Projeto TQS - Apresentação Final

GAMERENT
PLAY • DISCOVER • UNLOCK FUN

Duarte Branco 119253
Filipe Viseu 119192
Samuel Vinhas 119405
André Ferreira 119061

Professores:
Ilídio Oliveira
Sérgio Freire

Contexto

GAMERENT
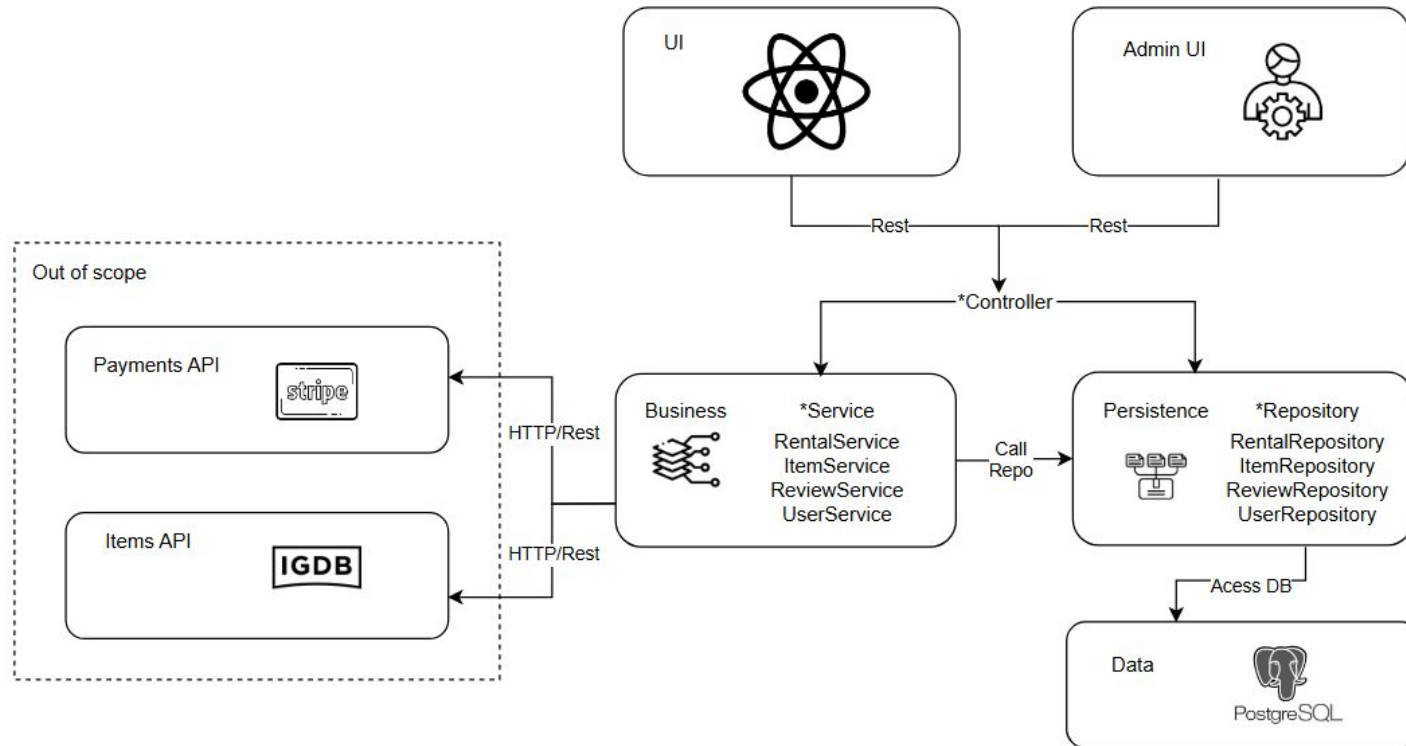PLAY • DISCOVER • UNLOCK FUN

Objetivo:

Desenvolver uma plataforma que permita aos utilizadores listar consolas, jogos e acessórios assim como alugar de outros utilizadores por períodos específicos a preços acessíveis
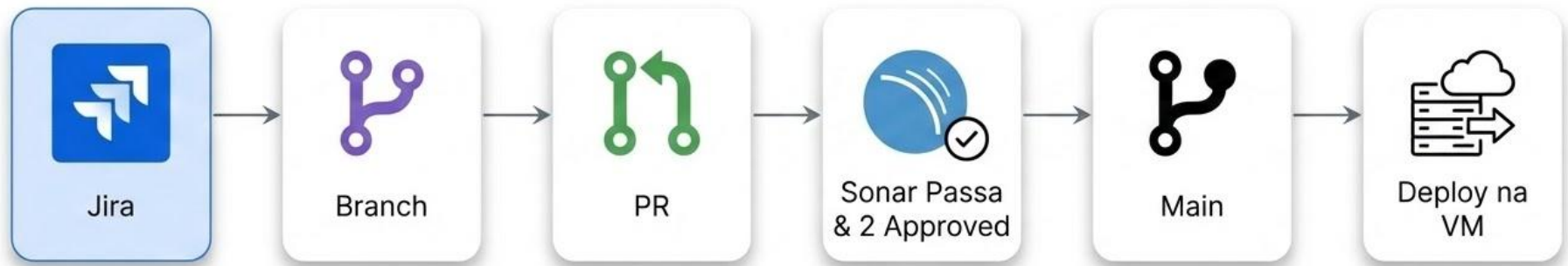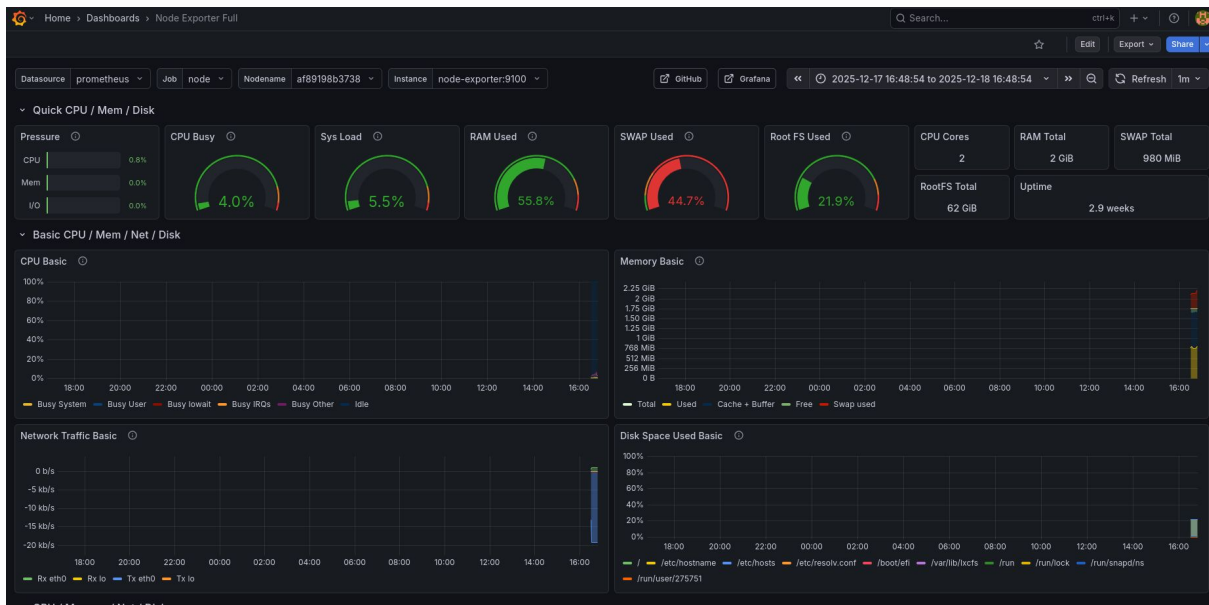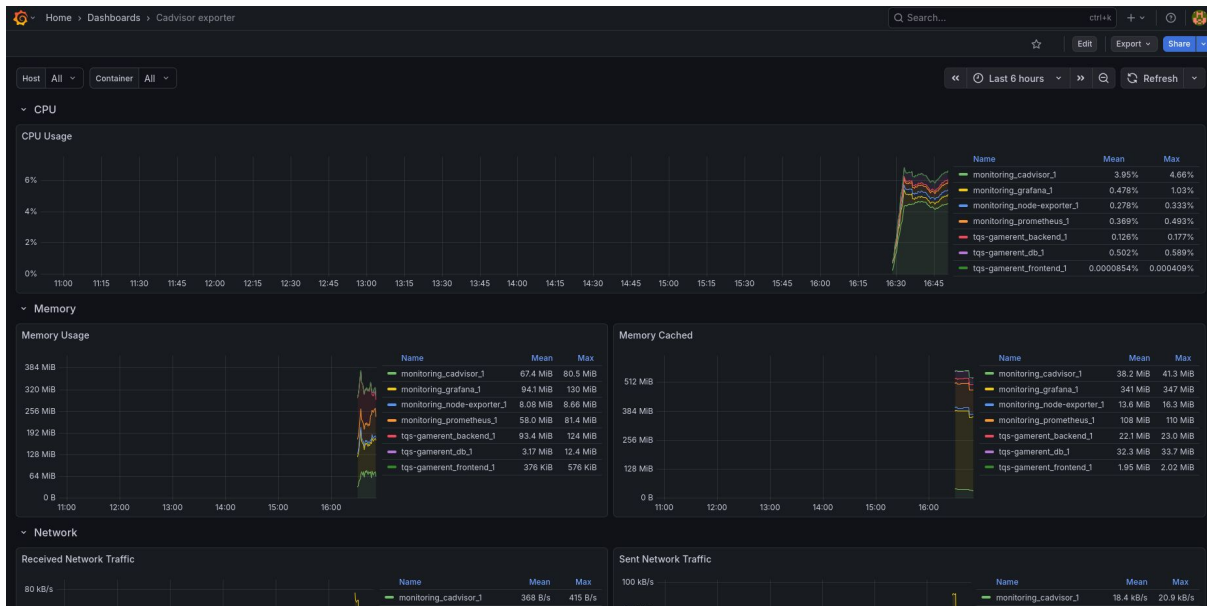
Arquitetura

Demonstração

Workflow

Jira → Branch → PR → Sonar Passa & 2 Approved → Main → Deploy na VM

System Observability (Grafana)

Node Exporter Full

System Observability (Grafana)

Cadvisor exporter

Testes - Service

```
1   @Test
2   void createBooking_ShouldCreateWhenNoConflict() {
3       when(itemRepository.findById(1L)).thenReturn(Optional.of(item));
4       when(bookingRepository.findByItemIdAndStatus(1L, BookingStatus.APPROVED))
5           .thenReturn(List.of());
6       when(bookingRepository.save(any(BookingRequest.class)))
7           .thenAnswer(invocation -> invocation.getArgument(0));
8
9       LocalDate start = LocalDate.now().plusDays(1);
10      LocalDate end = LocalDate.now().plusDays(5);
11
12      BookingRequest result = bookingService.createBooking(1L, 2L, start, end);
13
14
15      assertNotNull(result);
16      assertEquals(BookingStatus.PENDING, result.getStatus());
17      long days = java.time.temporal.ChronoUnit.DAYS.between(
18          LocalDate.of(2025,12,1), LocalDate.of(2025,12,5)) + 1;
19      assertEquals(15.0 * (double) days, result.getTotalPrice());
20      verify(bookingRepository, times(1)).save(any(BookingRequest.class));
21  }
```

Testes - Playwright

```java
@Test
void testBookingsPageShowsLoginPromptWhenNotAuthenticated() {
    String baseUrl = "http://localhost:3000";
    page.navigate(baseUrl + "/bookings");

    page.waitForTimeout(1000);

    // Should show some form of authentication requirement
    boolean needsAuth = page.isVisible("text=/log.*in/i") ||
                        page.url().contains("/auth") ||
                        page.isVisible("text=Please Log In");

    assertTrue(needsAuth, "Should indicate authentication is required");
}
```

Testes - Cucumber

```
1   Feature: Book Item

2     As a renter

3     I want to book an item

4     So that I can use it

5

6     Scenario: Renter books an item

7       Given a user "renter@example.com" exists

8       And an item "Switch" exists owned by "owner@example.com"

9       When the user "renter@example.com" books "Switch" for tomorrow

10      Then the booking status should be "PENDING"
```

Testes - Integration

```java
1   @Test
2   @XrayTest(key = "TGR-27")
3   @Tag("integration")
4   void createBooking_ShouldReturn200AndCreateBooking() throws Exception {
5       String json = """
6           {
7               "itemId": %d,
8               "userId": %d,
9               "startDate": "2035-12-01",
10              "endDate": "2035-12-05"
11          }
12          """.formatted(item.getId(), renter.getId());
13
14      mockMvc.perform(post("/api/bookings")
15              .contentType(MediaType.APPLICATION_JSON)
16              .content(json))
17              .andExpect(status().isOk())
18              .andExpect(jsonPath("$.id").isNumber())
19              .andExpect(jsonPath("$.itemId").value(item.getId()))
20              .andExpect(jsonPath("$.userId").value(renter.getId()))
21              .andExpect(jsonPath("$.status").value("PENDING"));
22  }
```

Testes - K6

```
# Load:
checks
✓ 'rate>0.99' rate=100.00%

http_req_duration
✓ 'p(95)<500' p(95)=8.37ms

http_req_failed
✓ 'rate<0.02' rate=0.00%

# Spike:
✓ status is 200
✓ response time < 1s

HTTP
http_req_duration..............: avg=5.66ms min=199.65µs med=5.5ms max=30.44ms p(90)=7.29ms p(95)=7.86ms
  { expected_response:true }...: avg=5.66ms min=199.65µs med=5.5ms max=30.44ms p(90)=7.29ms p(95)=7.86ms
http_req_failed................: 0.00%  0 out of 1192
http_reqs......................: 1192   56.421154/s

# Stress:
✓ status is 200

HTTP
http_req_duration..............: avg=8.88ms min=104.65µs med=4.66ms max=6.81s p(90)=5.98ms p(95)=6.59ms
  { expected_response:true }...: avg=8.88ms min=104.65µs med=4.66ms max=6.81s p(90)=5.98ms p(95)=6.59ms
http_req_failed................: 0.00%  0 out of 13473
http_reqs......................: 13473  70.454116/s
```

Sonar

Quality Gate: Sonar way ⓘ

✓ **Passed**

New Code   **Overall Code**

Security

0 Open issues   Ⓐ

Reliability

0 Open issues   Ⓐ

Maintainability

6 Open issues   Ⓐ

Accepted Issues

0

Coverage

86.5%

No conditions set
on **1.6k** Lines to cover

Duplications

1.0%

No conditions set
on **4.2k** Lines

Security Hotspots

0

Xray

Test Repository for project TQS Game Rent

Test Cases BETA   Create ▾

Test Repository

FOLDERS VIEW

Test Repository   20 (20)

Search 🔍   Only My Issues   Tests and Preconditions ▾   Filters ▾   •••

Showing 20 of 20 entries

TGR-41 USER-3
GENERIC DONE

TGR-40 USER-2
GENERIC DONE

TGR-39 USER-1
GENERIC DONE

TGR-38 REVIEW-2
GENERIC DONE

TGR-37 REVIEW-1
GENERIC DONE

TGR-36 PAY-1
GENERIC DONE

TGR-35 ITEM-3
GENERIC DONE

TGR-34 ITEM-2
GENERIC DONE

TGR-33 ITEM-1
GENERIC DONE

TGR-32 IGDB-1
GENERIC DONE

TGR-31 BOOK-5
GENERIC DONE

TGR-30 BOOK-4

Reflexões

Surpresas Positivas	Desafios
● Workflow & QA Gates	● Atrito Inicial
● Testes E2E	● Rigidez Sonarqube

Links

Deploy

Jira

http://deti-tqs-13.ua.pt:3000

http://ua-tqs-gamerent.atlassian.net