MagicLook

Equipa:
Ivan Horoshko - 120603 (Team Coordinator & DevOps master)
Maria-Aleksandra Korjenevskaya - 118769 (Product owner)
Gonçalo Floro Garcia Ferreira - 120189 (QA Engineer)

Conceito do Produto

O produto é uma aplicação online que serve para alugar roupas de cerimónia, permitindo que utilizadores encontrem, reservem e devolvam roupas para eventos únicos, promovendo a reutilização e reduzindo o desperdício.

Personas

Camila (Staff)
Idade: 30 anos

Alice (Cliente)
Idade: 25 anos

Epics

Epic 1: “Reserva de Roupa”

Epic 2: “Gestão de Itens”

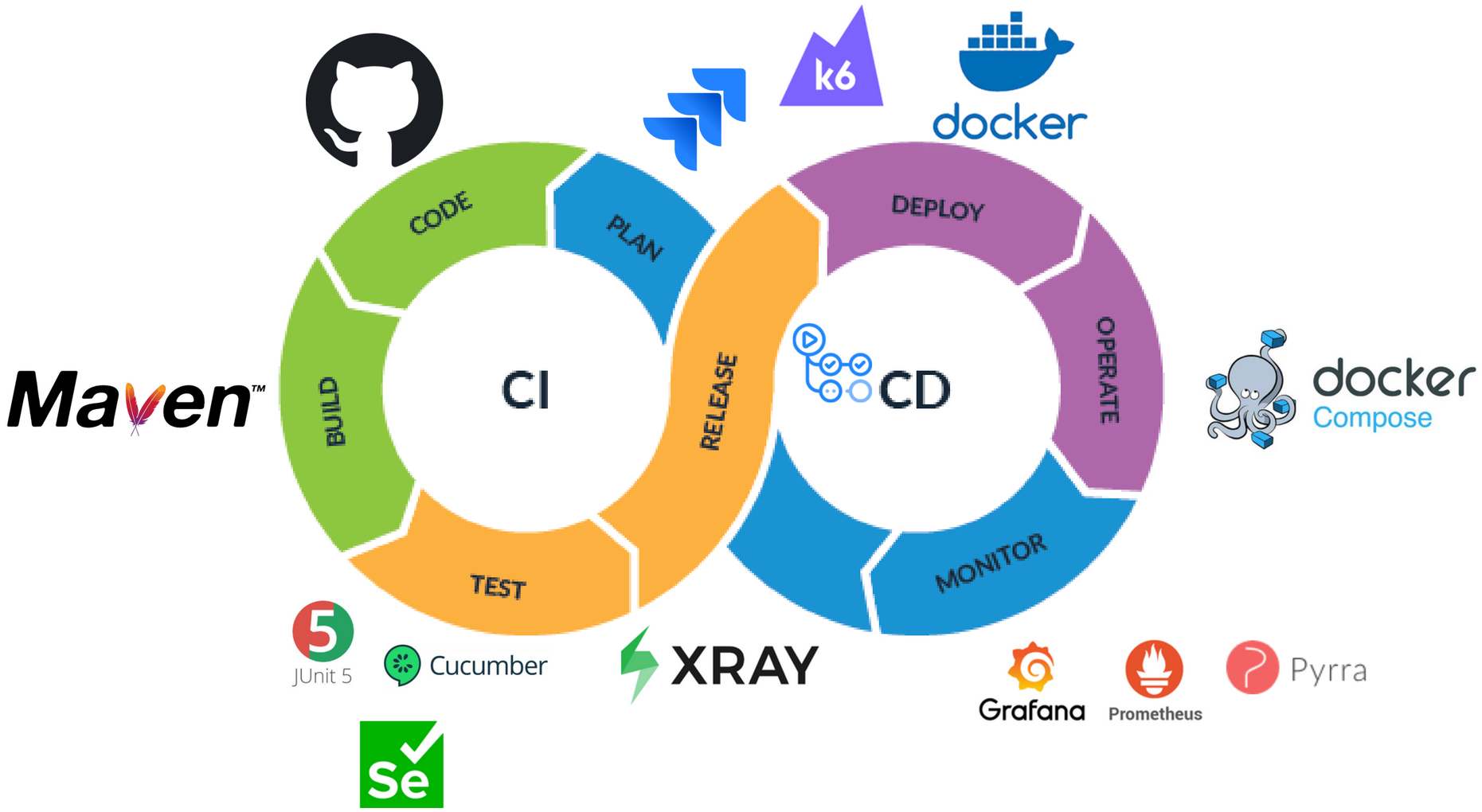
Epic 3: “Reporte de Danos”

Epic 4: “Perfis de Utilizadores”

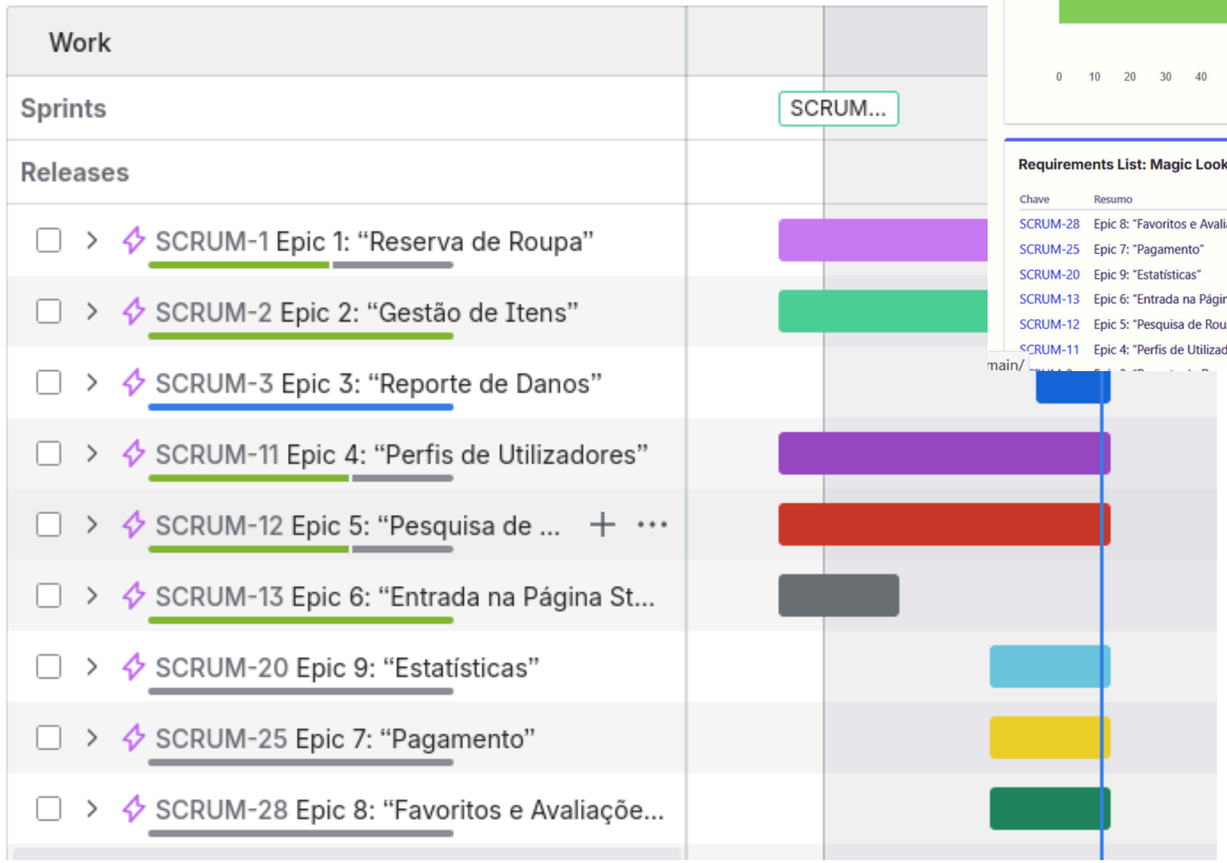
Epic 5: “Pesquisa de Roupas”

Epic 6: “Entrada na Página Staff”

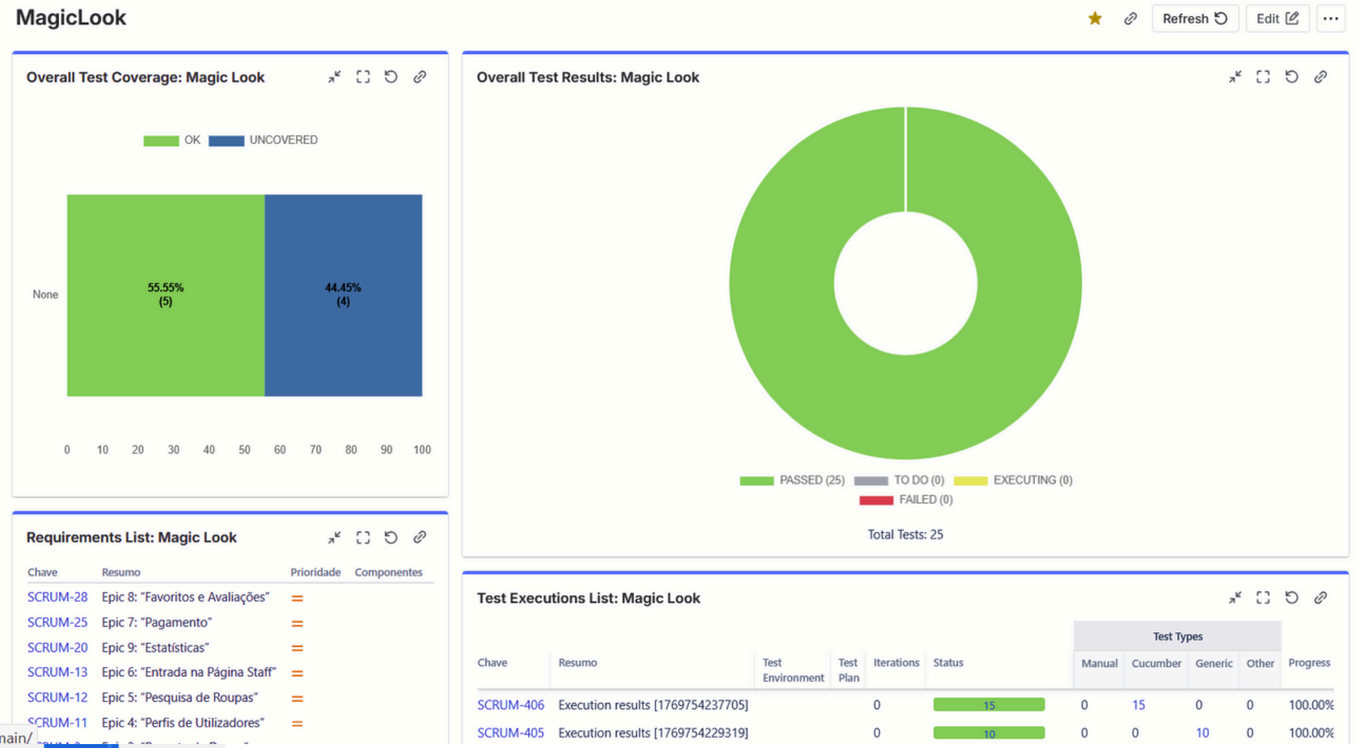
CI/CD Pipeline



Jira


Work
Sprints
SCRUM...
Releases
SCRUM-1 Epic 1: “Reserva de Roupa”
SCRUM-2 Epic 2: “Gestão de Itens”
SCRUM-3 Epic 3: “Reporte de Danos”
SCRUM-11 Epic 4: “Perfis de Utilizadores”
SCRUM-12 Epic 5: “Pesquisa de ...
SCRUM-13 Epic 6: “Entrada na Página St...
SCRUM-20 Epic 9: “Estatísticas”
SCRUM-25 Epic 7: “Pagamento”
SCRUM-28 Epic 8: “Favoritos e Avaliaçõe...

Xray


MagicLook
Refresh
Edit
Overall Test Coverage: Magic Look
OK
UNCOVERED
None
55.55%
(5)
44.45%
(4)
0 10 20 30 40 50 60 70 80 90 100
Overall Test Results: Magic Look
PASSED (25)
TO DO (0)
EXECUTING (0)
FAILED (0)
Total Tests: 25
Requirements List: Magic Look
Chave Resumo Prioridade Componentes
SCRUM-28 Epic 8: “Favoritos e Avaliações”
SCRUM-25 Epic 7: “Pagamento”
SCRUM-20 Epic 9: “Estatísticas”
SCRUM-13 Epic 6: “Entrada na Página Staff”
SCRUM-12 Epic 5: “Pesquisa de Roupas”
SCRUM-11 Epic 4: “Perfis de Utilizadores”
Test Executions List: Magic Look
Test Types
Chave Resumo Test Environment Test Plan Iterations Status Manual Cucumber Generic Other Progress
SCRUM-406 Execution results [1769754237705] 0 15 0 15 0 0 100.00%
SCRUM-405 Execution results [1769754229319] 0 10 0 0 10 0 100.00%

Conjunto QA - backend

Testes unitários:

- Serviços e controladores
- JUnit5, Mockito

```java
@Test
void testCreateBooking_ItemNotFound() {
    when(itemRepository.findById(bookingRequest.getItemId()))
        .thenReturn(Optional.empty());

    RuntimeException exception = assertThrows(RuntimeException.class, () -> {
        bookingService.createBooking(bookingRequest, testUser);
    });

    assertEquals("Item não encontrado", exception.getMessage());
    verify(itemRepository, times(1)).findById(bookingRequest.getItemId());
    verify(bookingRepository, never()).save(any(Booking.class));
}
```

Conjunto QA - backend

Testes de integração:

- H2 Database
- Spring Boot Test

```java
@DisplayName("POST /magiclook/staff/item → success creates item and redirects")
void addItem_success_createsItemAndRedirectsToDashboard() {
        String sessionCookie = loginAsStaff(seededUsername, seededPassword);

        // Check if exists
        Optional<Item> foundItem = itemRepository.findByAllCharacteristics("Vestido Azul", "Seda", "Azul",
                        "Zara", "F", "Vestido", "Curto", seededShopId);

        // Get initial itemSingle instances for that item
        Integer initialCount = 0;
        List<ItemSingle> itemSingles = List.of();
        if (foundItem.isPresent()) {

                Item item = foundItem.get();

                itemSingles = itemSingleRepository.findByItem_ItemId(item.getItemId());
                initialCount = itemSingles.size();

        }

        // Make POST request
        ResponseEntity<String> response = restTemplate.exchange(
                        url("/magiclook/staff/item"),
                        HttpMethod.POST,
                        multipartBody(false, null, sessionCookie),
                        String.class);

        Assertions.assertThat(response.getStatusCode())
                        .isEqualTo(HttpStatus.FOUND);
```

Conjunto QA - backend

```
▶ Run bash run-performance-tests.sh

╔═══════════════════════════════════════════════════╗
║ MAGICLOOK - EXECUÇÃO DE TESTES DE PERFORMANCE     ║
╚═══════════════════════════════════════════════════╝

Verificando disponibilidade da aplicação...
✓ Aplicação acessível
Executando 9 teste(s) de performance (sequencialmente)...
═══════════════════════════════════════════════════

[1/9] ▶ Smoke Test
```

Testes de Performance:

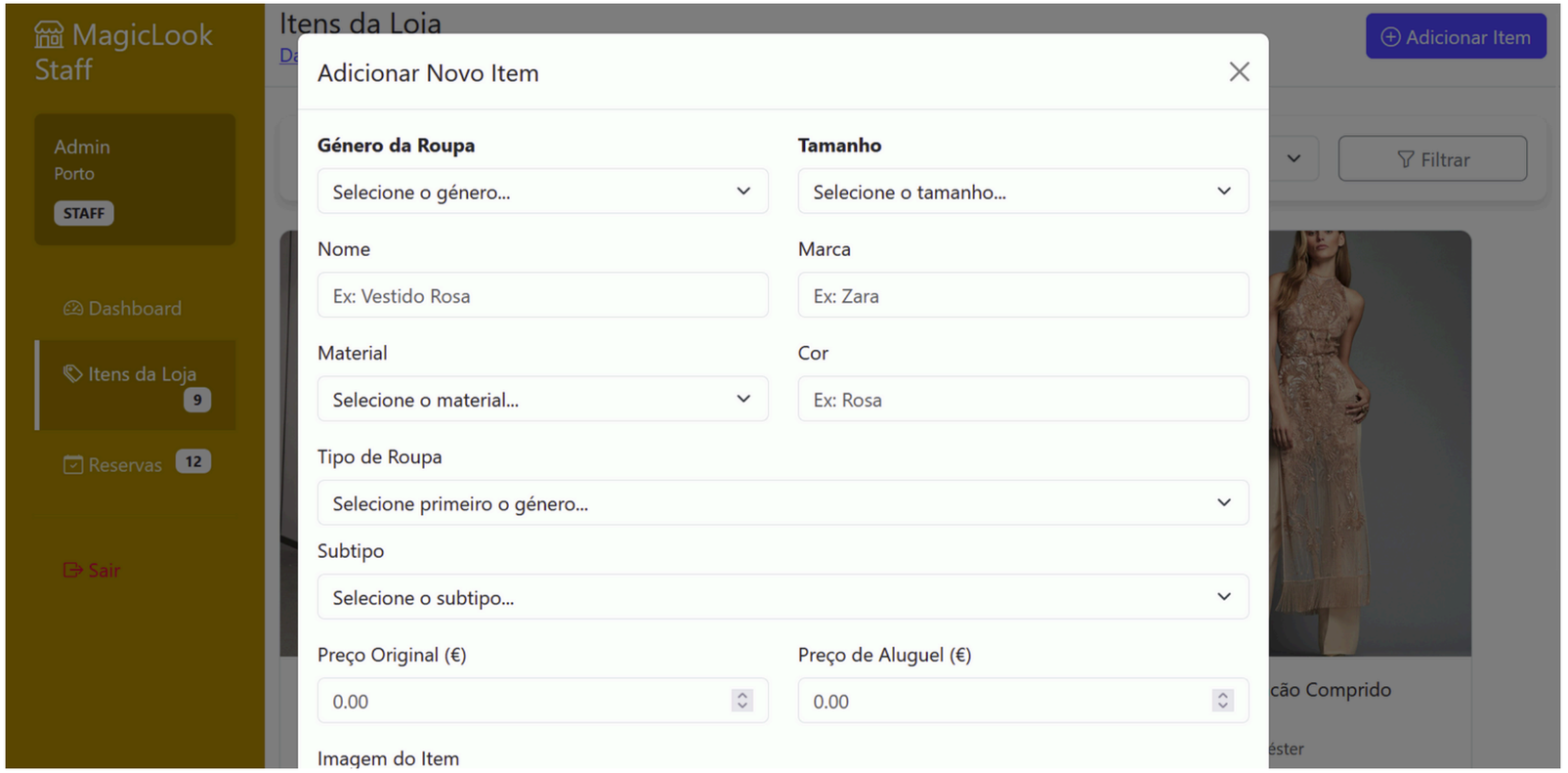
- K6

```
// smoke-test.js
import http from 'k6/http';
import { check, group } from 'k6';
import { Rate, Trend } from 'k6/metrics';

// Métricas customizadas
const errorRate = new Rate('errors');
const responseTimes = {
  home: new Trend('response_time_home'),
  login: new Trend('response_time_login'),
  items: new Trend('response_time_items'),
  availability: new Trend('response_time_availability'),
  booking: new Trend('response_time_booking'),
};

export const options = {
  vus: 3,  // Apenas 3 VUs para smoke test
  duration: '1m',  // 1 minuto é suficiente
  thresholds: {
    errors: ['rate<0.01'],  // Menos de 1% de erros
    http_req_duration: ['p(95)<3000'],  // 95% das requisições < 3s
    'response_time_home': ['p(95)<1000'],
    'response_time_login': ['p(95)<1500'],
  },
  noConnectionReuse: false,  // Reutilizar conexões
  discardResponseBodies: false,  // Manter corpos para verificações
};
```

Conjunto QA - frontend

Testes de interface:

- Cucumber & Selenium



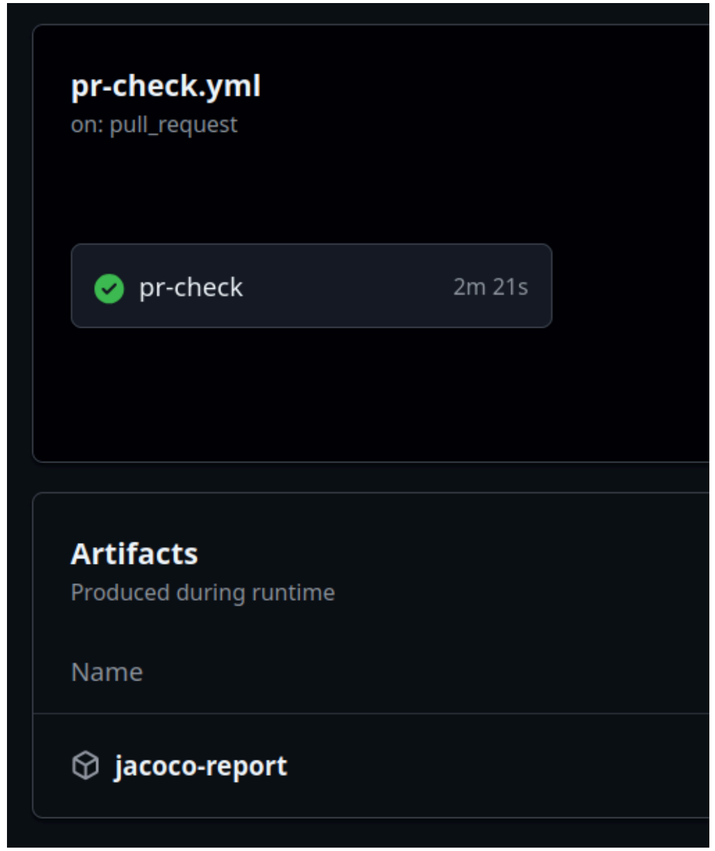
Implementação da QA

Workflow(CI/CD):

- Trigger: Em todos os Pull Request
- Backend: testes unitários e de integração
- Relatório de todos os resultados no Jira(Xray)

Quality gates & Acceptance criteria:

- Testes: todos passam(unitários e de integração)
- Cobertura de SonarQube: 80%



SonarQube

New Issues

0

No conditions set

Accepted Issues

0

Valid issues that were not fixed

Coverage

91.16%

Required: ≥ 80.0%
on **1.1k** New Lines to cover

Duplications

0.0%

Required: ≤ 3.0%
on **14k** New Lines

Security Hotspots

0

No conditions set

Observabilidade

Ferramentas de AI

- Resolução de problemas mais rápido:
redução do tempo de debbugging ao corrigir rapidamente problemas de segurança e qualidade

- Automatização de geração de testes:
ajuda na integração de testes com coverage consistente

Demo

http://deti-tqs-20.ua.pt