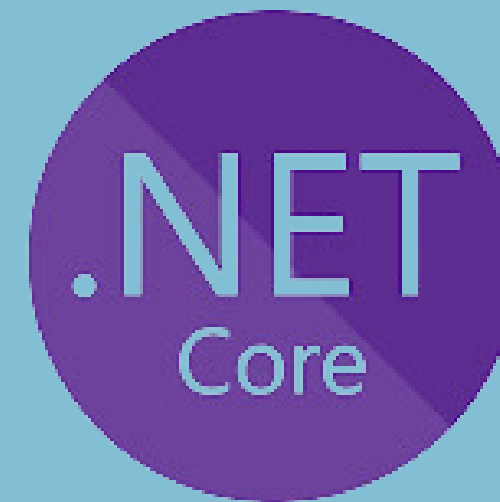


## ¿Qué versiones de C# utiliza habitualmente?

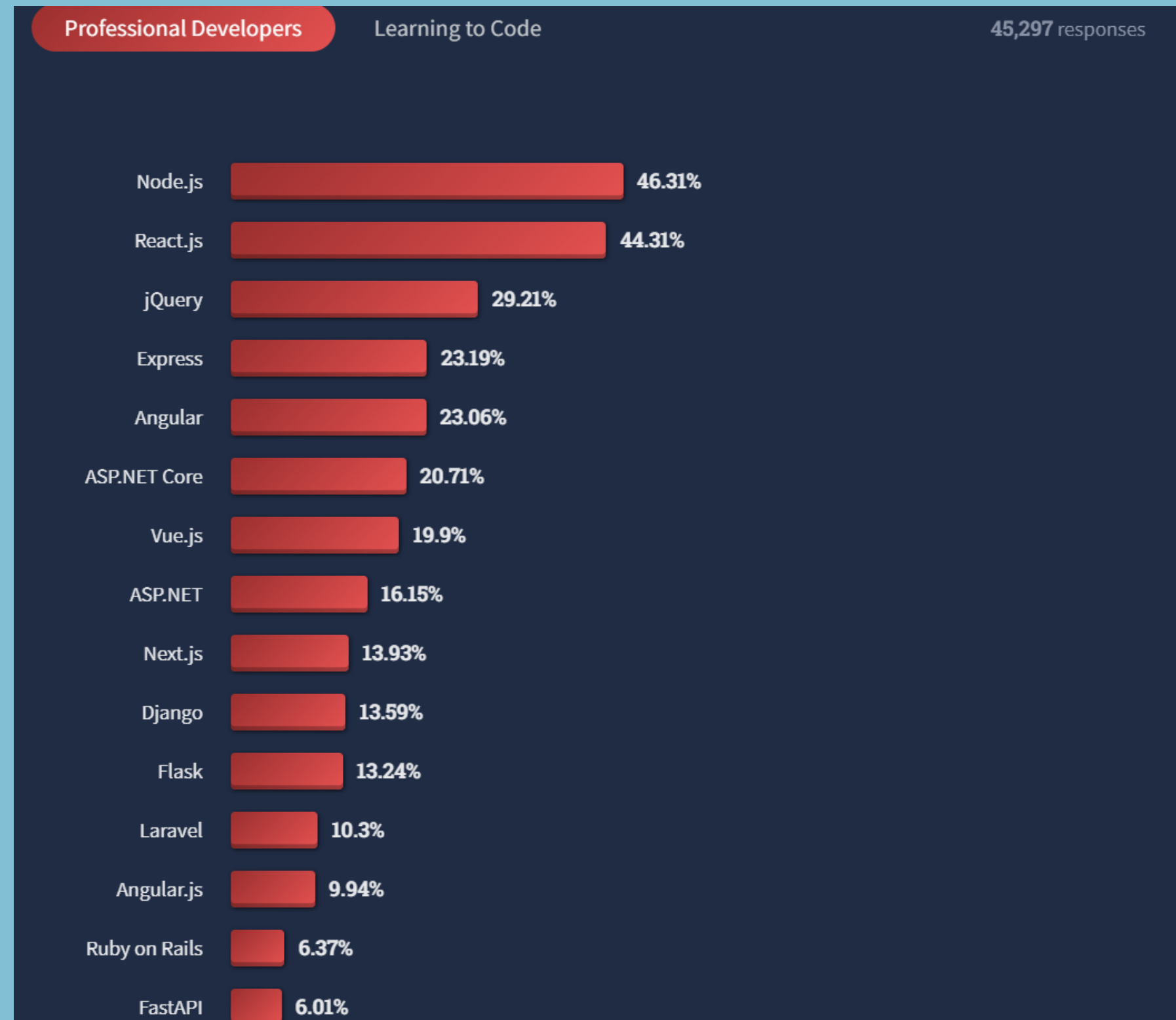




**/Stack/Frameworks**



# /Stack/Frameworks





## /Stack/Tecnologías



Entity Framework

Core



# /Stack/Valor Agregado



**stackshare**

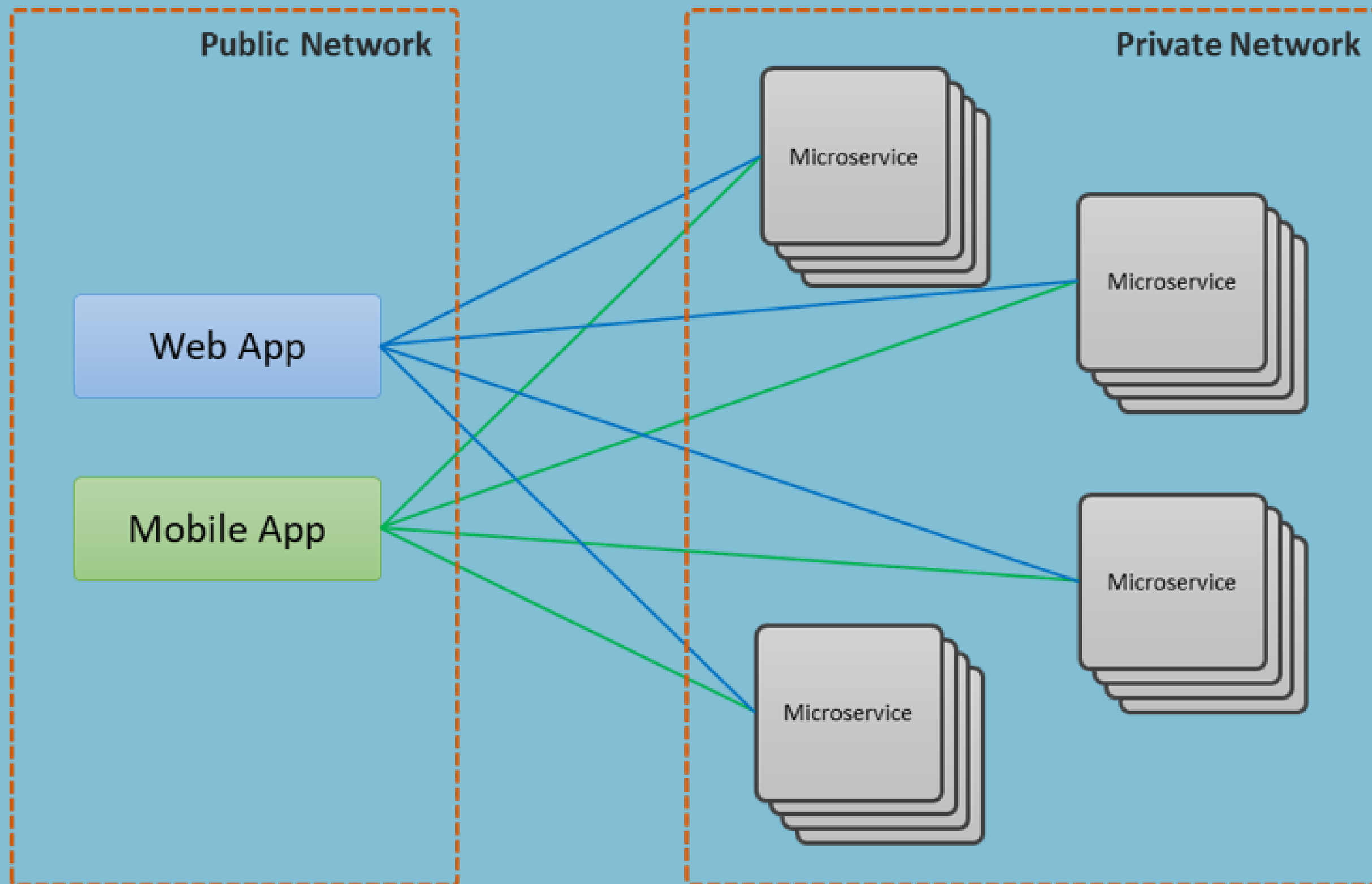
**What is a Tech Stack? What tools do popular tech companies use in 2023?**

A tech stack is defined as the set of technologies an organization uses to build a web or mobile application. It is a combination of programming languages, frameworks, libraries, patterns, servers, UI/UX solutions, software, and tools used by its developers.

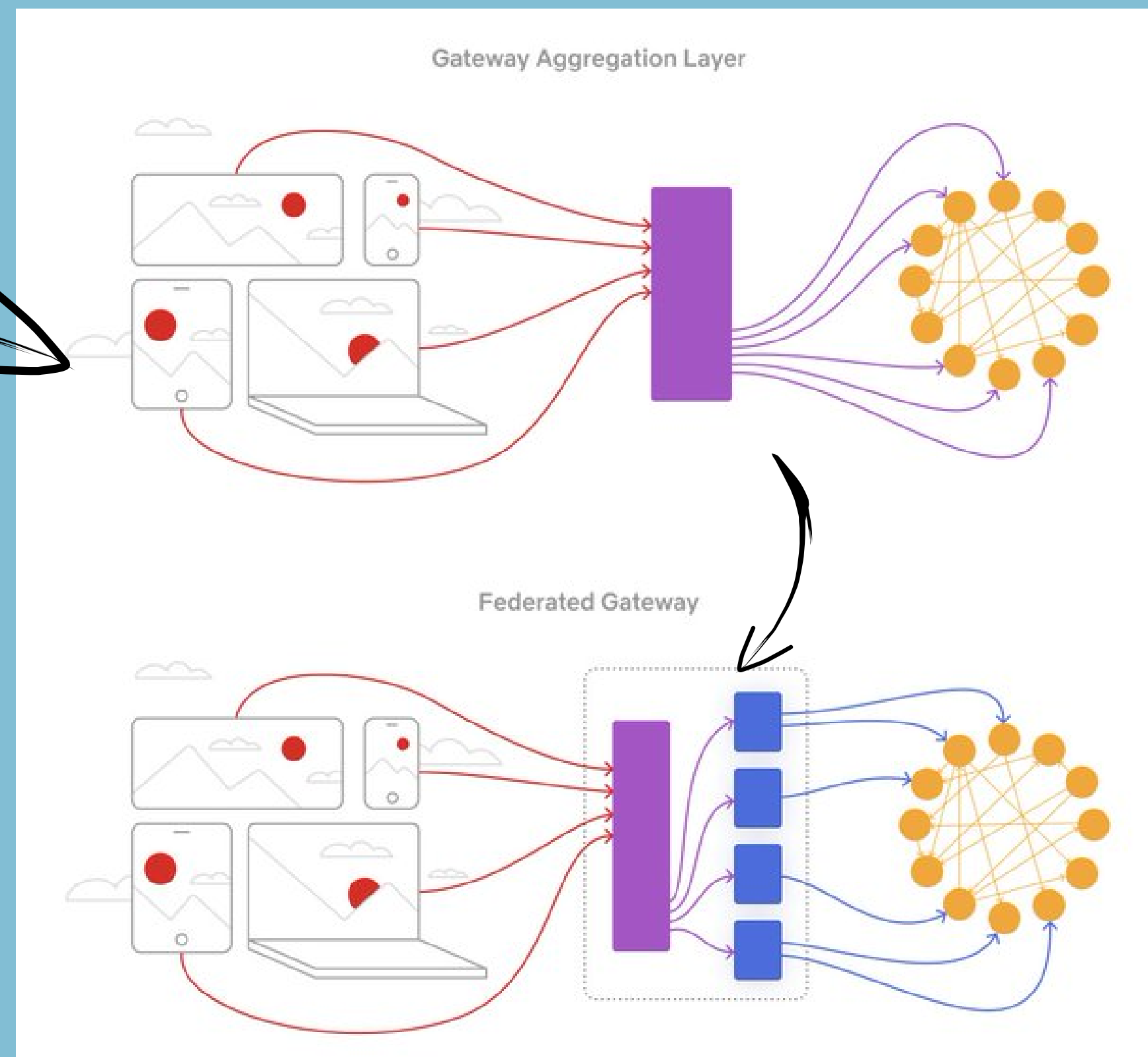
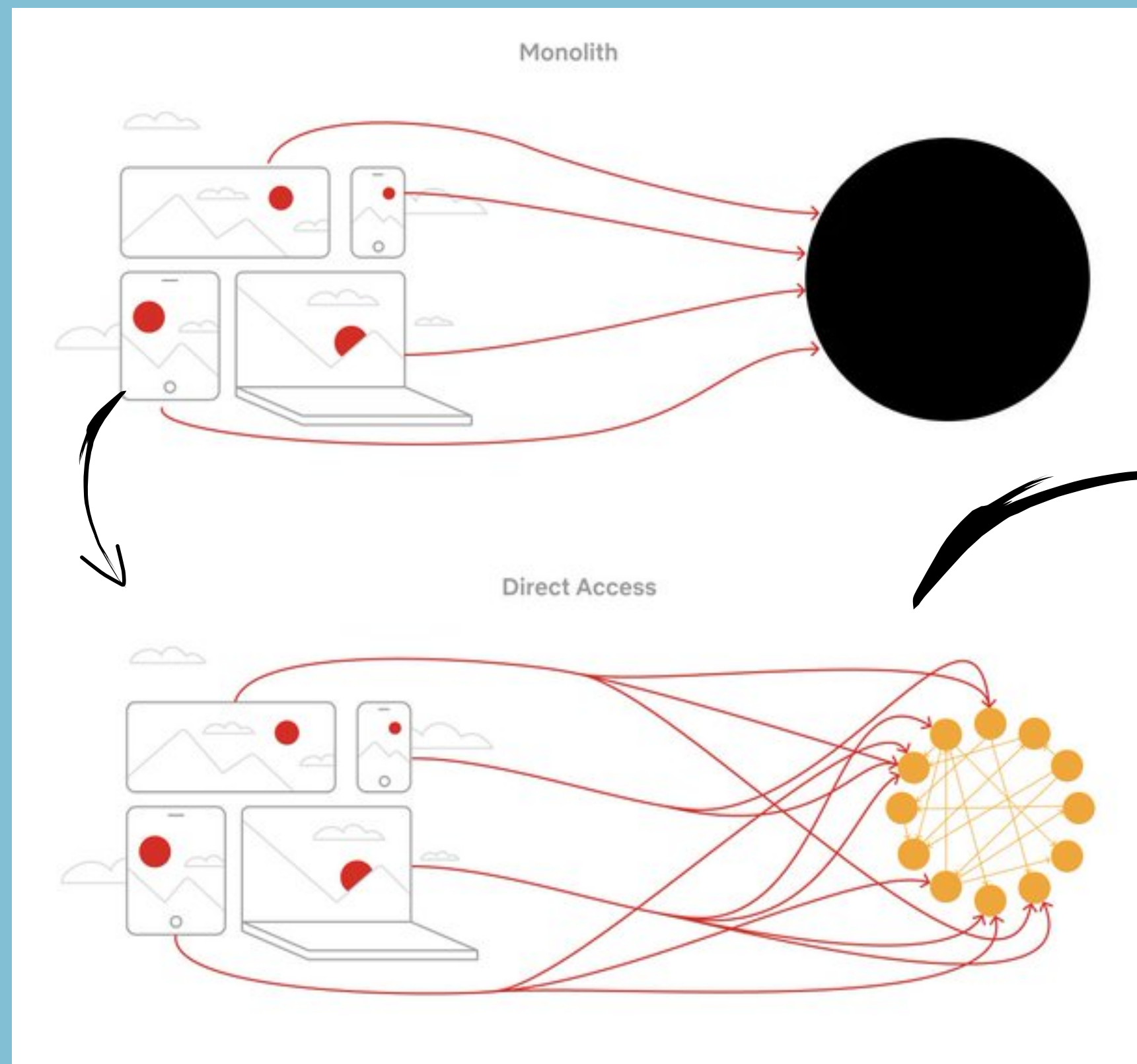
 StackShare



# /Stack/Patrón



# /Stack/Patrón

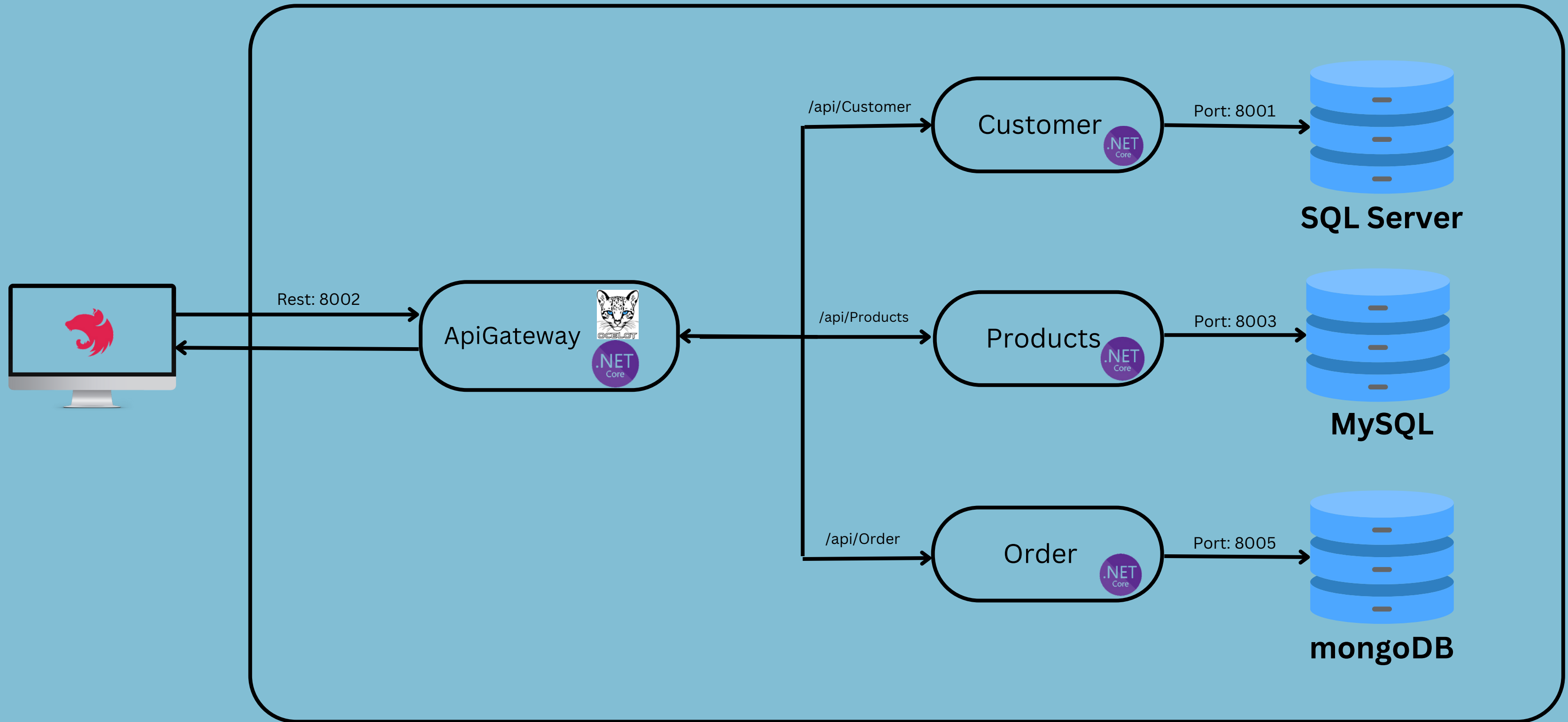


# /Principios

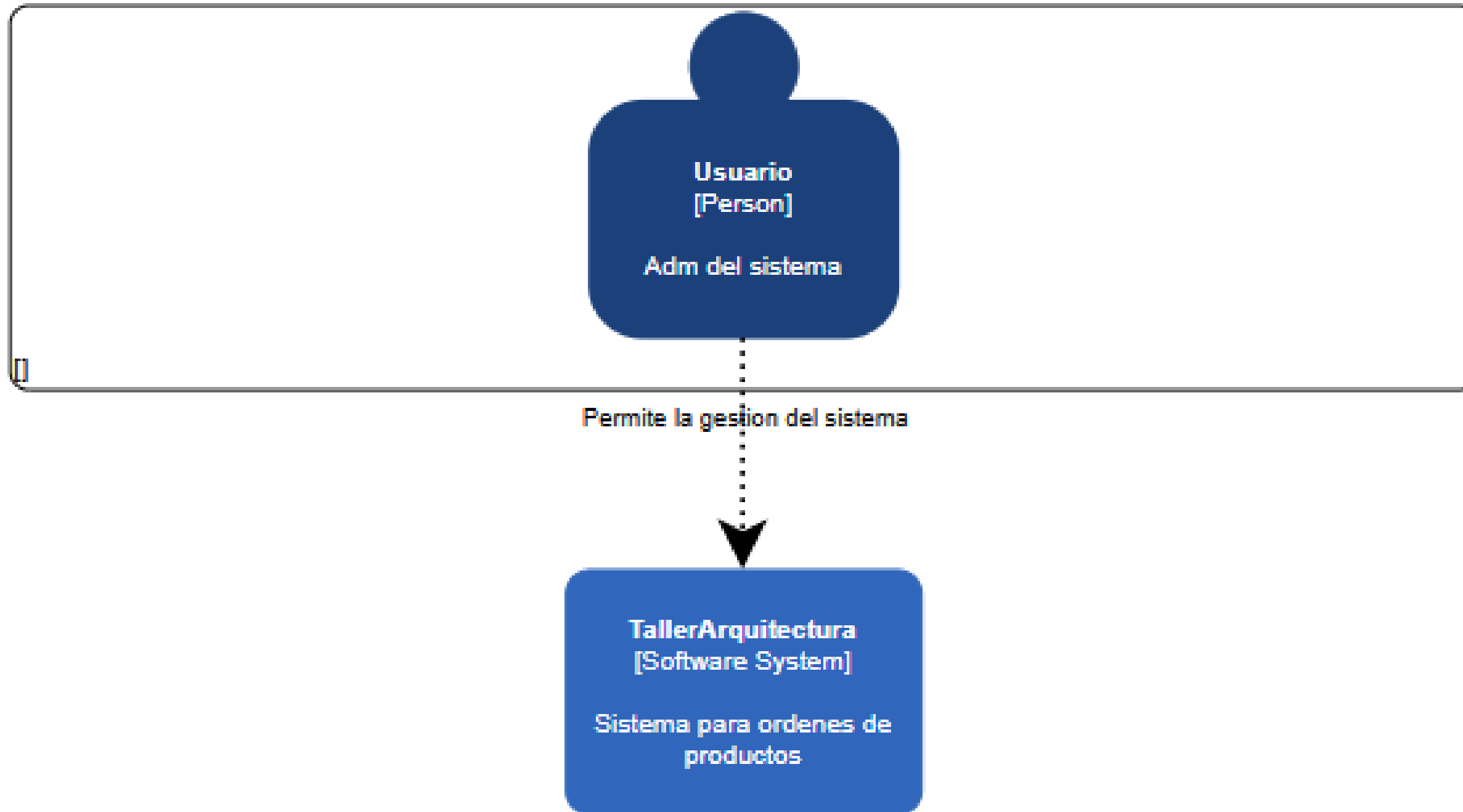
- Principio de Responsabilidad Única (SRP)
- Principio de Abierto/Cerrado (OCP)
- Principio de Sustitución de Liskov (LSP)
- Principio de Segregación de Interfaz (ISP)
- Principio de Inversión de Dependencia (DIP)

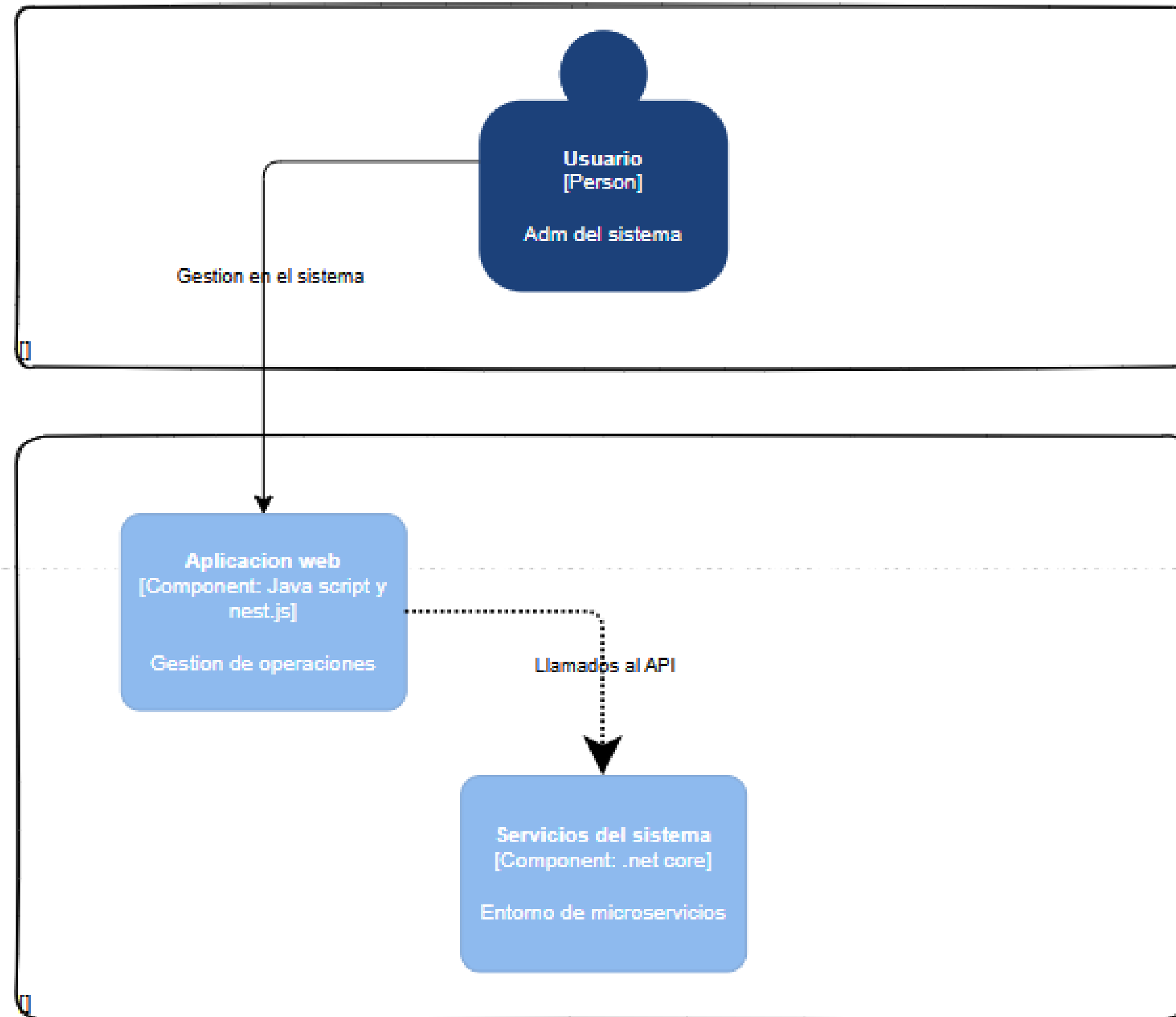
DIAGRAMAS

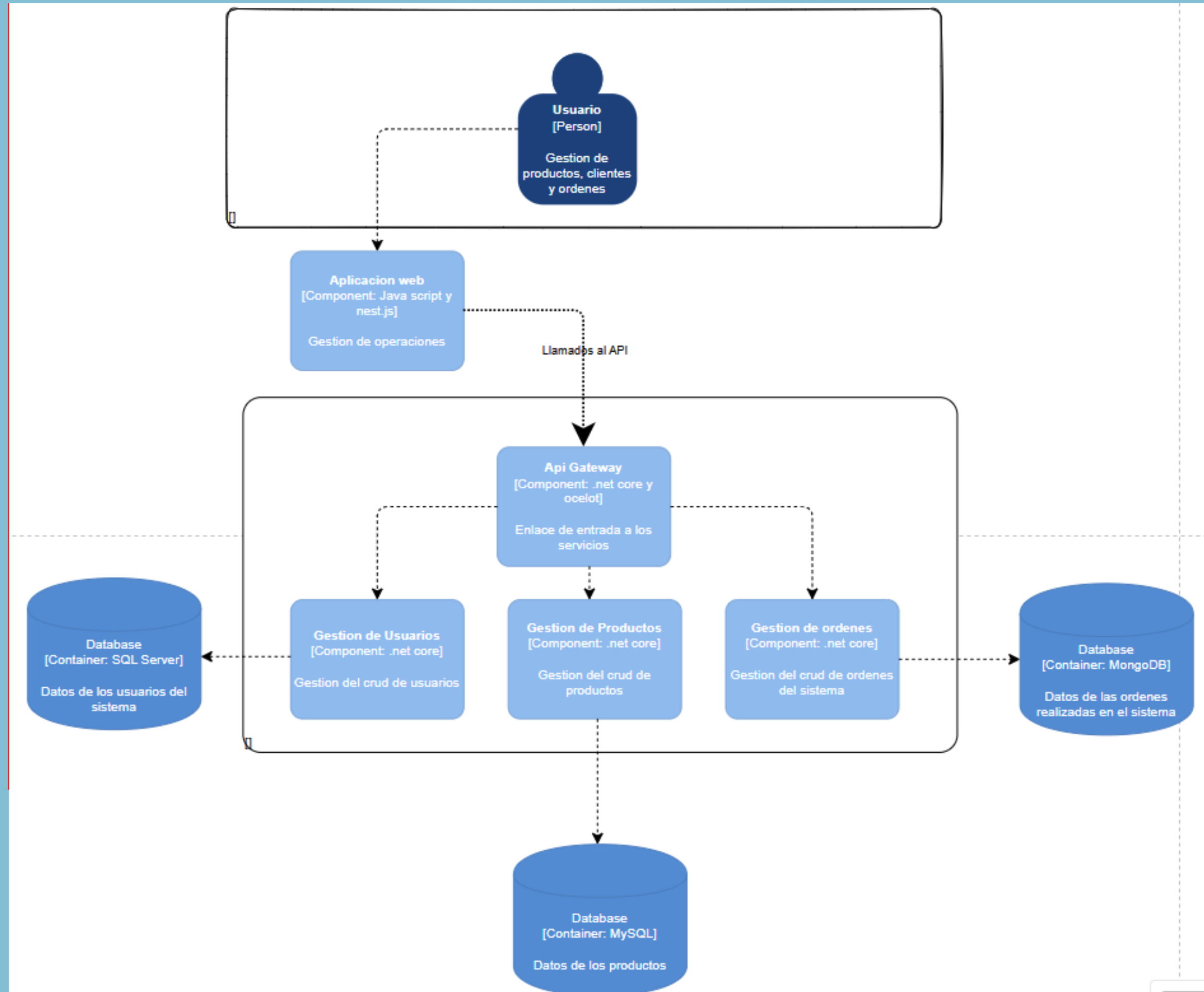
# Diagrama/Alto\_Nivel



# Diagrama/Bajo\_Nivel

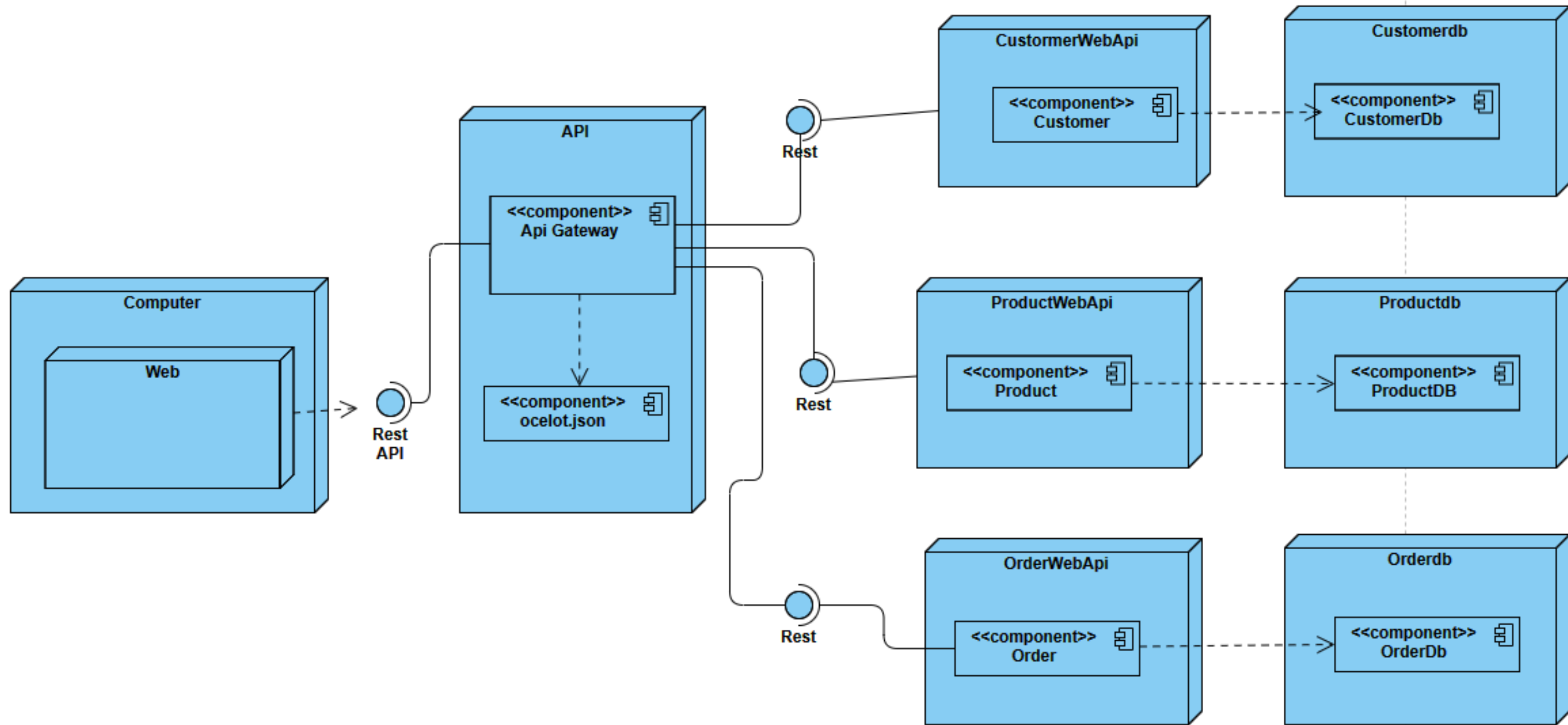




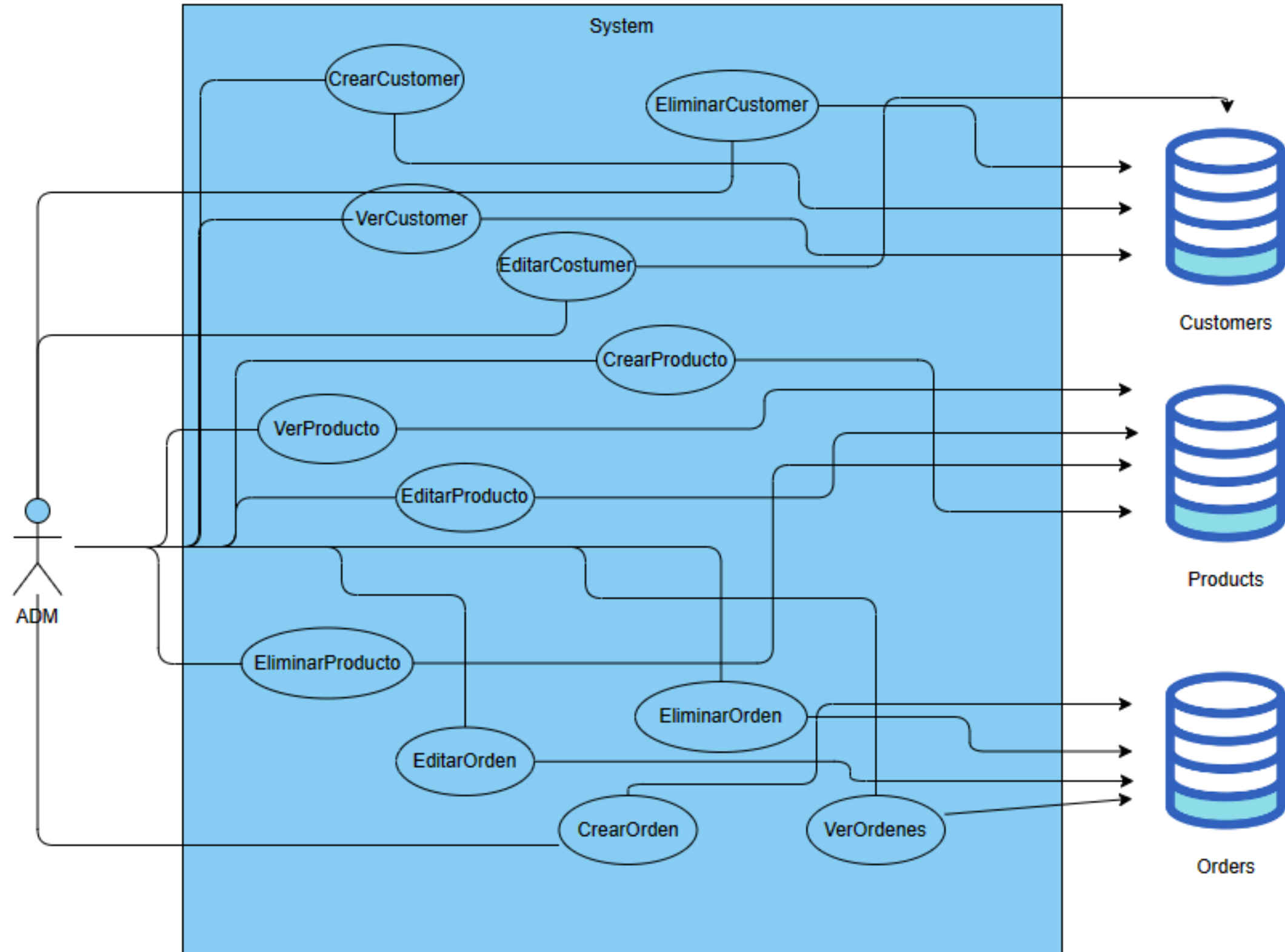




# Diagrama/Despliegue



# Diagrama/Casos de uso



# Github



ZiggyZao/  
**TallerArquitectura-2**



1

Contributor



0

Issues



0

Stars



0

Forks

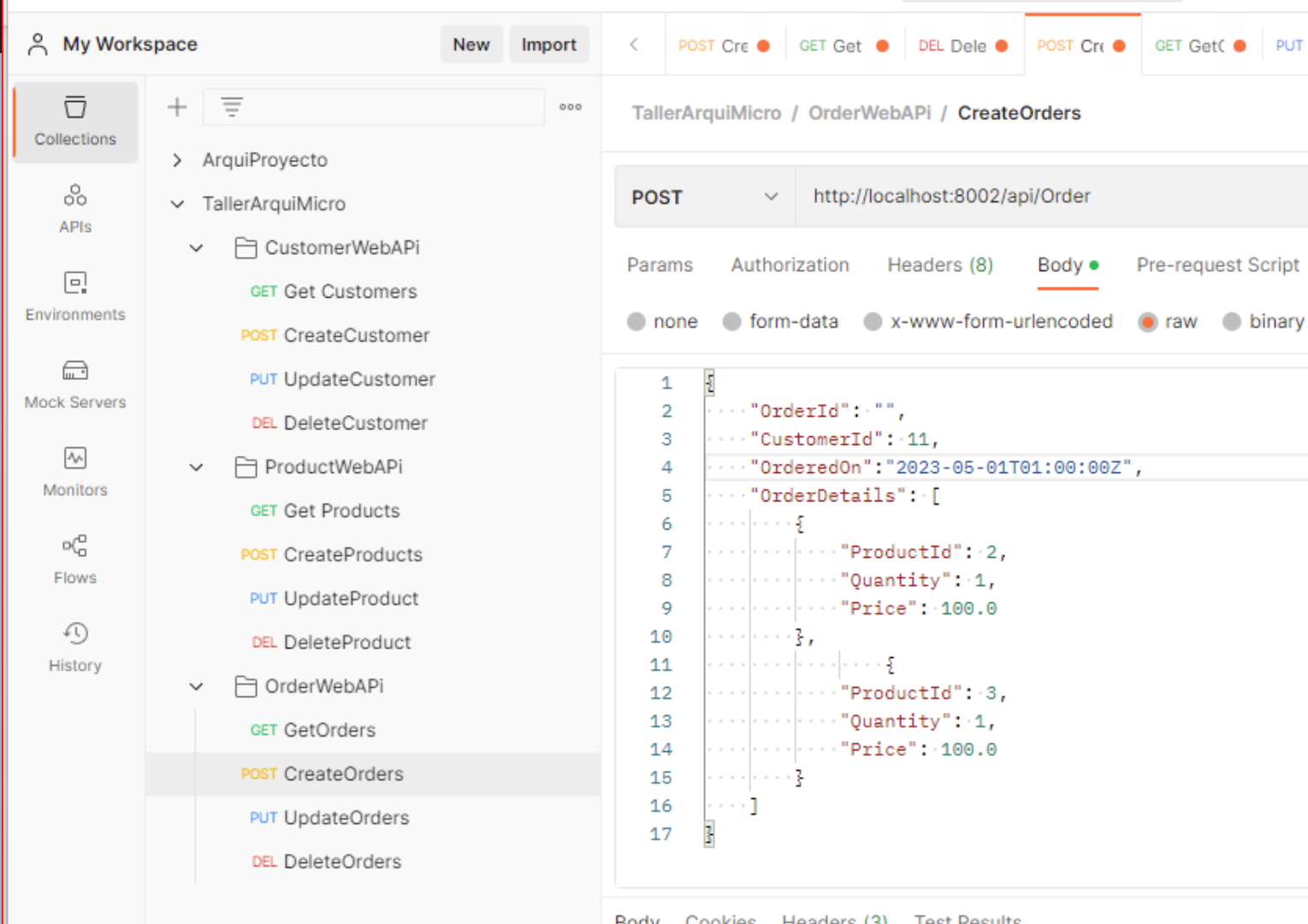
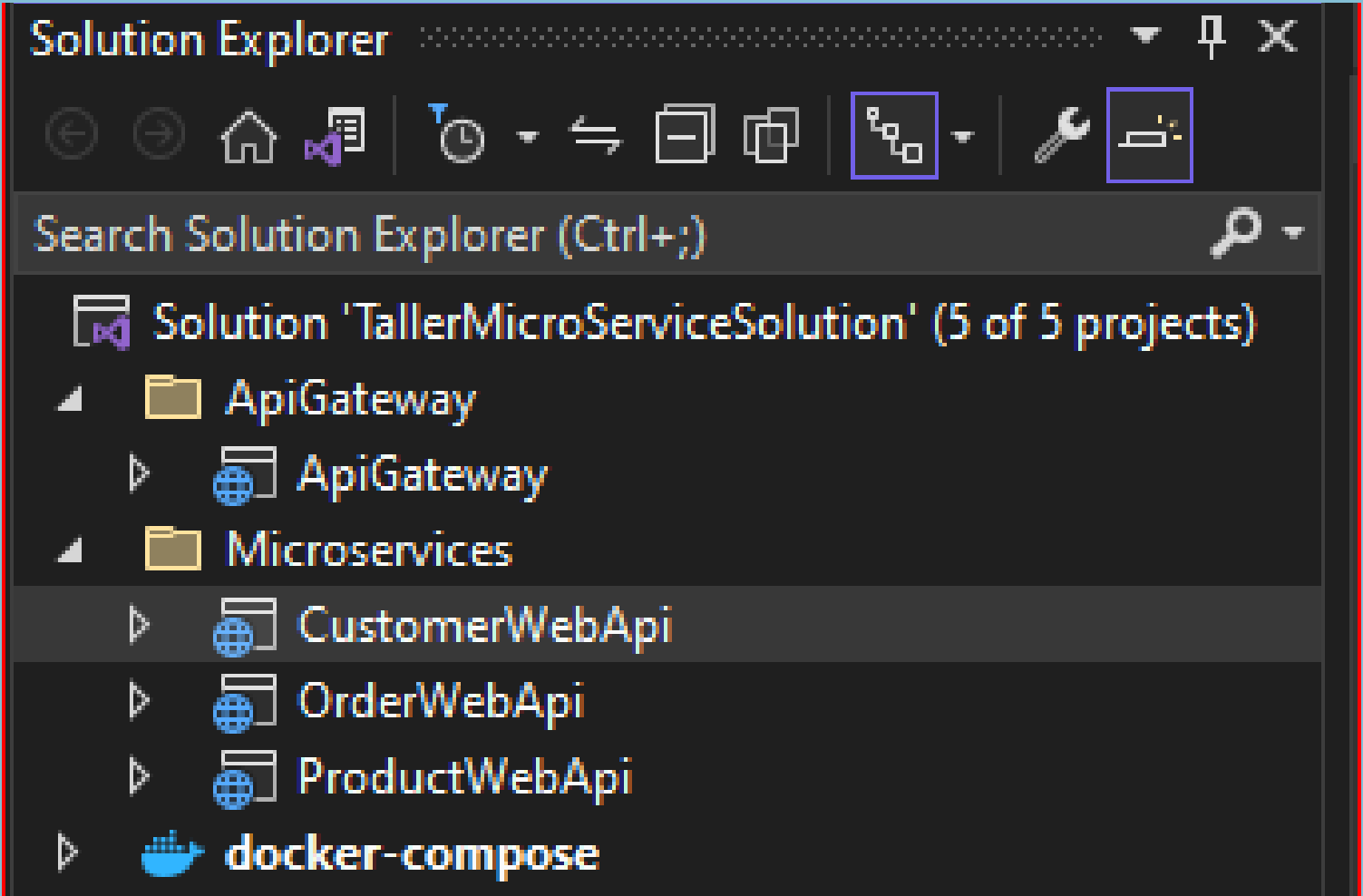


## **ZiggyZao/TallerArquitectura-2**

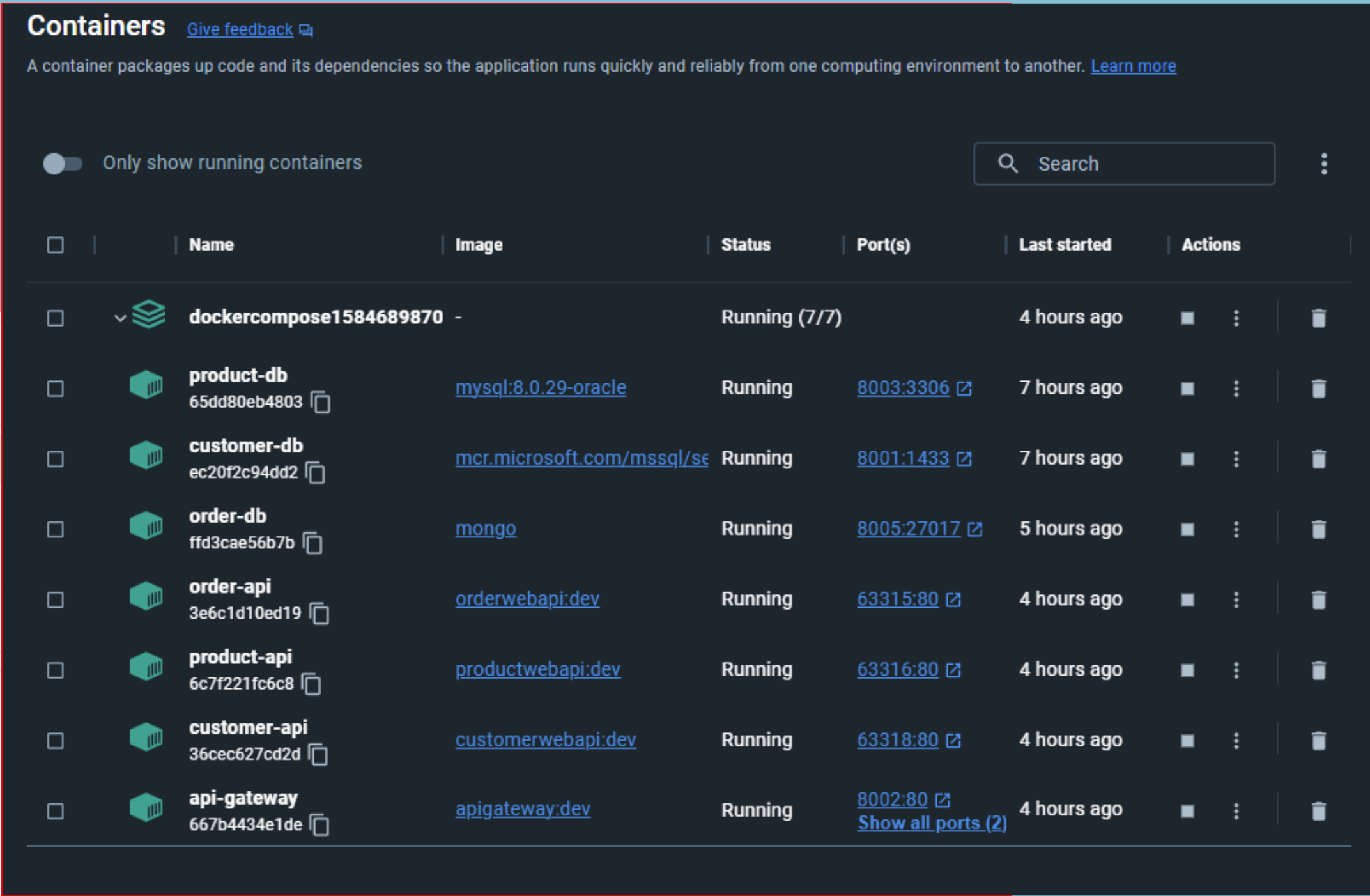
Contribute to ZiggyZao/TallerArquitectura-2 development by creating an account on GitHub.



GitHub



# /Demonstración



# /ValorAgregado

My Workspace

NewImport

POST AddOpe • GET AddOper • DEL DeleteOp • PUT UpdateO • POST ClientRegi • GET Get Cust • POST CreateC • TallerArquiM

No Environment

Collections

ArquiProyecto

TallerArquiMicro

CustomerWebAPI

GET Get Customers

POST CreateCustomer

PUT UpdateCustomer

DEL DeleteCustomer

ProductWebAPI

GET Get Products

POST CreateProducts

PUT UpdateProduct

DEL DeleteProduct

OrderWebAPI

GET GetOrders

POST CreateOrders

PUT UpdateOrders

DEL DeleteOrders

TallerArquiMicro / CustomerWebAPI / Get Customers

Save

Send

GET


http://localhost:8002/api/Customer

ParamsAuthHeaders (6)BodyPre-req. TestsSettingsCookies

Query Params

Key	Value	Description
Key	Value	Description

Response



Click Send to get a response

Code snippet

NodeJs - Native

```
1 var http = require('follow-redirects').
  http;
2 var fs = require('fs');
3
4 var options = {
5   'method': 'GET',
6   'hostname': 'localhost',
7   'port': 8002,
8   'path': '/api/Customer',
9   'headers': {
10   },
11   'maxRedirects': 20
12 };
13
14 var req = http.request(options, function
  (res) {
15   var chunks = [];
16
17   res.on("data", function (chunk) {
18     chunks.push(chunk);
19   });
20
21   res.on("end", function (chunk) {
22     var body = Buffer.concat(chunks);
23     console.log(body.toString());
24   });
25
26   res.on("error", function (error) {
27     console.error(error);
28   });
29 });
30
31 req.end();
```

# **Conclusiones & Lecciones aprendidas**

