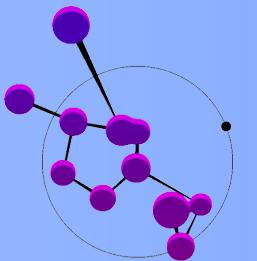


PROYECTO FINAL

INGESOFT 5

  Nicolás Cuéllar Molina, Samuel Alvarez Alban 





QUE ES EL PROYECTO?

Es un backend de comercio electrónico basado en microservicios (usuarios, productos, pagos, pedidos, etc.), cada uno ejecutado en contenedores y gestionado mediante un orquestador en la nube como Gcp Google Gloud

¿POR QUÉ MICROSERVICIOS?

Permiten escalabilidad, despliegue independiente y uso de diferentes tecnologías.

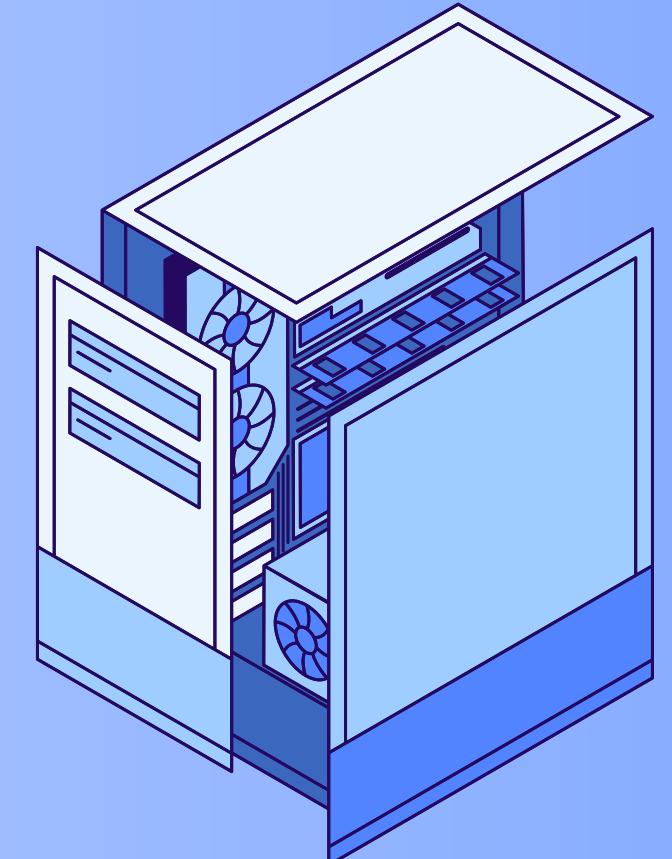


OBJETIVOS

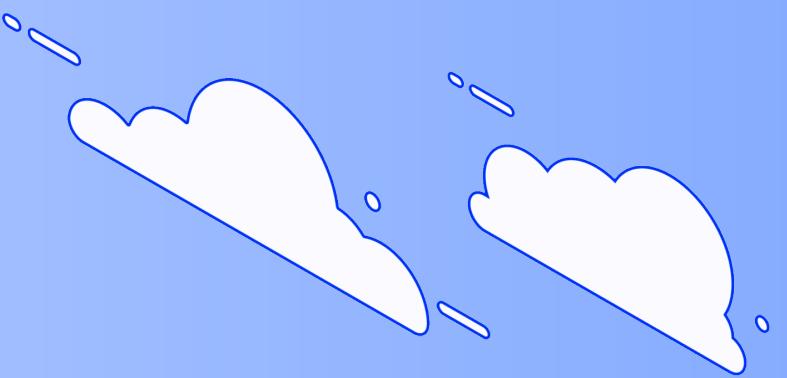
El objetivo fue aprender y aplicar conceptos de DevOps, microservicios y despliegue en la nube.

¿QUÉ RESUELVE?

- Autenticación y gestión de usuarios.
- Catálogo de productos y favoritos.
- Procesamiento de pagos y pedidos.
- Gestión de envíos.
- API Gateway centralizado.
- Configuración y descubrimiento de servicios.
- Observabilidad (logs, trazas, health checks).
- Integración DevOps y despliegue automatizado.

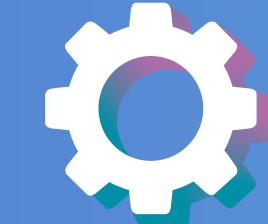


METODOLOGÍA ÁGIL



KANBAN

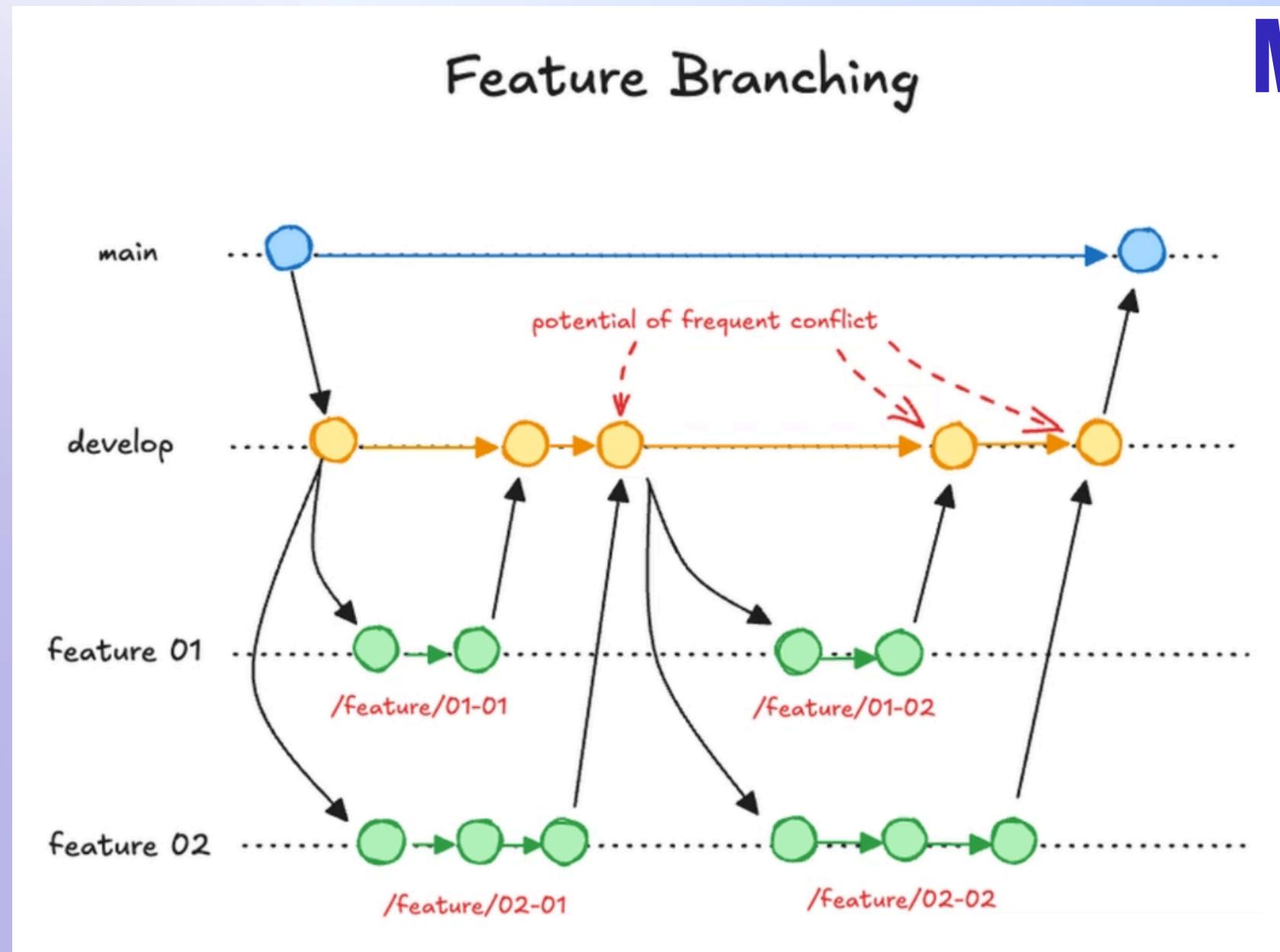
Seleccionada para este proyecto porque permite una gestión altamente visual y flexible del flujo de trabajo, lo cual resulta ideal en un entorno de desarrollo de microservicios donde las tareas suelen ser independientes, técnicas y con distinta complejidad.



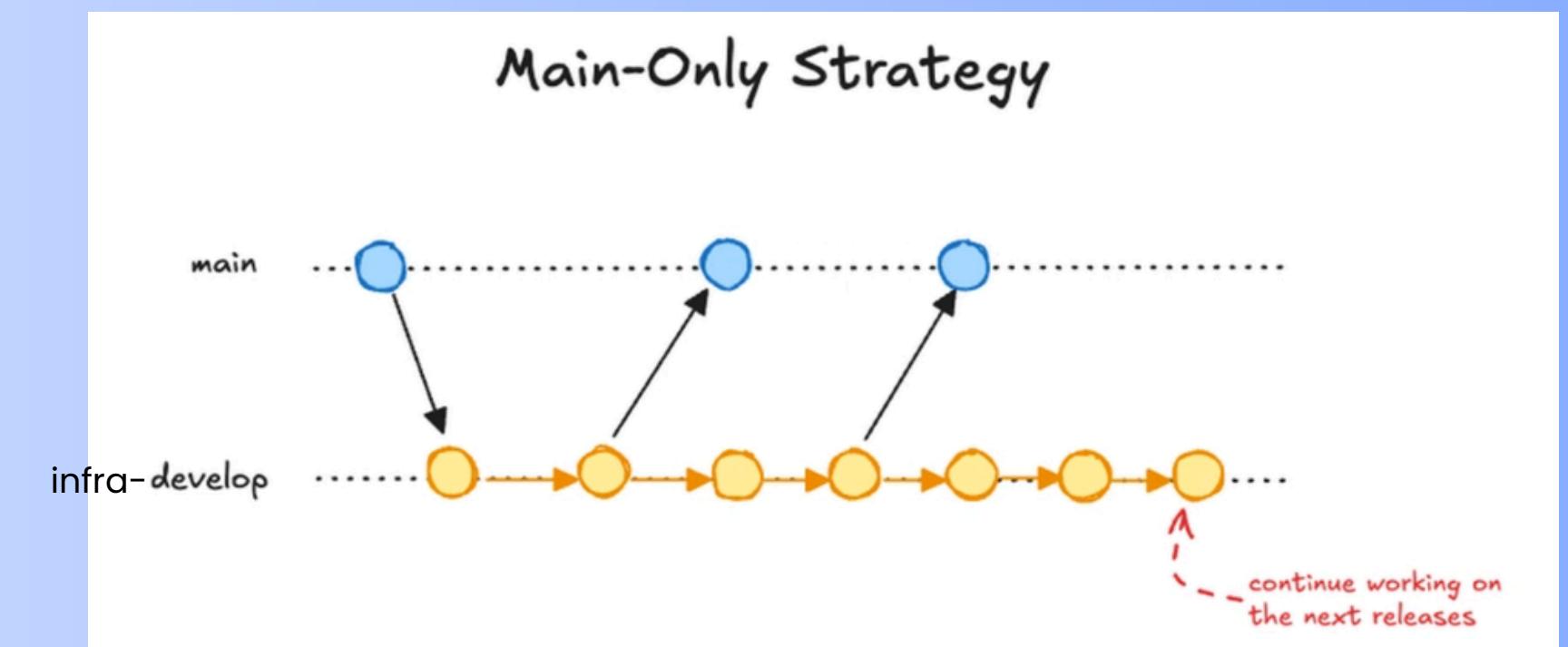
ESTRATEGIA DE BRANCHING

DESARROLLO

OPERACIONES



MASTER



GITHUB PROJECTS

⊕ **Microservices App Ecommerce Kanban**

Add status update Insights Workflows 7

Backlog Priority board Team items Roadmap My items + New view

Filter by keyword or by field Discard Save

Backlog 28 Estimate: 0 ...
This item hasn't been started
Draft
Crear configuraciones de ambiente (dev, stage, prod)
Draft
Crear diagramas de arquitectura IaC
Draft
Integrar SonarQube (estático) en CI
Draft
Integrar Trivy para escaneo de contenedores
Draft
+ Add item

In progress 3 / 3 Estimate: 0 ...
This is actively being worked on
Draft
Crear estructura de carpetas Terraform modular con state remoto
Draft
Implementar o mejorar al menos tres patrones adicionales
Draft
Documentar patrones implementados
+ Add item

In review 1 / 5 Estimate: 0 ...
This item is in review
Draft
Identificar y Documentar patrones actuales (propósito y uso)
+ Add item

Done 3 Estimate: 0 ...
This has been completed
Draft
Definir estrategias de Branching
Draft ...
[Documentar metodología ágil seleccionada \(Scrum/Kanban\)](#)
Draft
Crear Historias de Usuario
+ Add item

HISTORIAS DE USUARIO

Backlog del Proyecto E-commerce Microservices

Este documento presenta el backlog del proyecto estructurado según estándares de la industria, separando claramente las **Historias de Usuario** (valor de negocio) de las **Tareas Técnicas y Enablers** (infraestructura y soporte).

1. Product Backlog (Historias de Usuario)

HU-01: Exploración de Productos

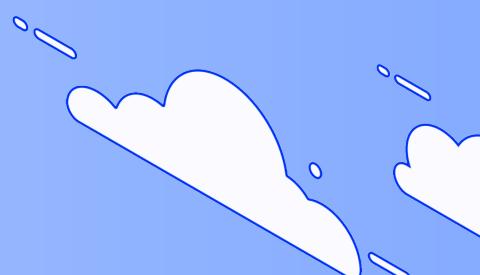
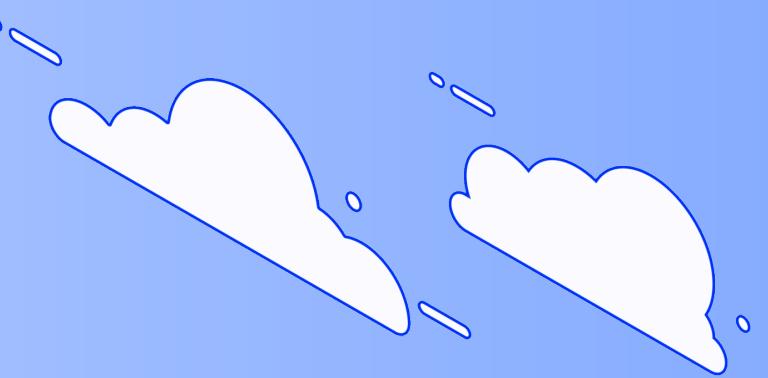
Como cliente de la tienda, Quiero ver un listado de productos disponibles con sus precios e imágenes, Para decidir qué artículos comprar.

Criterios de Aceptación:

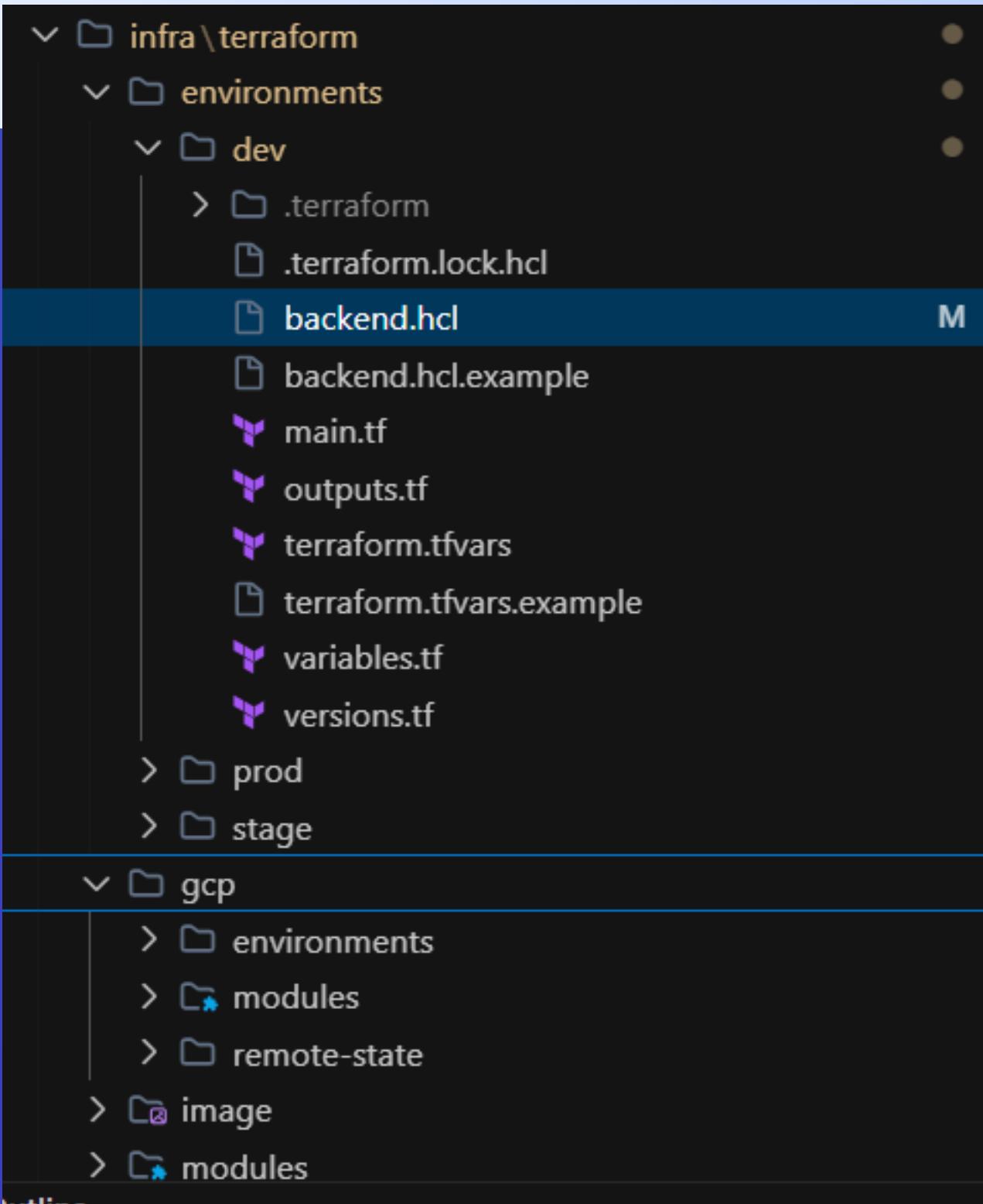
- El sistema muestra una lista paginada de productos (endpoint `GET /product-service/products`).
- Cada tarjeta de producto incluye: nombre, precio, imagen y stock disponible.
- El tiempo de carga del listado es menor a 2 segundos.

Prioridad: Media (P2) | Estimación: S

INFRAESTRUCTURA



INFRAESTRUCTURA COMO CODIGO

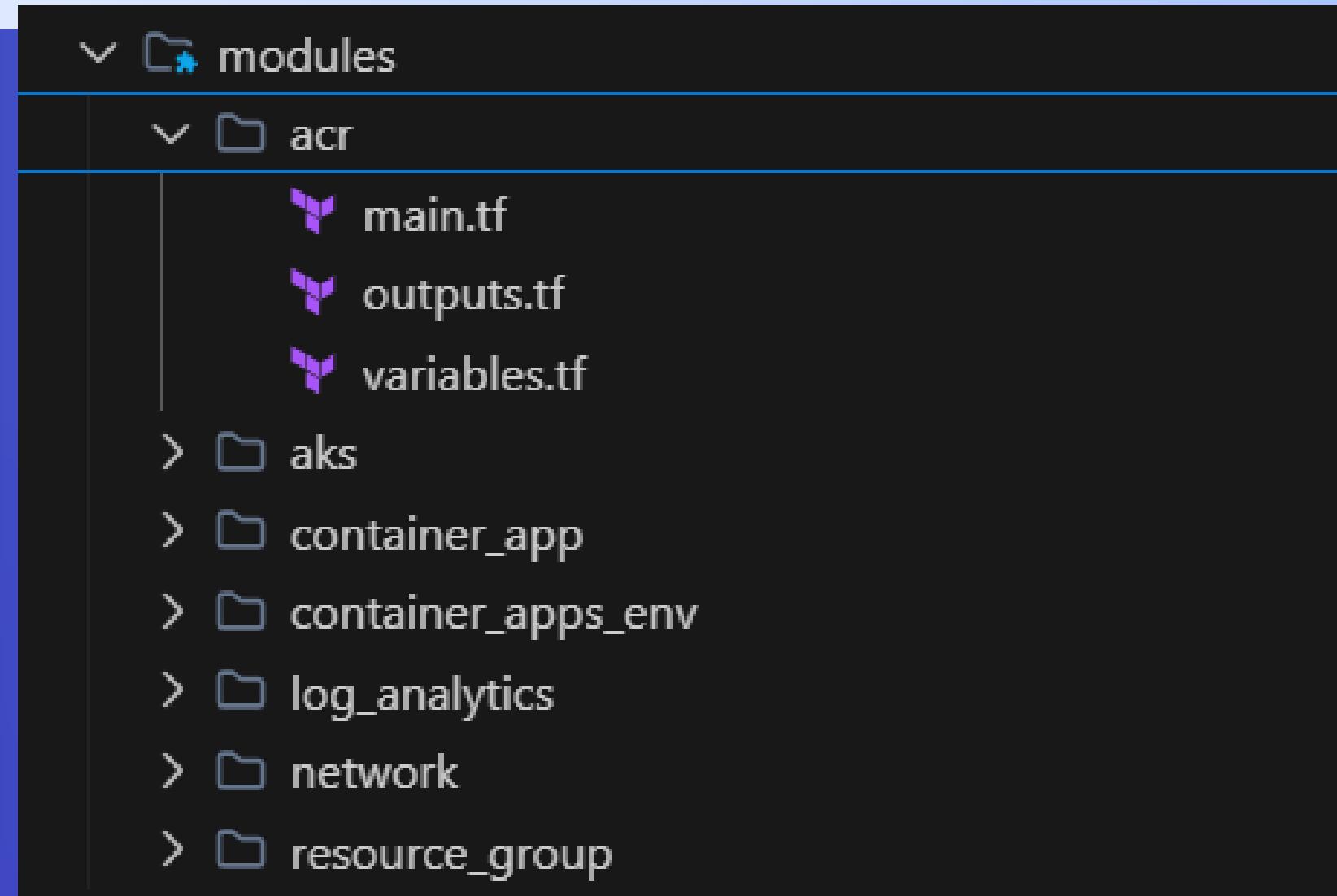


Se realizó una parte en gcp, luego se migró todo a Azure por créditos

Configuración para cada ambiente (dev, stage, prod)



ESTRUCTURA MODULAR



BACKEND REMOTO

 **terraform-state-rg**   

Grupo de recursos

Buscar Buscar

+ Crear  Administrar vista   No agrupar

 **Información general**

 Registro de actividad

 Control de acceso (IAM)

 Etiquetas

 Visualizador de recursos

 Eventos

> Configuración

> Administración de costos

> Supervisión

> Automation

> Ayuda

 **Essentials** 

Recursos Recomendaciones

Filtrar por cualquier ca... 

Tipo es igual que **todo**  Ubicación es igual que **todo** 

+ Agregar filtro

	Nombre ↑	Tipo
<input type="checkbox"/>	tfstateeco2025dev	 Cuenta de almacenamiento
<input type="checkbox"/>	tfstateeco2025prod	 Cuenta de almacenamiento
<input type="checkbox"/>	tfstateeco2025stage	 Cuenta de almacenamiento

INFRAESTRUCTURA EN NUBE (GPC)

Estado de la prueba gratuita: crédito de \$1,164,410.98 y 88 días restantes. Activa tu cuenta completa para obtener acceso ilimitado a todas las funciones de Google Cloud. Usa los créditos restantes y paga solo por lo que uses. [Descartar](#) [Activar](#)

Google Cloud eco-microservices-dev Buscar (/) recursos, documentos, productos y más Buscar ⚡ 📁 🔍 ⓘ S

Cloud Run Servicios Implementar contenedor Conectar repo Escribir una función Actualizar

Descripción general Servicios Trabajos Grupos de trabajadores Asignaciones de dominios

Cada servicio expone un extremo único y ajusta automáticamente la escala de la infraestructura subyacente para controlar las solicitudes entrantes. Implementa una imagen de contenedor, código fuente o función para crear un servicio.

Filtro Filtrar servicios

<input type="checkbox"/>	Nombre ↑	Tipo de implementación	Solicitudes/seg	Región	Autenticación	Ingress	Última implementación
<input type="checkbox"/>	eco-dev-api-gateway	(8) Contenedor	0	us-central1	Permitir sin autenticación	Todas	hace 2 días
<input type="checkbox"/>	eco-dev-cloud-config	(8) Contenedor	0	us-central1	Permitir sin autenticación	Interna	hace 2 días
<input type="checkbox"/>	eco-dev-order-service	(8) Contenedor	0	us-central1	Necesita autenticación	Interna	hace 2 días
<input type="checkbox"/>	eco-dev-product-service	(8) Contenedor	0	us-central1	Necesita autenticación	Interna	hace 2 días
<input type="checkbox"/>	eco-dev-service-discovery	(8) Contenedor	0	us-central1	Permitir sin autenticación	Interna	hace 2 días
<input type="checkbox"/>	eco-dev-user-service	(8) Contenedor	0	us-central1	Necesita autenticación	Interna	hace 2 días
<input type="checkbox"/>	eco-dev-zipkin	(8) Contenedor	0	us-central1	Permitir sin autenticación	Interna	hace 2 días

Notas de versión

INFRAESTRUCTURA EN NUBE (AZURE)

Microsoft Azure

Buscar recursos, servicios y documentos (G+)

Copilot

1084896570@u.icesi.edu.co
UNIVERSIDAD ICESI (ICESLEDU.CO)

Inicio > Centro de Kubernetes | Clústeres > eco-stage-aks

eco-stage-aks | Cargas de trabajo

Servicio de Kubernetes

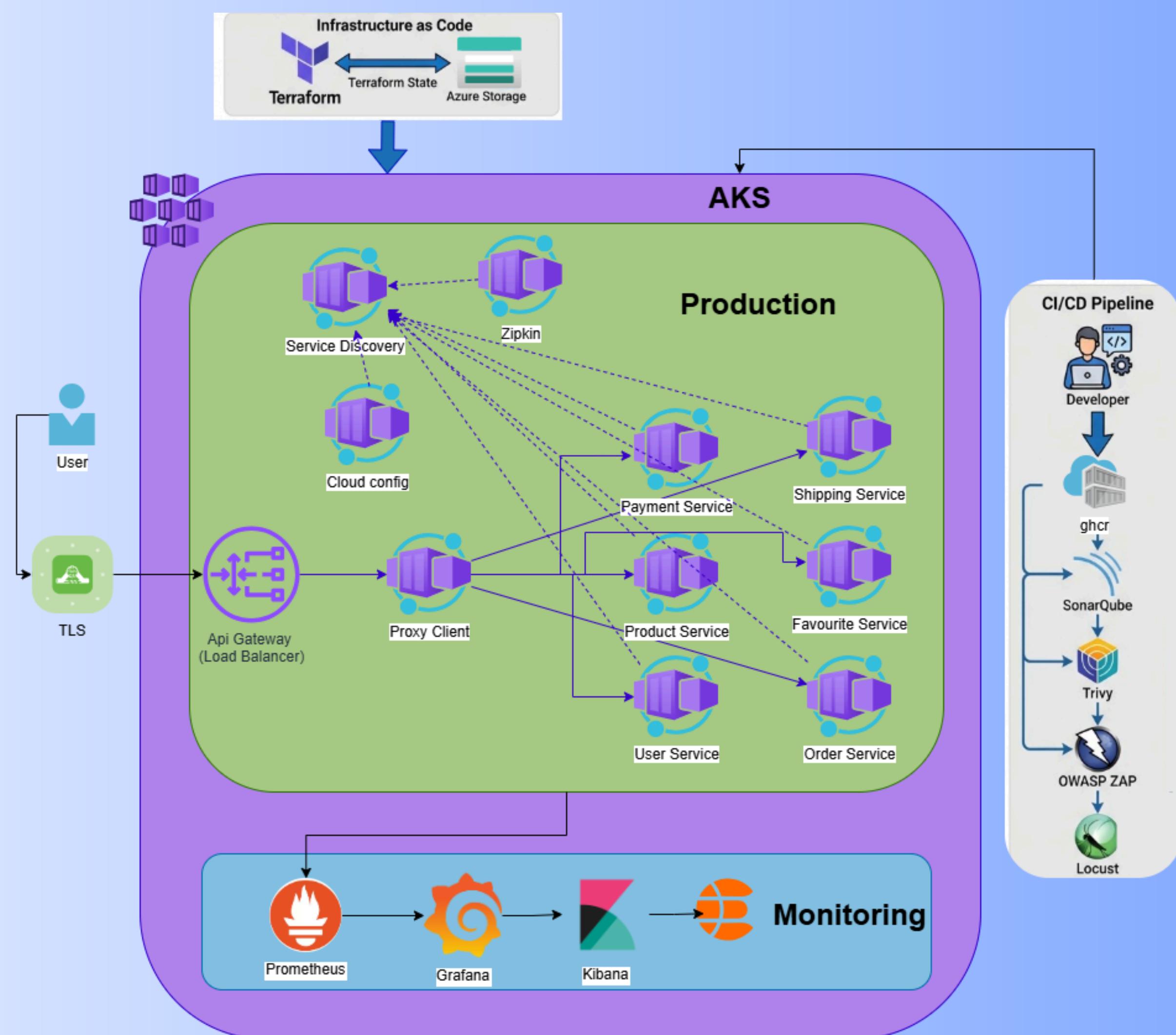
Buscar Crear Escalar Eliminar Actualizar Mostrar etiquetas Enviar comentarios

<input type="checkbox"/>	azure-policy-webhook	kube-system	✓ 1/1	18 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	gatekeeper-audit	gatekeeper-system	✓ 1/1	18 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	gatekeeper-controller	gatekeeper-system	✓ 2/2	18 horas	<div style="width: 100%;">100%</div>	<div style="width: 100%;">100%</div>		
<input type="checkbox"/>	ama-logs-rs	kube-system	✓ 1/1	18 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	api-gateway	eCommerce-stage	✓ 1/1	18 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	cloud-config	eCommerce-stage	✓ 1/1	18 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	favourite-service	eCommerce-stage	✓ 1/1	18 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	order-service	eCommerce-stage	✓ 1/1	18 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	product-service	eCommerce-stage	✓ 1/1	18 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	service-discovery	eCommerce-stage	✓ 1/1	18 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	zipkin	eCommerce-stage	✓ 1/1	18 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	payment-service	eCommerce-stage	✓ 1/1	17 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	proxy-client	eCommerce-stage	✓ 1/1	17 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	shipping-service	eCommerce-stage	✓ 1/1	17 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		
<input type="checkbox"/>	user-service	eCommerce-stage	✓ 1/1	17 horas	<div style="width: 20%;">20%</div>	<div style="width: 20%;">20%</div>		

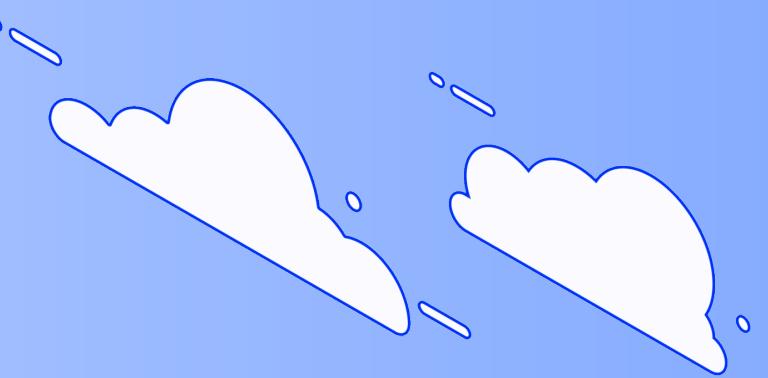
Buscar Recursos y servicios

Información general Registro de actividad Control de acceso (IAM) Etiquetas Supervisar Diagnosticar y solucionar problemas Microsoft Defender for Cloud Análisis de costos Visualizador de recursos Recursos de Kubernetes Espacios de nombres Cargas de trabajo Servicios y entradas Directivas de red Almacenamiento Configuración

DIAGRAMA DE ARQUITECTURA



PATRONES DE DISEÑO



PATRÓN DE MICROSERVICIOS

- El sistema está compuesto por múltiples servicios independientes (user-service, order-service, payment-service, etc.), cada uno con su propia lógica, base de datos y despliegue.

```
> api-gateway
> cloud-config
> favourite-service
> images
> infra\terraform
> k8s
> order-service
> payment-service
> product-service
> proxy-client
> service-discovery
> shipping-service
> src
> tests
> user-service
```

CONTROLLER-SERVICE-REPOSITORY (MVC)

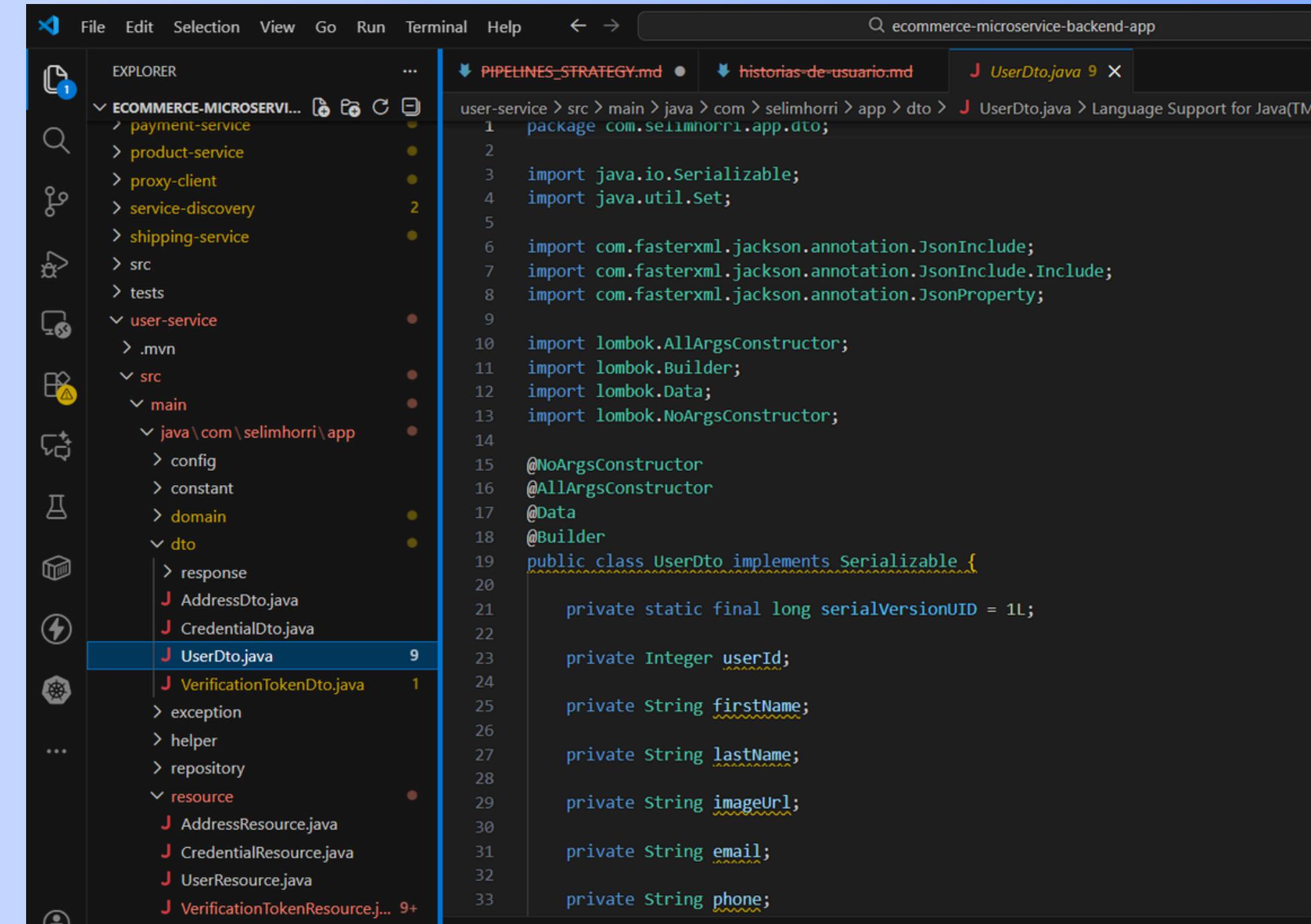
- Controller/Resource: Manejan las peticiones HTTP y delegan la lógica.
 - Service: Implementan la lógica de negocio.
 - Repository: Acceso a datos, generalmente usando Spring Data JPA.

The screenshot shows a Java code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Q ecommerce-microservice-backend-app
- Left Sidebar (EXPLORER):** Shows the project structure under ECOMMERCE-MICROSERVICE-BACKEND-APP:
 - payment-service
 - product-service
 - proxy-client
 - service-discovery
 - shipping-service
 - src
 - tests
 - user-service
 - .mvn
 - src
 - main
 - java\com\selimhorri\app
 - config
 - constant
 - domain
 - dto
 - exception
 - helper
 - repository
 - resource
 - AddressResource.java
 - CredentialResource.java
- Central Area:** The main code editor window displays `VerificationTokenResource.java`. The code implements a REST controller for verification tokens using Spring Boot annotations like `@RestController`, `@RequestMapping`, and `@PostMapping`. It interacts with a `VerificationTokenService` to handle token retrieval and saving.
- Bottom Navigation:** PROBLEMS (94), OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, PLAYWRIGHT.
- Bottom Status Bar:** PowerShell Extension, git checkout circuit, Switched to a new branch 'infra', PS C:\Users\CTecn\Desktop\ecommerce-microservice-backend-app>

DTO (DATA TRANSFER OBJECT)

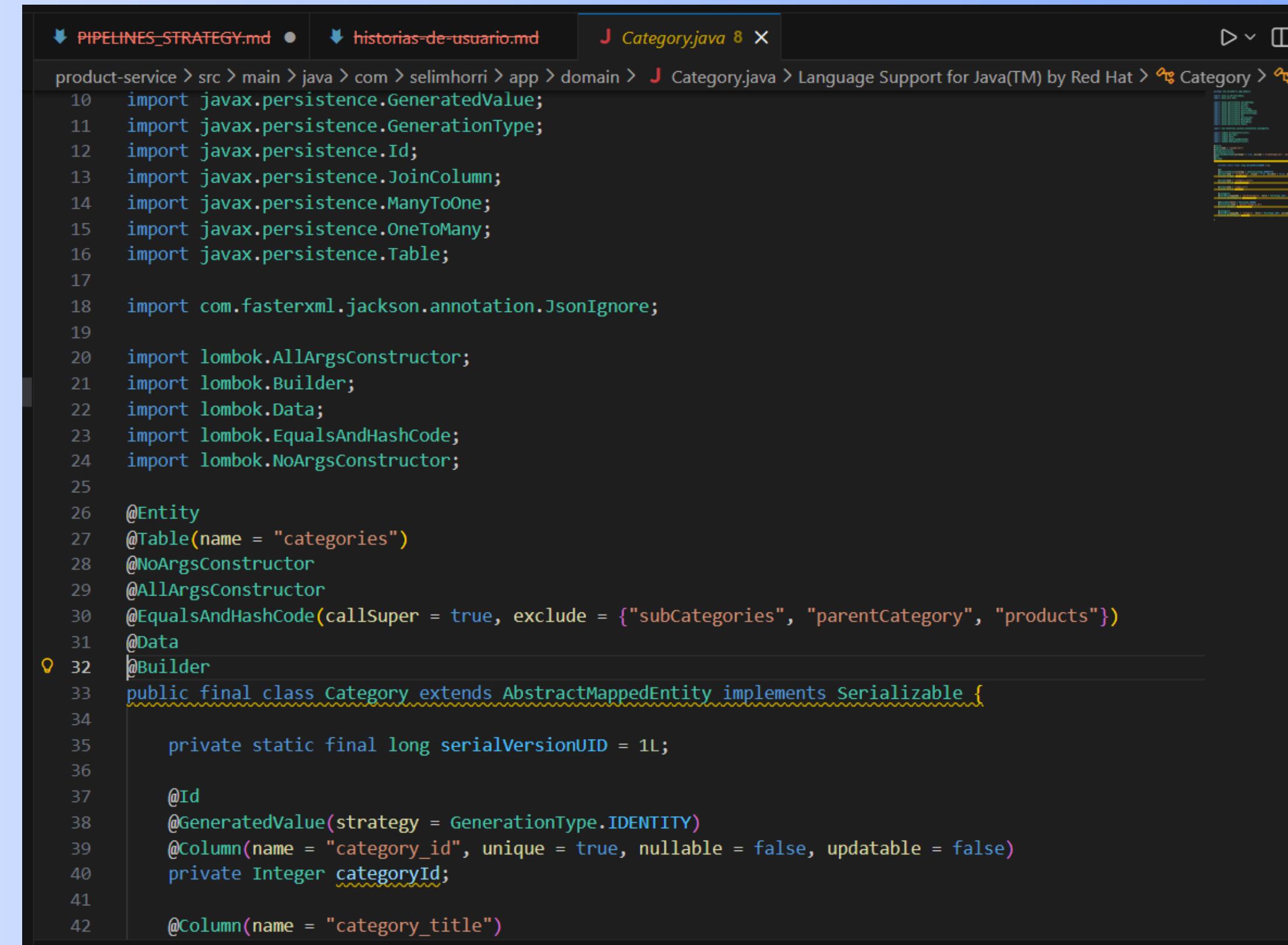
- Clases como OrderDto, FavouriteDto, VerificationTokenDto se usan para transferir datos entre capas y servicios, desacoplando la entidad de dominio de la representación externa.



```
File Edit Selection View Go Run Terminal Help < > ecommerce-microservice-backend-app
EXPLORER
ECOMMERCE-MICROSERV... payment-service product-service proxy-client service-discovery shipping-service src tests user-service .mvn src main java\com\selimhorri\app config constant domain dto response AddressDto.java CredentialDto.java UserDto.java 9 VerificationTokenDto.java 1 exception helper repository resource AddressResource.java CredentialResource.java UserResource.java VerificationTokenResource.j... 9+
PIPELINES-STRATEGY.md historias-de-usuario.md UserDto.java
user-service > src > main > java > com > selimhorri > app > dto > UserDto.java > Language Support for Java(TM)
1 package com.selimhorri.app.dto;
2
3 import java.io.Serializable;
4 import java.util.Set;
5
6 import com.fasterxml.jackson.annotation.JsonInclude;
7 import com.fasterxml.jackson.annotation.JsonInclude.Include;
8 import com.fasterxml.jackson.annotation.JsonProperty;
9
10 import lombok.NoArgsConstructor;
11 import lombok.AllArgsConstructor;
12 import lombok.Data;
13 import lombok.NoArgsConstructor;
14
15 @NoArgsConstructor
16 @AllArgsConstructor
17 @Data
18 @Builder
19 public class UserDto implements Serializable {
20
21     private static final long serialVersionUID = 1L;
22
23     private Integer userId;
24
25     private String firstName;
26
27     private String lastName;
28
29     private String imageUrl;
30
31     private String email;
32
33     private String phone;
```

BUILDER

- Uso de Lombok @Builder en DTOs y entidades para facilitar la creación de objetos complejos de manera fluida y legible.



The screenshot shows a Java code editor with the file `Category.java` open. The code defines a class `Category` that extends `AbstractMappedEntity` and implements `Serializable`. It uses Lombok annotations to generate boilerplate code for construction and equality. The imports at the top include various persistence annotations from `javax.persistence` and Lombok annotations from `com.fasterxml.jackson.annotation` and `lombok`.

```
10 import javax.persistence.GeneratedValue;
11 import javax.persistence.GenerationType;
12 import javax.persistence.Id;
13 import javax.persistence.JoinColumn;
14 import javax.persistence.ManyToOne;
15 import javax.persistence.OneToMany;
16 import javax.persistence.Table;
17
18 import com.fasterxml.jackson.annotation.JsonIgnore;
19
20 import lombok.AllArgsConstructor;
21 import lombok.Builder;
22 import lombok.Data;
23 import lombok.EqualsAndHashCode;
24 import lombok.NoArgsConstructor;
25
26 @Entity
27 @Table(name = "categories")
28 @NoArgsConstructor
29 @AllArgsConstructor
30 @EqualsAndHashCode(callSuper = true, exclude = {"subCategories", "parentCategory", "products"})
31 @Data
32 @Builder
33 public final class Category extends AbstractMappedEntity implements Serializable {
34
35     private static final long serialVersionUID = 1L;
36
37     @Id
38     @GeneratedValue(strategy = GenerationType.IDENTITY)
39     @Column(name = "category_id", unique = true, nullable = false, updatable = false)
40     private Integer categoryId;
41
42     @Column(name = "category_title")
```

EVENT LISTENER

- Uso de `@EventListener` en clases como `DefaultUserConfig` para reaccionar a eventos del ciclo de vida de la aplicación.

```
7
8  @Component
9  @Slf4j
10 @RequiredArgsConstructor
11 <public class DefaultUserConfig {>
12
13     private final UserRepository userRepository;
14     private final CredentialRepository credentialRepository;
15     private final BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
16
17     @EventListener(ApplicationReadyEvent.class)
18     public void init() {
19         // Verificar si ya existe el usuario testuser
20         if (credentialRepository.findByUsername(username: "testuser").isEmpty()) {
21             log.info("*** Creating default testuser ***");
22
23             // Crear credenciales de testuser
24             Credential testCredential = Credential.builder()
25                 .username(username: "testuser")
26                 .password(passwordEncoder.encode("testuser"))
27                 .roleBasedAuthority(RoleBasedAuthority.ROLE_ADMIN)
28                 .isEnabled(isEnabled: true)
29                 .isAccountNonExpired(isAccountNonExpired: true)
30                 .isAccountNonLocked(isAccountNonLocked: true)
31                 .isCredentialsNonExpired(isCredentialsNonExpired: true)
32                 .build();
33
34             // Crear usuario testuser
35             User testUser = User.builder()
36                 .firstName(firstName: "Test")
37                 .lastName(lastName: "User")
38                 .username(username: "testuser")
39                 .password(passwordEncoder.encode("testuser"))
40                 .roleBasedAuthority(RoleBasedAuthority.ROLE_ADMIN)
41                 .isEnabled(isEnabled: true)
42                 .isAccountNonExpired(isAccountNonExpired: true)
43                 .isAccountNonLocked(isAccountNonLocked: true)
44                 .isCredentialsNonExpired(isCredentialsNonExpired: true)
45                 .build();
46
47             userRepository.save(testUser);
48             credentialRepository.save(testCredential);
49         }
50     }
51 }
```

EVENT LISTENER

- Uso de `@EventListener` en clases como `DefaultUserConfig` para reaccionar a eventos del ciclo de vida de la aplicación.

```
7
8  @Component
9  @Slf4j
10 @RequiredArgsConstructor
11 <public class DefaultUserConfig {
```

```
12
13     private final UserRepository userRepository;
14     private final CredentialRepository credentialRepository;
15     private final BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
```

```
16
17     @EventListener(ApplicationReadyEvent.class)
18     public void init() {
19         // Verificar si ya existe el usuario testuser
20         if (credentialRepository.findByUsername(username: "testuser").isEmpty()) {
21             log.info("*** Creating default testuser ***");
22
23             // Crear credenciales de testuser
24             Credential testCredential = Credential.builder()
25                 .username(username: "testuser")
26                 .password(passwordEncoder.encode("testuser"))
27                 .roleBasedAuthority(RoleBasedAuthority.ROLE_ADMIN)
28                 .isEnabled(isEnabled: true)
29                 .isAccountNonExpired(isAccountNonExpired: true)
30                 .isAccountNonLocked(isAccountNonLocked: true)
31                 .isCredentialsNonExpired(isCredentialsNonExpired: true)
32                 .build();
33
34             // Crear usuario testuser
35             User testUser = User.builder()
36                 .firstName(firstName: "Test")
```

PATRÓN DE EXCEPCIÓN/RESPUESTA UNIFICADA

- Clases como ExceptionMsg centralizan la estructura de los mensajes de error, siguiendo el patrón de objeto de error.

The screenshot shows a Java code editor with the following details:

- Project Structure:** ECOMMERCE-MICROSERVICE-BACKEND-APP > user-service > src > main > java > com > selimhorri > app > exception > ApiExceptionHandler.java
- Code Content:** The code defines a class `ApiExceptionHandler` with two `@ExceptionHandler` annotations. The first handles `BindException` and logs validation errors. The second handles `RuntimeException` and logs API request errors.

```
25  @RequiredArgsConstructor
26  public class ApiExceptionHandler {
27
28      @ExceptionHandler(value = {
29          MethodArgumentNotValidException.class,
30          HttpMessageNotReadableException.class
31      })
32      public <T extends BindException> ResponseEntity<ExceptionMsg> handleValidationException(T e) {
33
34          log.info("**ApiExceptionHandler controller, handle validation exception*\n");
35          final var badRequest = HttpStatus.BAD_REQUEST;
36
37          return new ResponseEntity<>(
38              ExceptionMsg.builder()
39                  .msg("*" + e.getBindingResult().getFieldError().getDefaultMessage() + "!")
40                  .httpStatus(badRequest)
41                  .timestamp(ZonedDateTime.now(ZoneId.systemDefault()))
42                  .build(), badRequest);
43
44      }
45
46      @ExceptionHandler(value = {
47          UserObjectNotFoundException.class,
48          CredentialNotFoundException.class,
49          VerificationTokenNotFoundException.class,
50          AddressNotFoundException.class
51      })
52      public <T extends RuntimeException> ResponseEntity<ExceptionMsg> handleApiRequestException(T e) {
53
54          log.info("**ApiExceptionHandler controller, handle API request*\n");
55          final var badRequest = HttpStatus.BAD_REQUEST;
56
57          return new ResponseEntity<>(
58              ExceptionMsg.builder()
59                  .msg("An error occurred while processing your request")
60                  .httpStatus(badRequest)
61                  .timestamp(ZonedDateTime.now(ZoneId.systemDefault()))
62                  .build(), badRequest);
63      }
64  }
```

- Editor Tools:** The bottom navigation bar includes PROBLEMS (175), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), PORTS, PLAYWRIGHT, and PowerShell Extension.

MAPPING HELPER (MAPPER)

- Clases como `UserMappingHelper`, `OrderMappingHelper` implementan el patrón Mapper para convertir entre entidades y DTOs.

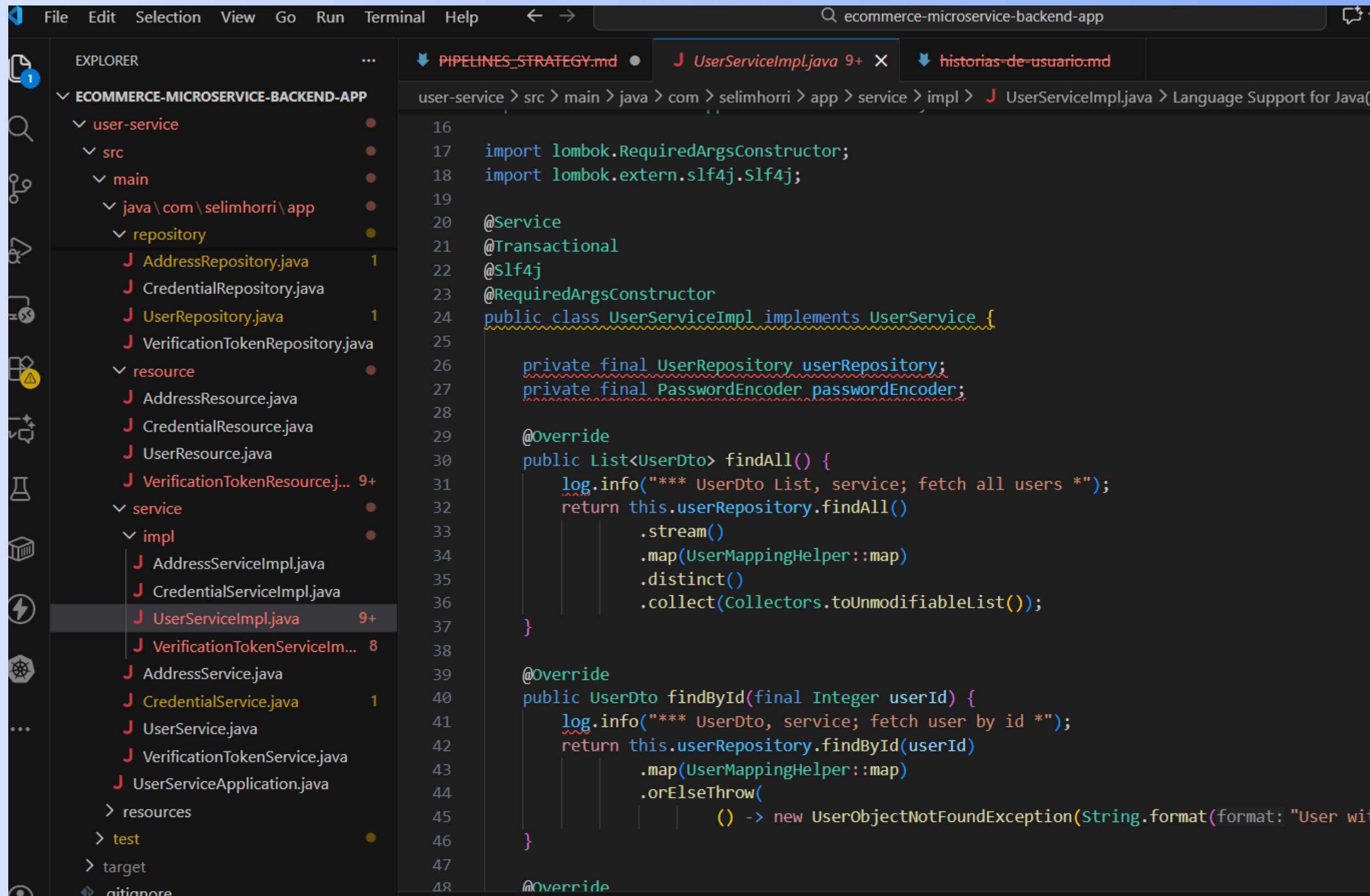
The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure under "ECOMMERCE-MICROSERVICES". The "user-service" folder contains "src", "main", "java\com\selimhorri\app", "config", "constant", "domain", "dto", "exception", "payload", "wrapper", "helper", "repository", and several Java files like PasswordEncoderConfig.java, AddressDto.java, CredentialDto.java, UserDto.java, VerificationTokenDto.java, ApiExceptionHandler.java, and others.
- EDITOR:** The "AddressMappingHelper.java" file is open. The code implements the `AddressMappingHelper` interface, which has two static methods: `map(Address)` and `map(AddressDto)`. Both methods return an `AddressDto` object built from the input entity using `AddressDto.builder()`.
- STATUS BAR:** Shows the file path: "user-service > src > main > java > com > selimhorri > app > helper > AddressMappingHelper.java".

```
package com.selimhorri.app.helper;
import com.selimhorri.app.domain.Address;
import com.selimhorri.app.domain.User;
import com.selimhorri.app.dto.AddressDto;
import com.selimhorri.app.dto.UserDto;
public interface AddressMappingHelper {
    public static AddressDto map(final Address address) {
        return AddressDto.builder()
            .addressId(address.getAddressId())
            .fullAddress(address.getFullAddress())
            .postalCode(address.getPostalCode())
            .city(address.getCity())
            .userDto(
                UserDto.builder()
                    .userId(address.getUser().getUserId())
                    .firstName(address.getUser().getFirstName())
                    .lastName(address.getUser().getLastName())
                    .imageUrl(address.getUser().getImageUrl())
                    .email(address.getUser().getEmail())
                    .phone(address.getUser().getPhone())
                    .build())
            .build();
    }
    public static Address map(final AddressDto addressDto) {
        return Address.builder()
            .addressId(addressDto.getAddressId())
            .fullAddress(addressDto.getFullAddress())
            .postalCode(addressDto.getPostalCode())
            .city(addressDto.getCity());
    }
}
```

INYECCIÓN DE DEPENDENCIAS

- Uso extensivo de la inyección de dependencias de Spring (@Autowired, @RequiredArgsConstructor).



The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows the project structure under "ECOMMERCE-MICROSERVICE-BACKEND-APP". The "user-service" module contains "src/main/java/com/selimhorri/app" which includes "repository" and "resource" packages, and "service/impl" package. "repository" contains AddressRepository.java, CredentialRepository.java, UserRepository.java, and VerificationTokenRepository.java. "resource" contains AddressResource.java, CredentialResource.java, UserResource.java, and VerificationTokenResource.java. "service/impl" contains AddressServiceImpl.java, CredentialServiceImpl.java, UserServiceImpl.java (highlighted in red), and VerificationTokenServiceimpl.java.
- Code Editor:** Displays the "UserServiceImpl.java" file. The code uses Lombok annotations like @Service, @Transactional, and @Slf4j. It imports RequiredArgsConstructor and slf4j. The class implements UserService and overrides findAll() and findById(). Both methods log.info() calls and return results from userRepository.
- Search Bar:** Shows the search term "ecommerce-microservice-backend-app".
- Top Bar:** Includes File, Edit, Selection, View, Go, Run, Terminal, Help, and navigation icons.

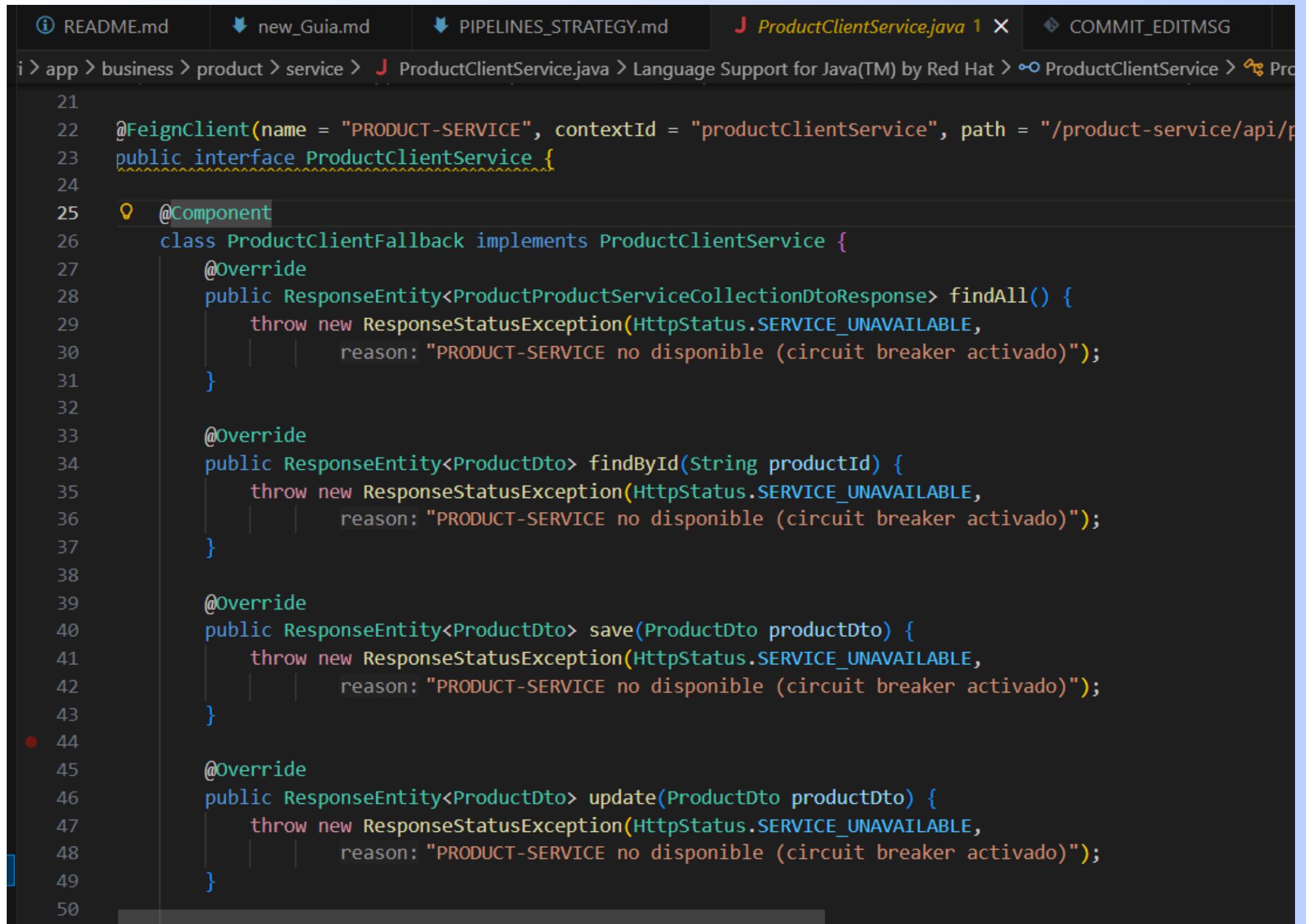
```
File Edit Selection View Go Run Terminal Help ← → 🔍 ecommerce-microservice-backend-app

EXPLORER
ECOMMERCE-MICROSERVICE-BACKEND-APP
  user-service
    src
      main
        java\com\selimhorri\app
          repository
            AddressRepository.java
            CredentialRepository.java
            UserRepository.java
            VerificationTokenRepository.java
          resource
            AddressResource.java
            CredentialResource.java
            UserResource.java
            VerificationTokenResource.j...
          service
            impl
              AddressServiceImpl.java
              CredentialServiceImpl.java
              J UserServiceImpl.java 9+
              VerificationTokenService... 8
              AddressService.java
              CredentialService.java 1
              UserService.java
              VerificationTokenService.java
              UserServiceApplication.java
            resources
            test
            target
            .gitignore

PIPELINES_STRATEGY.md • J UserServiceImpl.java 9+ X historias-de-usuario.md
user-service > src > main > java > com > selimhorri > app > service > impl > J UserServiceImpl.java > Language Support for Java(16
17 import lombok.RequiredArgsConstructor;
18 import lombok.extern.slf4j.Slf4j;
19
20 @Service
21 @Transactional
22 @Slf4j
23 @RequiredArgsConstructor
24 public class UserServiceImpl implements UserService {
25
26     private final UserRepository userRepository;
27     private final PasswordEncoder passwordEncoder;
28
29     @Override
30     public List<UserDto> findAll() {
31         log.info("### UserDto List, service; fetch all users *");
32         return this.userRepository.findAll()
33             .stream()
34             .map(UserMappingHelper::map)
35             .distinct()
36             .collect(Collectors.toUnmodifiableList());
37     }
38
39     @Override
40     public UserDto findById(final Integer userId) {
41         log.info("### UserDto, service; fetch user by id *");
42         return this.userRepository.findById(userId)
43             .map(UserMappingHelper::map)
44             .orElseThrow(
45                 () -> new UserObjectNotFoundException(String.format("User wi
46     }
47
48     @Override
```

PATRONES MEJORADOS

CIRCUIT BRAKER



The screenshot shows a code editor with a dark theme. The file being edited is `ProductClientService.java`. The code implements a `ProductClientService` interface using FeignClient and includes a fallback class (`ProductClientFallback`) that throws a `ResponseStatusException` with a reason message indicating the circuit breaker has been activated. The code is annotated with `@Override` for each method.

```
1 21
2 22 @FeignClient(name = "PRODUCT-SERVICE", contextId = "productClientService", path = "/product-service/api/p
3 23 public interface ProductClientService {
4
5 24
6 25     @Component
7     class ProductClientFallback implements ProductClientService {
8
9 26         @Override
10    public ResponseEntity<ProductProductServiceCollectionDtoResponse> findAll() {
11        throw new ResponseStatusException(HttpStatus.SERVICE_UNAVAILABLE,
12            reason: "PRODUCT-SERVICE no disponible (circuit breaker activado)");
13    }
14
15    @Override
16    public ResponseEntity<ProductDto> findById(String productId) {
17        throw new ResponseStatusException(HttpStatus.SERVICE_UNAVAILABLE,
18            reason: "PRODUCT-SERVICE no disponible (circuit breaker activado)");
19    }
20
21    @Override
22    public ResponseEntity<ProductDto> save(ProductDto productDto) {
23        throw new ResponseStatusException(HttpStatus.SERVICE_UNAVAILABLE,
24            reason: "PRODUCT-SERVICE no disponible (circuit breaker activado)");
25    }
26
27    @Override
28    public ResponseEntity<ProductDto> update(ProductDto productDto) {
29        throw new ResponseStatusException(HttpStatus.SERVICE_UNAVAILABLE,
30            reason: "PRODUCT-SERVICE no disponible (circuit breaker activado)");
31    }
32
33    @Override
34    public ResponseEntity<ProductDto> delete(String productId) {
35        throw new ResponseStatusException(HttpStatus.SERVICE_UNAVAILABLE,
36            reason: "PRODUCT-SERVICE no disponible (circuit breaker activado)");
37    }
38
39    @Override
40    public ResponseEntity<ProductDto> patch(ProductPatchRequest productPatchRequest) {
41        throw new ResponseStatusException(HttpStatus.SERVICE_UNAVAILABLE,
42            reason: "PRODUCT-SERVICE no disponible (circuit breaker activado)");
43    }
44
45    @Override
46    public ResponseEntity<ProductDto> patch(String productId, ProductPatchRequest productPatchRequest) {
47        throw new ResponseStatusException(HttpStatus.SERVICE_UNAVAILABLE,
48            reason: "PRODUCT-SERVICE no disponible (circuit breaker activado)");
49    }
50}
```

Evita que un servicio siga llamando a otro servicio externo cuando este último está fallando de forma continua. Funciona igual que un fusible eléctrico:

BULKHEAD

```
resilience4j:  
  bulkhead:  
    instances:  
      productServiceBulkhead:  
        maxConcurrentCalls: 50  
        maxWaitDuration: 100ms
```

```
@GetMapping("/{productId}")  
@Bulkhead(name = "productServiceBulkhead",  
fallbackMethod = "findByIdFallback")  
public ResponseEntity<ProductDto> findById  
(@PathVariable("productId") final String productId)  
{  
  log.info("** Proxy Client: Fetching product {}  
with Bulkhead protection **", productId);  
  return ResponseEntity.ok(this.  
productClientService.findById(productId).getBody  
());  
}
```

Aísla los recursos utilizados por diferentes partes de la aplicación para evitar que un fallo en una parte agote todos los recursos del sistema (como hilos de ejecución), afectando a otras funcionalidades. Se inspira en los compartimentos estancos de los barcos

FEATURE TOGGLE

Apagado

```
Content : {  
    "collection" : [ {  
        "productId" : 1,  
        "productTitle" : "asus",  
        "imageUrl" : "xxx",  
        "sku" : "dfqejklejrkn",  
        "priceUnit" : 0.0,  
        "quantity" : 50,  
        "category" : "laptop",  
        "status" : "available"  
    } ]  
}
```

Encendido

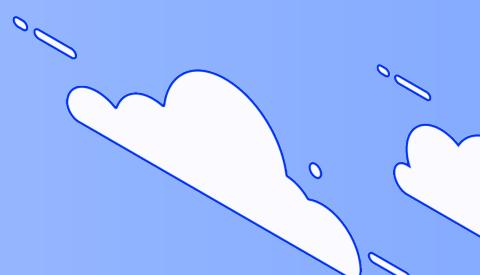
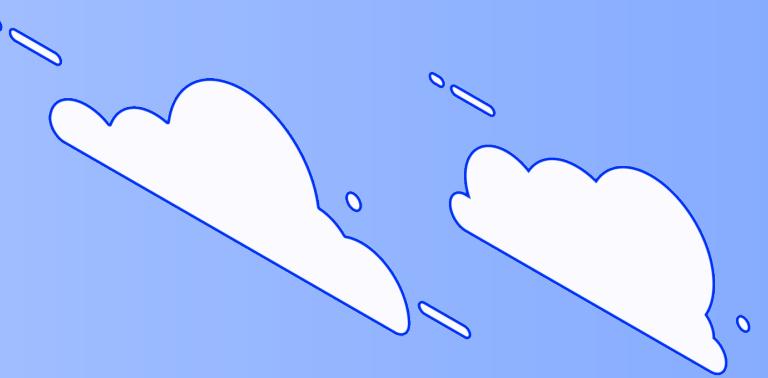
```
< Date: Sun, 23 Nov 2025 21:11:40 GMT  
<  
{"message": "Esta es la versión V2 del endpoint de productos", "data":  
  "version": "V2", "status": "ok", "products": [{"id": 1, "title": "Laptop Asus", "image": "xxx", "sku": "dfqejklejrkn", "unit": 0.0, "quantity": 50, "category": "laptop", "status": "available"}]}>  
  connection #0 to host localhost:8900 left intact
```

Comando

```
kubectl set env deployment/proxy-client FEATURES_ENABLE_V2_ENDPOINT=true -n ecommerce
```

Permite modificar el comportamiento del sistema en tiempo de ejecución sin necesidad de cambiar el código ni volver a desplegar una nueva versión de la aplicación. Funciona como un interruptor de luz.

CI/CD AVANZADO



PIPELINES CI/CD

Dev

← Build & Push - Proxy Client

fix: update status message in AuthenticationController for clarity an...

Summary

Triggered via push 3 days ago
Nicolas-CM pushed → 372d17b dev

build-proxy-client.yml
on: push

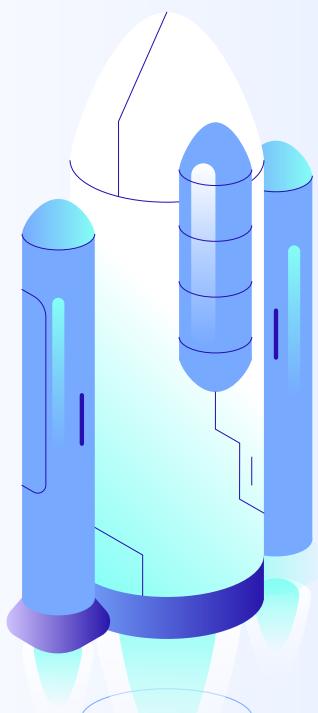
build-and-push 51s

Jobs

- ✓ build-and-push
- ✓ Integration Tests - favourite-service
- ✓ Integration Tests - payment-service
- ✓ Integration Tests - user-service
- ✓ Unit Tests - favourite-service
- ✓ Unit Tests - order-service
- ✓ Unit Tests - payment-service
- ✓ Unit Tests - product-service
- ✓ Unit Tests - shipping-service
- ✓ Unit Tests - user-service

← Infrastructure DEV - Terraform

✓ Merge pull request #15 from Nicolas-CM/infra #20



← DEV - Tests & Deploy

fix: update status message in AuthenticationController for clarity an... #28

Summary

Triggered via push 3 days ago
Nicolas-CM pushed → 372d17b dev

Status Success Total duration Artifacts

dev.yml
on: push

Matrix: Unit Tests
6 jobs completed Show all jobs

Matrix: Integration Tests
3 jobs completed Show all jobs

Deploy to AKS dev 3m

```
graph LR; A[Matrix: Unit Tests] --- B[Matrix: Integration Tests]; B --- C[Deploy to AKS dev]
```

← Infrastructure DEV - Terraform

✓ Merge pull request #15 from Nicolas-CM/infra #20

Summary

Re-run triggered 4 days ago
Nicolas-CM → a686f0a dev

Status Success Total duration Artifacts

infra-dev-terraform.yml
on: push

Terraform Plan 40s

Terraform Apply 1m 39s

← CI-CD Stage - Full Validation & Deploy

feat: Enhance CI/CD workflow with AKS deployment, E2E tests, and perf... #20

[Re-run all jobs](#)

[Summary](#)

Triggered via push 19 minutes ago

Nicolas-CM pushed → d81716e stage

Status: Success | Total duration: 8m 1s | Artifacts: 2

ci-cd-stage.yml
on: push

Deploy to AKS Stage (Helm) | E2E Tests | Performance Tests | OWASP ZAP Security Scan | Notify Failure

Jobs:
 ✓ Deploy to AKS Stage (Helm)
 ✓ E2E Tests
 ✓ Performance Tests
 ✓ OWASP ZAP Security Scan
 ⚡ Notify Failure

Run details | Usage | Workflow file

PIPELINES

CI/CD

← Infrastructure STAGE - Terraform

Merge pull request #18 from Nicolas-CM/dev #3

[Summary](#)

Triggered via push 3 days ago

Nicolas-CM pushed → ba4ba41 stage

Status: Success | Total duration: 2m 27s | Artifacts: 1

infra-stage-terraform.yml
on: push

Terraform Plan | Terraform Apply

Jobs:
 ✓ Terraform Plan
 ✓ Terraform Apply

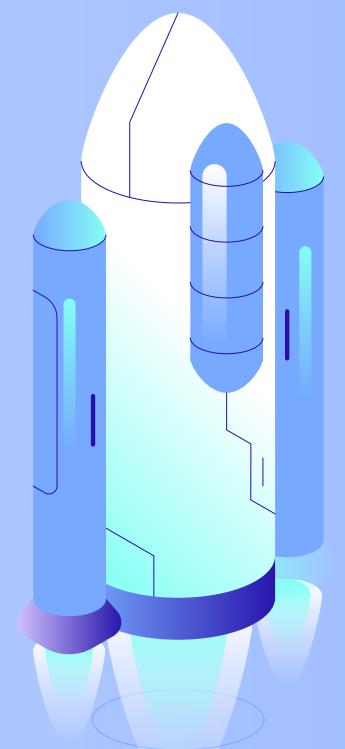
Run details | Usage | Workflow file

Terraform Plan | Terraform Apply

Artifacts
Produced during runtime

Name

- cube e2e-test-results
- cube performance-test-results
- cube zap-scan-results



Stage

PIPELINES CI/CD

Prod

Triggered via push 3 minutes ago

Nicolas-CM pushed -o- [d5d21b4](#) master

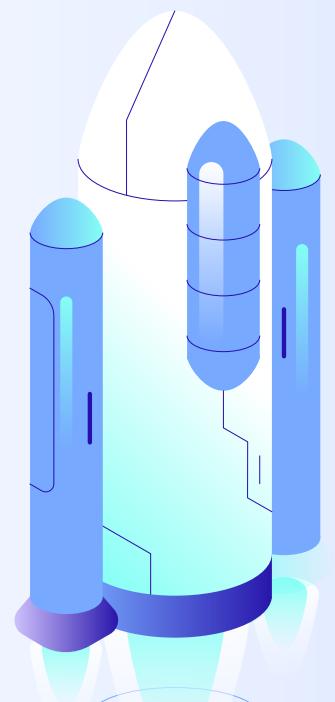
Status: Success | Total duration: 2m 28s | Artifacts: 1

ci-cd-prod.yml
on: push

Request Production Approval → Deploy to AKS Prod (Helm) → Generate Production Release → Notify Success

Run details

```
graph LR; A[Request Production Approval] -- "6s" --> B[Deploy to AKS Prod (Helm)]; B -- "40s" --> C[Generate Production Rel...]; C -- "6s" --> D[Notify Success];
```



Terraform Plan

Terraform Apply

Run details

Usage

Workflow file

infra-prod-terraform.yml
on: push

Terraform Plan → Terraform Apply

```
graph LR; A[Terraform Plan] -- "46s" --> B[Terraform Apply];
```

NOTIFICACIONES

✖ Dev Pipeline Failed or Cancelled: Nicolas-CM/ecommerce

GitHub Actions <nicolascuellar.molina2@gmail.com>
para mí ▾

The dev pipeline for Nicolas-CM/ecommerce-microservice-backend-app has failed or was cancelled.

Repository: Nicolas-CM/ecommerce-microservice-backend-app

Commit: 4a17169e0f9c9cb8f8a9ba4ffeab0e05867e883e

Author: Nicolas-CM

Branch: dev

Check the logs here: <https://github.com/Nicolas-CM/ecommerce-microservice-backend-app/actions/runs/19800730815>

Environments / Configure prod

Deployment protection rules

Configure reviewers, timers, and custom rules that must pass before deployments to this environment can proceed.

Required reviewers

Specify people or teams that may approve workflow runs when they access this environment.

Add up to 5 more reviewers

Search for people or teams...



Nicolas-CM

X

Production deployment is awaiting approval.

Repository : Nicolas-CM/ecommerce-microservice-backend-app

Branch : master

Commit : 7e83bfff121a60508ebb7fd69abb8adc512d7dce

Workflow : CI-CD Prod - Full Validation & Deploy

Run : <https://github.com/Nicolas-CM/ecommerce-microservice-backend-app/actions/runs/19800730815>

Microservice image tags for production:

Values override for PRODUCTION environment

imageTags:

cloud-config: "0.2.0"

service-discovery: "0.2.0"

api-gateway: "0.0.1"

product-service: "0.2.0"

order-service: "0.2.0"

user-service: "0.2.0"

payment-service: "0.2.0"

shipping-service: "0.2.0"

favourite-service: "0.2.0"

proxy-client: "0.0.1"

Environments

New environment

You can configure environments with protection rules, variables, and secrets. [Learn more about configuring environments.](#)

prod

2 protection rules

5 secrets



stage

5 secrets



dev

5 secrets



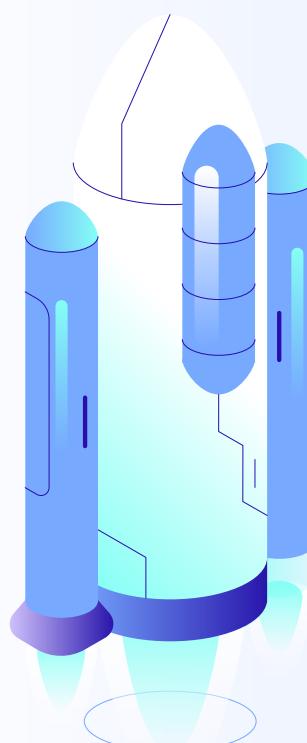
Responder

Reenviar



DOMINIO

cuellarapp.online



DNS CHECK

cuellarapp.online A Search

CD Flag Refresh: 20 sec.

Location	IP Addresses	Status
San Francisco CA, United States OpenDNS	76.223.105.230 172.168.124.212 13.248.243.5	✓
Mountain View CA, United States Google	172.168.124.212 76.223.105.230 13.248.243.5	✓
Berkeley, US Quad9	13.248.243.5 172.168.124.212 76.223.105.230	✓
New York, United States Oracle Corporation	172.168.124.212 76.223.105.230 13.248.243.5	✓
Kansas City, United States WholeSale Internet, Inc.	76.223.105.230 172.168.124.212 13.248.243.5	✓
United States CenturyLink	13.248.243.5 172.168.124.212 76.223.105.230	✓

CHECK DNS PROPAGATION

Whether you have recently changed your DNS records, switched web host, or started a new website - checking whether the DNS records are propagated globally is essential. DNS Checker provides a free DNS propagation check service to check Domain Name System records against a selected list of DNS servers in multiple regions worldwide. Perform a quick DNS propagation lookup for any hostname or domain, and check DNS data collected from all available DNS Servers to confirm that the DNS records are fully propagated.

DNS Propagation Map by DNSChecker.org



PRUEBAS

PRUEBAS UNITARIAS

- J UserServiceApplicationTests.java
- J OrderServiceApplicationTests.java
- J ProductServiceApplicationTests.java
- J PaymentServiceApplicationTests.java
- J ApiGatewayApplicationTests.java

```
[INFO] [INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[INFO] -----
[INFO] Reactor Summary for user-service 0.1.0:
[INFO]
[INFO] user-service ..... SUCCESS [ 12.868 s]
[INFO] user-service ..... SUCCESS [ 12.868 s]
[INFO] product-service ..... SUCCESS [ 5.161 s]
[INFO] favourite-service ..... SUCCESS [ 5.488 s]
[INFO] order-service ..... SUCCESS [ 4.649 s]
[INFO] shipping-service ..... SUCCESS [ 5.316 s]
[INFO] payment-service ..... SUCCESS [ 5.038 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 39.480 s
[INFO] Finished at: 2025-11-01T20:40:39-05:00
[INFO] -----
```

```
↓ architecture.copilotmd U J UserServiceApplicationTests.java 9+ X
user-service > src > test > java > com > selimhorri > app > J UserServiceApplicationTests.java > Language Support for Java(TM)
27 import com.selimhorri.app.service.impl.userServiceImpl;
28
29 @ExtendWith(MockitoExtension.class)
30 class UserServiceApplicationTests {
31
32     @Mock
33     private UserRepository userRepository;
34
35     @Mock
36     private PasswordEncoder passwordEncoder;
37
38     @InjectMocks
39     private UserServiceImpl userService;
40
41     private UserDto testUserDto;
42     private User testUser;
43
44     @BeforeEach
45     void setup() {
46         // Preparar datos de prueba
47         CredentialDto credentialDto = CredentialDto.builder()
48             .username(username: "testuser")
49             .password(password: "plainPassword123")
50             .roleBasedAuthority(RoleBasedAuthority.ROLE_USER)
51             .isEnabled(isEnabled: true)
52             .isAccountNonExpired(isAccountNonExpired: true)
53             .isAccountNonLocked(isAccountNonLocked: true)
54             .isCredentialsNonExpired(isCredentialsNonExpired: true)
55             .build();
56
57         AddressDto addressDto = AddressDto.builder()
58             .fullAddress(fullAddress: "123 Test Street")
59             .postalCode(postalCode: "12345")
```

PRUEBAS DE INTEGRACIÓN

```
PS C:\Users\CTecn\Desktop\ecommerce-microservice-backend-app> mvn -Pintegration verify
[INFO] --- failsafe:2.22.2:integration-test (default) @ payment-service ---
[INFO] --- failsafe:2.22.2:verify (default) @ payment-service ---
[INFO] -----
[INFO] Reactor Summary for ecommerce-microservice-backend 0.1.0:
[INFO]
[INFO] ecommerce-microservice-backend ..... SUCCESS [ 58.799 s]
[INFO] service-discovery ..... SUCCESS [ 6.940 s]
[INFO] cloud-config ..... SUCCESS [ 3.566 s]
[INFO] api-gateway ..... SUCCESS [ 3.342 s]
[INFO] proxy-client ..... SUCCESS [ 4.397 s]
[INFO] user-service ..... SUCCESS [ 6.711 s]
[INFO] product-service ..... SUCCESS [ 6.276 s]
[INFO] favourite-service ..... SUCCESS [ 6.492 s]
[INFO] order-service ..... SUCCESS [ 6.430 s]
[INFO] shipping-service ..... SUCCESS [ 6.891 s]
[INFO] payment-service ..... SUCCESS [ 7.357 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
```

- [J UserControllerIntegrationTest.java](#)
- [J FavouriteServiceIntegrationTest.java](#)
- [J PaymentServiceIntegrationTest.java](#)
- [J ShippingServiceIntegrationTest.java](#)

```
↓ architecture.copilotmd U J UserControllerIntegrationTest.java 9+ X
user-service > src > test > java > com > selimhorri > app > integration > J UserControllerIntegrationTest.java > Language Support
33 class UserControllerIntegrationTest {
34
35     @Autowired
36     private MockMvc mockMvc;
37
38     @Autowired
39     private ObjectMapper objectMapper;
40
41     private UserDto testUserDto;
42
43     @BeforeEach
44     void setUp() {
45         // Setup test data
46         Set<AddressDto> addressDtos = new HashSet<>();
47         AddressDto addressDto = AddressDto.builder()
48             .fullAddress(fullAddress: "123 Test Street")
49             .postalCode(postalCode: "12345")
50             .city(city: "Test City")
51             .build();
52         addressDtos.add(addressDto);
53
54         CredentialDto credentialDto = CredentialDto.builder()
55             .username(username: "integrationuser")
56             .password(password: "password123")
57             .roleBasedAuthority(RoleBasedAuthority.ROLE_USER)
58             .isEnabled(isEnabled: true)
59             .isAccountNonExpired(isAccountNonExpired: true)
60             .isAccountNonLocked(isAccountNonLocked: true)
61             .isCredentialsNonExpired(isCredentialsNonExpired: true)
62             .build();
63
64         testUserDto = UserDto.builder()
```

PRUEBAS E2E

Light Dark

Summary Total Requests 21 Failed Tests 0 Skipped Tests 0

Newman Run Dashboard

Tuesday, 04 November 2025 23:52:03

TOTAL ITERATIONS 1 

TOTAL ASSERTIONS 42 

TOTAL FAILED TESTS 0 

TOTAL SKIPPED TESTS 0 

FILE INFORMATION

Collection: E-Commerce Microservices E2E Tests

COLLECTION DESCRIPTION

End-to-End tests for microservices architecture validating complete user flows through API Gateway

TIMINGS AND DATA

⌚ Total run duration: 3.8s
📊 Total data received: 14.64KB
⌚ Average response time: 97ms

SUMMARY ITEM	TOTAL	FAILED
Requests	21	0
Prerequest Scripts	2	0
Test Scripts	21	0
Assertions	42	0
Skipped Tests	0	-

PRUEBAS RENDIMIENTO

LOCUST

Host: http://127.0.0.1:18080 | Status: RUNNING | Users: 75 | RPS: 36.9 | Failures: 0% | EDIT | STOP | RESET |

STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA ! LOGS LOCUST CLOUD

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	GET /favourite-service/api/favourites	1522	0	32	180	790	62.85	22	2053	1589	7.6	0
GET	GET /payment-service/api/payments	1517	0	24	78	440	38.25	16	920	967	7.9	0
GET	GET /product-service/api/products	1520	0	11	41	180	18.66	7	945	1055	7.8	0
GET	GET /product-service/api/products/{id}	1520	0	10	43	230	18.76	7	939	236	6.4	0
GET	GET /shipping-service/api/shippings	1520	0	38	180	690	65.91	26	1151	1631	7.2	0
Aggregated		7599	0	25	99	510	40.89	7	2053	1095.78	36.9	0

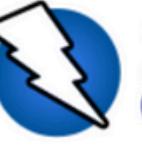
0 fallos en 7.599 requests → excelente estabilidad.

RPS: 36.9 → para pruebas iniciales está bien; no indica saturación.

Latencias muy sanas:
Median: 25 ms

95th percentile: 99 ms
99th percentile: 510 ms

PRUEBAS DE SEGURIDAD

 ZAP by Checkmarx ZAP Scanning Report

Site: <http://68.211.65.139:8080>

Generated on Sat, 29 Nov 2025 17:46:26

ZAP Version: 2.16.1

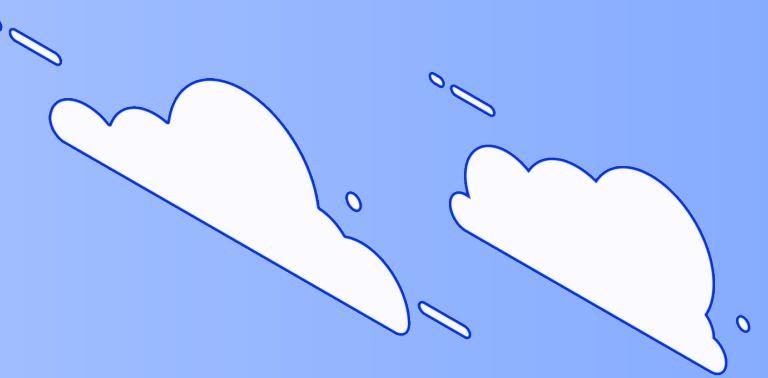
ZAP by [Checkmarx](#)

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	0
Low	1
Informational	5
False Positives	0

Busca vulnerabilidades comunes en aplicaciones web, como inyecciones, configuraciones inseguras o exposición de datos.

CHANGE MANAGEMENT Y RELEASE NOTES



CHANGE MANAGEMENT

- Registro del cambio: Cada modificación se documenta indicando motivo y alcance.
- Revisión obligatoria: Todo cambio requiere aprobación mediante Pull Request.
- Validación automática: El pipeline ejecuta tests, seguridad y performance antes del merge.
- Promoción por ambientes: Los cambios pasan ordenadamente por dev → stage → prod.
- Despliegue controlado: Producción solo se actualiza con aprobación previa.
- Rollback claro: Cada release define cómo revertir el cambio si algo falla.
- Trazabilidad completa: Artifacts y versiones dejan historial automático.
- Monitoreo posterior: Métricas y logs verifican la estabilidad después del despliegue.



RELEASE NOTES

Las Release Notes registran qué cambios, features, bugs corregidos.

The screenshot shows a GitHub release page for a repository. At the top, there are tabs for "Releases" (which is selected) and "Tags". On the right, there are buttons for "Draft a new release" and "Find a tag". The main content area displays a release titled "Production Release 2025.12.01-d5d21b4" (marked as "Latest"). The release was made "2 minutes ago" by the user "github-actions" under the branch "prod-2025.12...." and tag "d5d21b4". A "Compare" button is available. Below the title, the release notes are listed under the heading "Changes".

Production Release 2025.12.01-d5d21b4 Latest

2 minutes ago
github-actions
prod-2025.12....
d5d21b4
Compare ▾

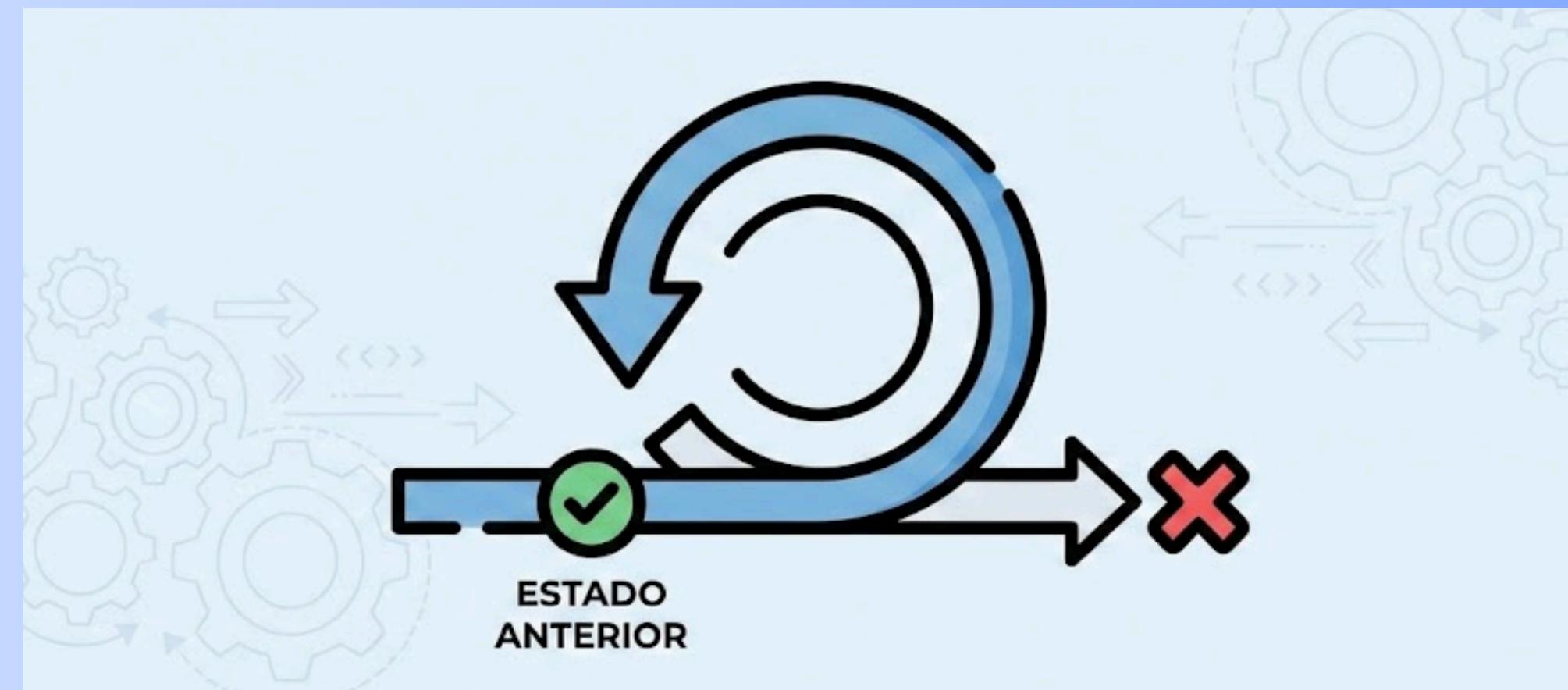
Production Release 2025.12.01-d5d21b4

Changes

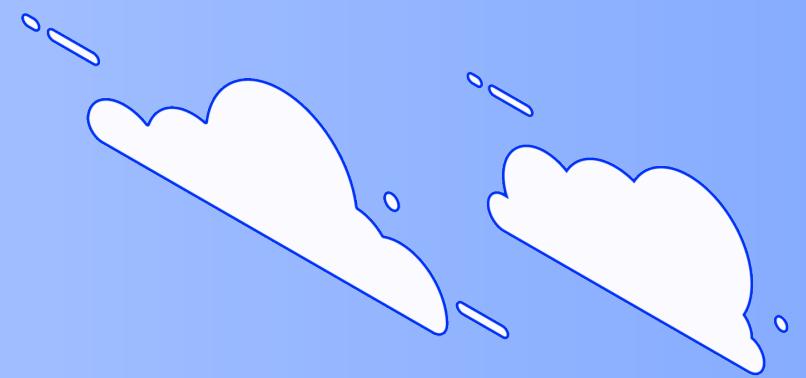
- feat: Add security scan workflow with Trivy, SonarQube, and OWASP ZAP ([d5d21b4](#))
- fix: Simplify image tags extraction in CI/CD workflow ([b9a32fd](#))
- fix: Update image tags output delimiter for CI/CD workflow ([33b518f](#))
- refactor: code structure for improved readability and maintainability ([01acdcf](#))
- fix: Update Zipkin image tag and change imagePullPolicy to IfNotPresent ([8855b3d](#))
- fix: Remove manual approval step and adjust deployment dependencies ([00b77e4](#))
- fix: Update manual approval step to use GITHUB_TOKEN for authentication ([bb6a80a](#))
- fix: Update checkout action to use actions/checkout@v4 in CI/CD workflow ([7e83bff](#))

PLAN DE ROLLBACK

- Detectar fallo: alertas o métricas anómalas tras el release.
- Revertir versión: desplegar la imagen anterior o aplicar terraform rollback.
- Restaurar datos: usar último backup si aplica (DB/Storage).
- Validar: revisar logs, health-checks y monitoreo.
- Documentar: registrar causa y acciones tomadas.



OBSERVABILIDAD Y MONITOREO



MONITOREO

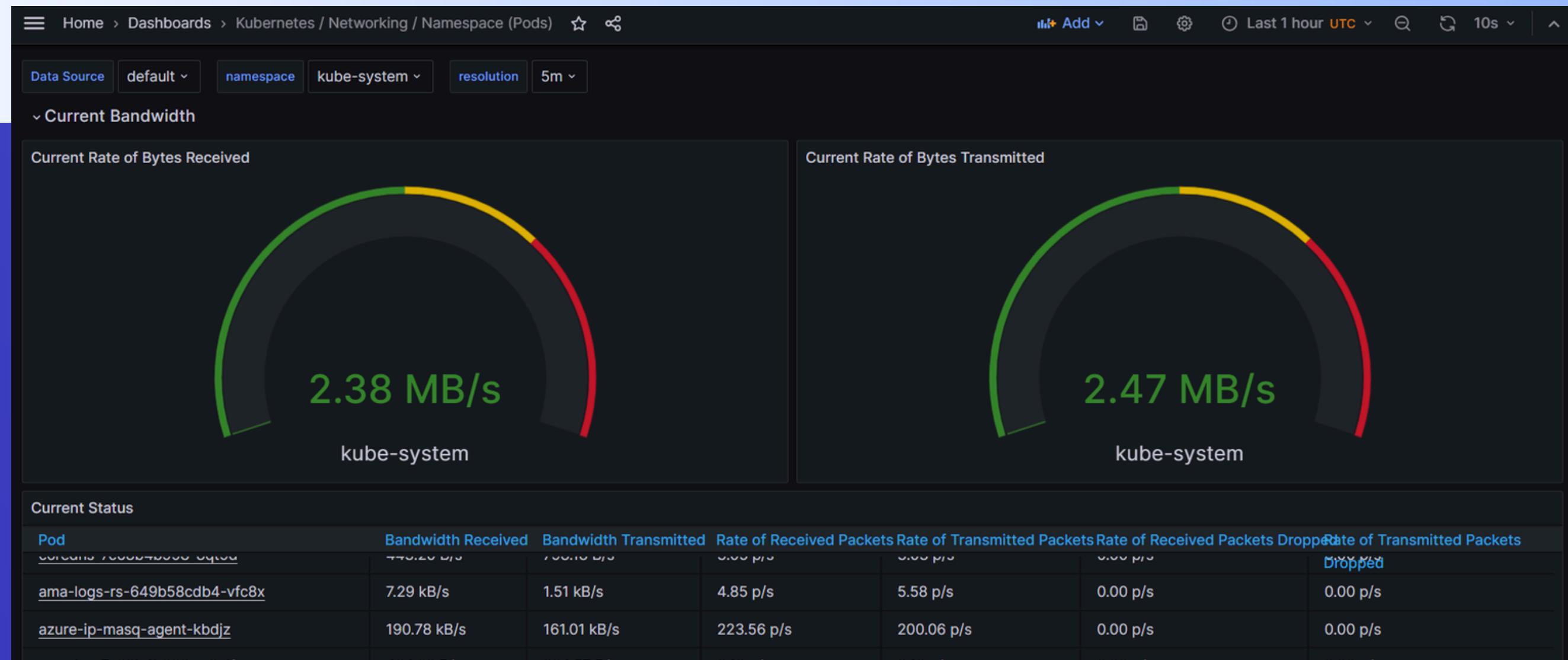
Prometheus

The screenshot shows the Prometheus Targets page. At the top, there are navigation links: Prometheus, Alerts, Graph, Status ▾, and Help. Below this is a search bar with filters: All scrape pools ▾, All (selected), Unhealthy, Collapse All, and a search input for Filter by endpoint or labels. A blue header bar displays the target details: serviceMonitor/monitoring/ingress-nginx-controller/0 (1/1 up). The main table has columns: Endpoint, State, Labels, and Last Scrape. One row is shown for the endpoint http://10.30.2.61:10254/metrics, which is UP. The Labels column lists container=controller, endpoint=metrics, instance=10.30.2.61:10254, job=ingress-nginx-controller-metrics, namespace=ingress-nginx, pod=ingress-nginx-controller-6f66cf4579-98z7s, and service=ingress-nginx-controller-metrics. The Last Scrape column shows 20.21s ago.

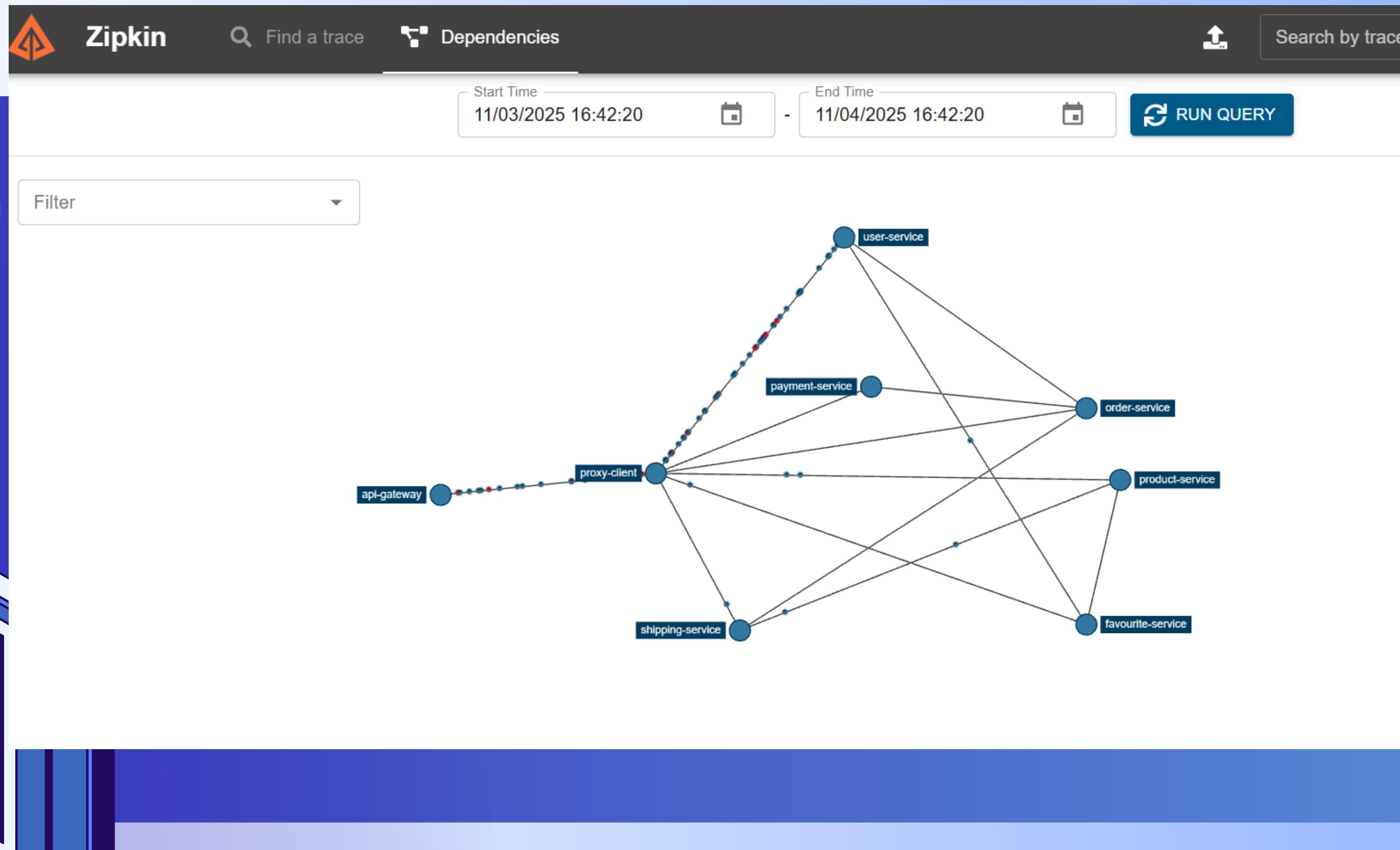
Grafana

The screenshot shows the Grafana Dashboards page. At the top, there is a search bar with the placeholder Q Search or jump to... and a keyboard shortcut ctrl+k. Below the search bar is a sidebar with Home > Dashboards. The main area is titled Dashboards and contains a list of available dashboards. Each dashboard entry includes a checkbox for Name, a preview icon, the dashboard name, and its associated tags. The dashboards listed are: Alertmanager / Overview (alertmanager-mixin), CoreDNS (coredns dns), etcd (etcd-mixin), Grafana Overview (kubernetes-mixin), Kubernetes / API server (kubernetes-mixin), Kubernetes / Compute Resources / Multi-Cluster (kubernetes-mixin), Kubernetes / Compute Resources / Cluster (kubernetes-mixin), Kubernetes / Compute Resources / Namespace (Pods) (kubernetes-mixin), Kubernetes / Compute Resources / Namespace (Workloads) (kubernetes-mixin), and Kubernetes / Compute Resources / Node (Pods) (kubernetes-mixin).

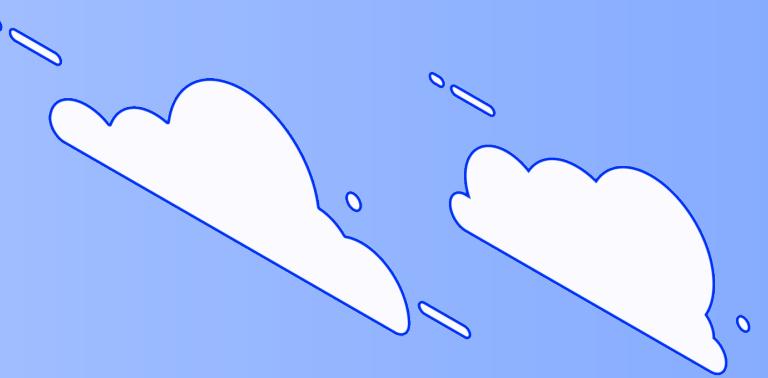
GRAFANA DASHBOARD



MONITOREO ZIPKIN



SEGURIDAD



ESCANEO CONTINUO

✓ feat: Enhance OWASP Dependency-Check step with optional NVD API Key s... #4

Summary

Triggered via push 11 minutes ago

Nicolas-CM pushed → ad13c82 master

Status: Success | Total duration: 8m 16s | Artifacts: 6

Jobs:

- ✓ Build & Unit Tests SONARQ...
- ✓ Trivy Filesystem Scan
- ✓ Trivy Config Scan
- ✓ Trivy Image Scan (api-gatew...
- ✓ Trivy Image Scan (cloud-con...
- ✓ Trivy Image Scan (favourite-...
- ✓ Trivy Image Scan (order-serv...
- ✓ Trivy Image Scan (payment-...
- ✓ Trivy Image Scan (product-s...
- ✓ Trivy Image Scan (proxy-clie...
- ✓ Trivy Image Scan (service-di...
- ✓ Trivy Image Scan (shipping-...
- ✓ Trivy Image Scan (user-servi...
- ✗ OWASP ZAP Security Scan
- ✓ OWASP Dependency Check
- ✓ Security Scan Summary

security-scan.yml
on: push

Build & Unit Tests SONARQ... 2m 53s → **OWASP Dependency...** 5m 10s → **Security Scan Summary** 2s

Matrix: Trivy Image Scan

10 jobs completed

Show all jobs

Trivy Filesystem Scan 4m 30s → **Trivy Config Scan** 23s → **OWASP ZAP Security Scan** 0s

```
graph LR; A[Build & Unit Tests SONARQ...) --> B[OWASP Dependency...]; B --> C[Security Scan Summary]; D[Matrix: Trivy Image Scan] --- E[10 jobs completed]; F[Trivy Filesystem Scan] --- G[Trivy Config Scan]; G --- H[OWASP ZAP Security Scan]
```

ESCANEO CONTINUO

Trivy

 trivy-config-scan	20.3 KB
 trivy-image-scan-api-gateway	26.5 KB
 trivy-image-scan-proxy-client	30.9 KB

ZAP

 **ZAP Scanning Report**

Site: <http://68.211.65.139:8080>

Generated on Sat, 29 Nov 2025 17:46:26

ZAP Version: 2.16.1

ZAP by [Checkmarx](#)

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	0
Low	1
Informational	5
False Positives:	0

Sonarqube (SonarCloud)

SonarQube cloud My Projects My Issues Explore [Upgrade](#)

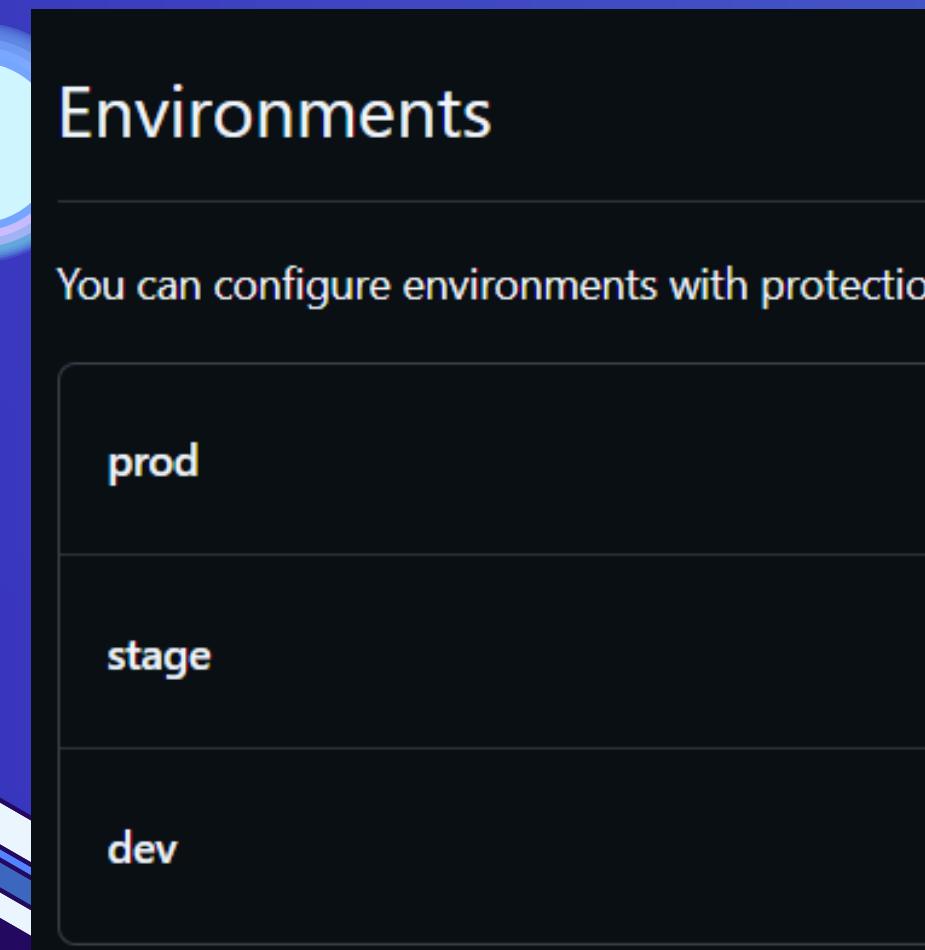
Nicolas Cuellar > ecommerce-microservice-backend-app > stage Last analysis had a warning

Summary Issues Security Hotspots More

New Code Overall Code

Security	3 Open issues	E	Reliability	0 Open issues	A	Maintainability	49 Open issues	C
Accepted Issues	0	G	Coverage	12.5%	O	Duplications	11.3%	H
No conditions set on 1.3k Lines to cover				No conditions set on 13k Lines				
Security Hotspots								I
1								J

GESTIÓN SEGURA DE SECRETOS



Name	Last updated	
AZURE_CLIENT_ID	4 hours ago	
AZURE_CLIENT_SECRET	4 hours ago	
AZURE_SUBSCRIPTION_ID	4 hours ago	
AZURE_TENANT_ID	4 days ago	
GHCR_TOKEN	4 days ago	
GHCR_USERNAME	4 days ago	
MAIL_PASSWORD	3 hours ago	
MAIL_TO	3 hours ago	
MAIL_USERNAME	3 hours ago	

RBAC

```
resource "kubernetes_service_account" "dev_user" {
  metadata {
    name      = "dev-user"
    namespace = data.kubernetes_namespace.ecommerce_dev.metadata[0].name
  }
}

resource "kubernetes_role" "dev_reader" {
  metadata {
    name      = "dev-reader"
    namespace = data.kubernetes_namespace.ecommerce_dev.metadata[0].name
  }

  rule {
    api_groups = ["", "apps", "networking.k8s.io"]
    resources  = ["pods", "services", "configmaps", "secrets",
                 "deployments", "ingresses"]
    verbs      = ["get", "list", "watch"]
  }
}
```

Útil para: desarrolladores, QA o monitoreo con permisos limitados.

El usuario resultante solo puede visualizar recursos del namespace, sin crear, modificar o borrar nada.

```
C:\Users\nicol>kubectl auth can-i list pods --as=system:serviceaccount:ecommerce-dev:dev-user -n ecommerce-dev
yes
```

```
C:\Users\nicol>kubectl auth can-i create pods --as=system:serviceaccount:ecommerce-dev:dev-user -n ecommerce-dev
no
```

PORTAL AZURE

Recursos

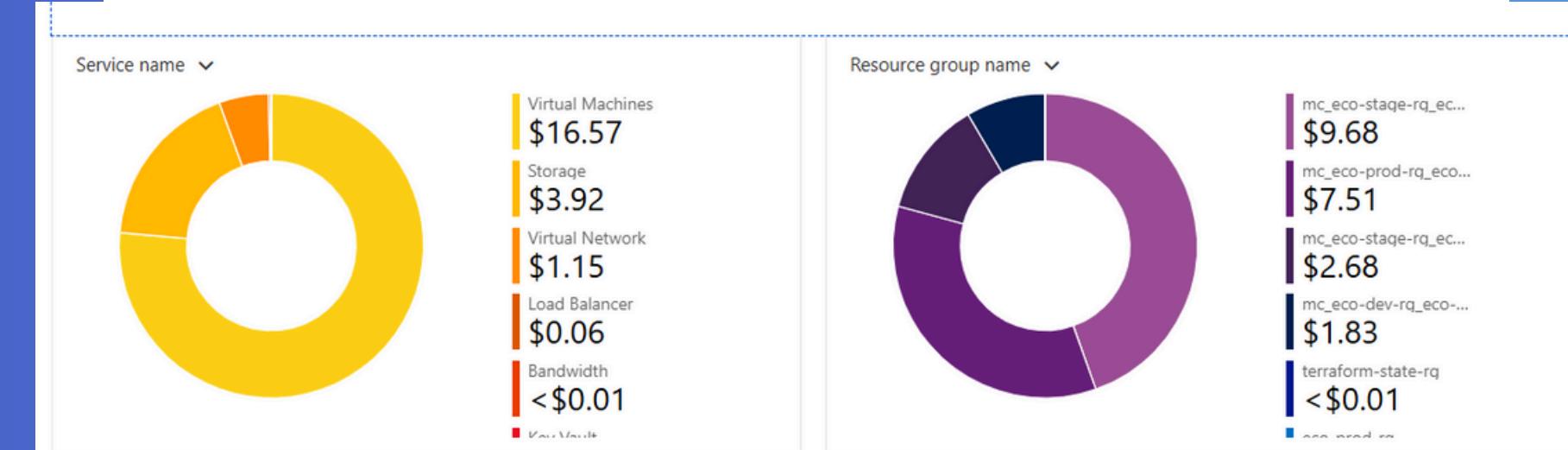
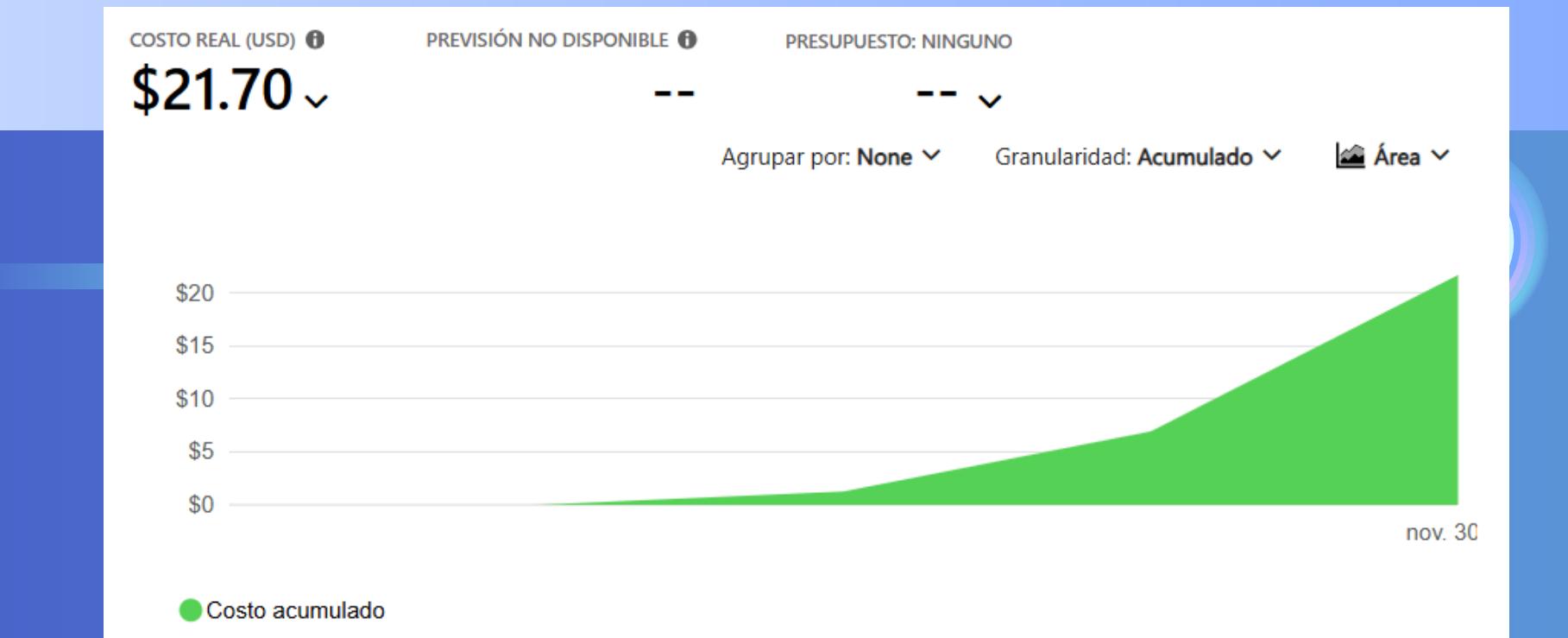
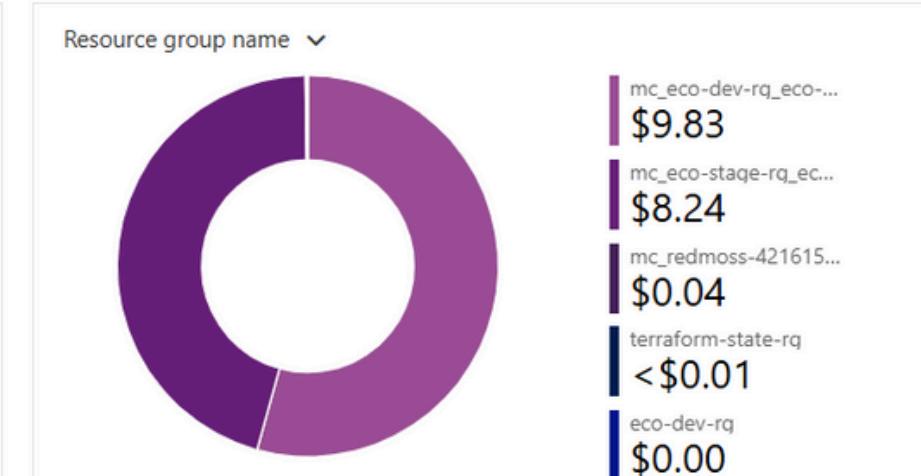
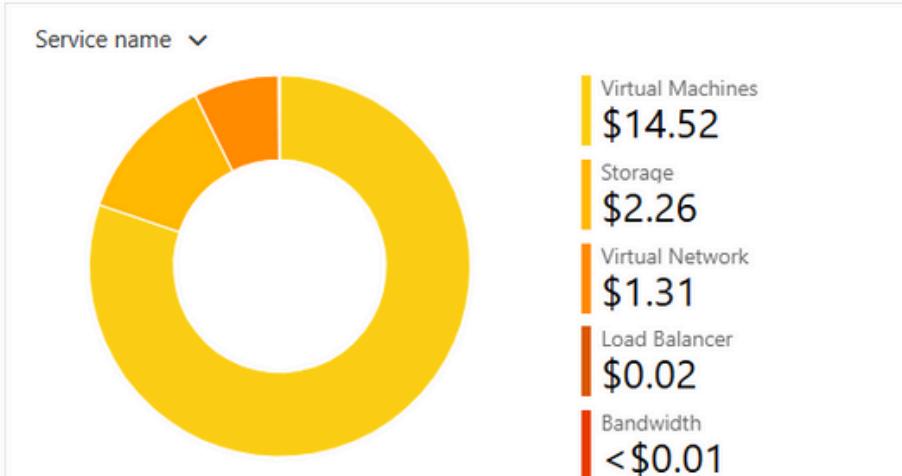
Reciente Favorito

Nombre	Tipo	Última consulta
 terraform-state-rg	Grupo de recursos	hace unos segundos
 eco-prod-aks	Servicio de Kubernetes	hace 14 horas
 eco-prod-rg	Grupo de recursos	hace 1 día
 eco-stage-aks	Servicio de Kubernetes	hace 1 día
 eco-stage-rg	Grupo de recursos	hace 1 día
 eco-dev-aks	Servicio de Kubernetes	hace 2 días
 eco-dev-rg	Grupo de recursos	hace 2 días
 tfstateeco2025dev	Cuenta de almacenamiento	hace 3 días

[Ver todo](#)

Nombre ↑	Suscripción	Ubicación
 eco-dev-rg	... Azure for Students	Mexico Central
 eco-prod-rg	... Azure for Students	Central US
 eco-stage-rg	... Azure for Students	Chile Central

COSTOS



INCONVENIENTES

- Cambio de Infra (gcp a Azure)
- TLS
- ELK Stack (Elasticsearch, Logstash, Kibana)
- Créditos de Azure (Cambio de cuenta)
- Restricción de región
- Restricción de VM
- Problemas de arquitectura en las imágenes (arm64)
- Crear imágenes multi-arch
- SonarCloud (cambiar la main branch)

```
ecture}{ '\n'}{end}"  
aks-default-27198991-vmss000000 arm64  
aks-pool2-11762582-vmss000000 amd64
```

RESULTADOS Y BENEFICIOS

CONCLUSIONES

- 
- Despliegue automatizado
 - Arquitectura en la nube implementada con buenas prácticas
 - Alta disponibilidad
 - Monitoreo centralizado con Log Analytics + Prometheus + Grafana
 - Reducción de errores manuales
 - Control de costos

LECCIONES APRENDIDAS

- Importancia de la infraestructura como código
- Gestión de permisos y roles en Azure
- Despliegue en kubernetes muy similar a minikube
- Todo automatizado (CI/CD)
- Buenas prácticas de seguridad
- Documentación y visualización



GRACIAS

REPOSITORIO