deti universidade de aveiro departamento de eletrónica, telecomunicações e informática

HW1: Mid-term assignment report

Pedro Lima Baía Coelho [104247], 2023-04-10

1 Introduçao

1.1 Visão geral do trabalho

Este relatório apresenta o projeto individual a médio prazo necessário para a TQS, abrangendo tanto as características do produto de software como a estratégia de garantia de qualidade adotada.
O objetivo deste projeto individual era o desenvolvimento de uma aplicação web que fornecesse informações acerca da qualidade do ar numa determinada região/cidade. Esta aplicação foi desenvolvida segundo uma aplicação web multi-camadas, em SpringBoot, com o apoio de uma API externa e ainda testes desenvolvidos para garantir a qualidade da aplicação.

1.2 Limitações

Neste momento, ainda não foram desenvolvidos os testes funcionais para a interface web (usando o Selenium WebDriver, por exemplo). Além disso, a aplicação tem só o apoio de uma API externa, podendo levar a problemas futuros caso esta API deixe de estar disponível.

2 Especificação do produto

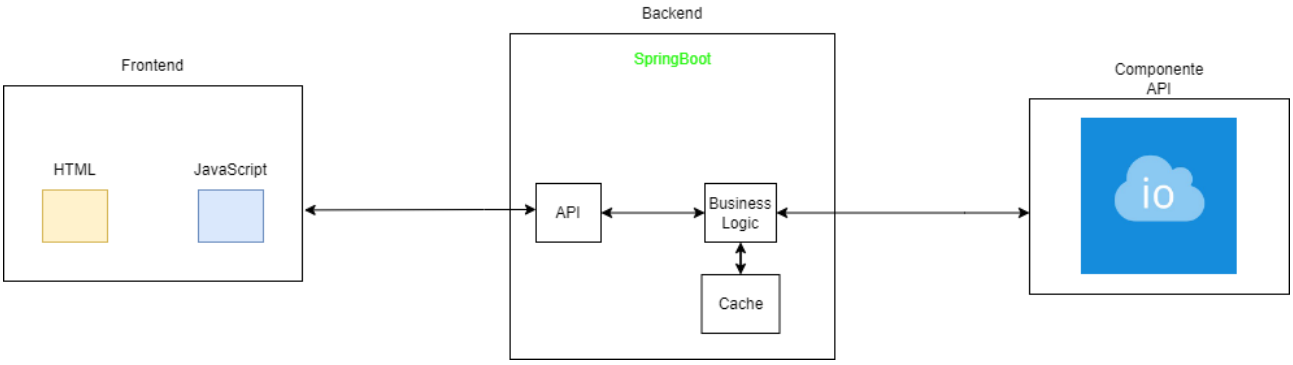
2.1 Alcance funcional e interações suportadas

A aplicação web desenvolvida tem como foco mostrar ao utilizador, de forma simples e minimalista, informações sobre a qualidade do ar numa determinada região/cidade. Quando o utilizador abre a página web, depara-se com uma textbox onde pode inserir o nome de uma cidade que deseja saber a informação da sua qualidade do ar.

2.2 Arquitetura do sistema

A arquitetura do sistema está dividida em três secções:

- Frontend usando **HTML** e **JavaScript**;
- Backend usando **SpringBoot**;
- Componente API usando a **Weatherbit API**.



2.3 API para desenvolvedores

A aplicação web tem apenas dois endpoints:

- /city/{city} – acesso a uma cidade específica que será indicada na parte {city} do url
- /cache – acesso às informações acerca do cache implementado (hits, misses e requests)

3 Garantia de Qualidade

3.1 Estratégia global para os testes

No geral, foi testado tudo um pouco, ficando a faltar como referi anteriormente os testes funcionais para a interface web. Foram desenvolvidos testes de integração utilizando ferramentas como o Mockito e o MockMvc.

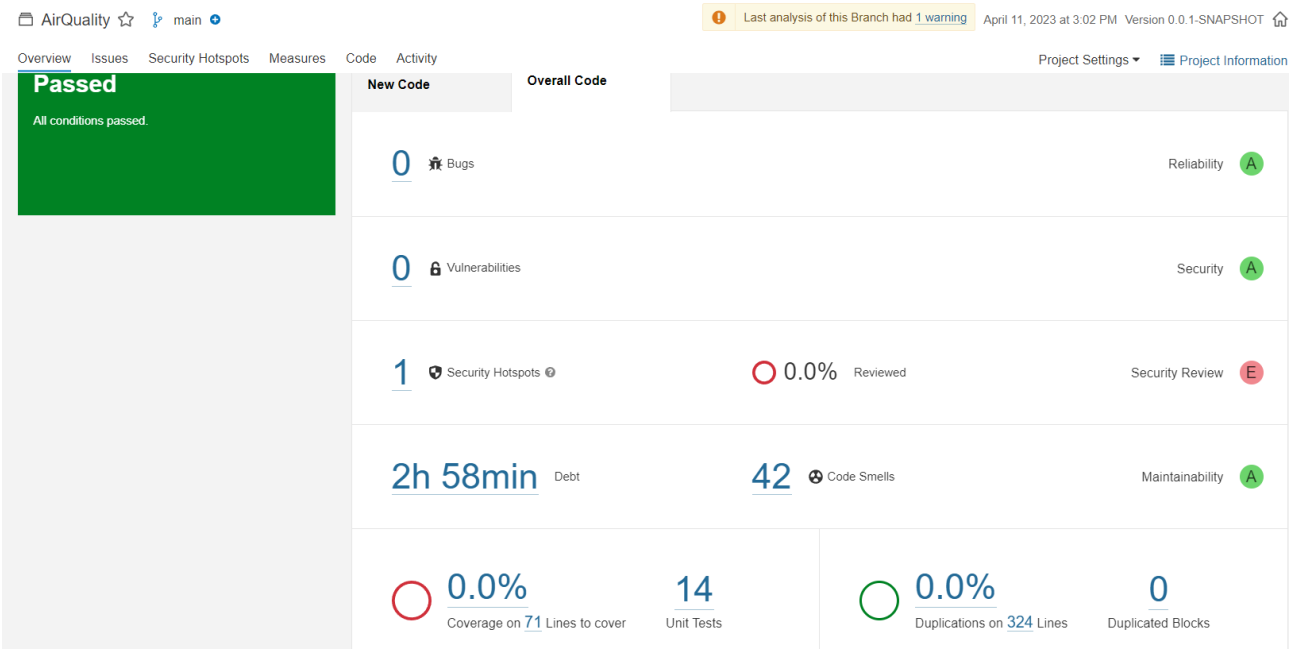
3.2 Testes de unidade e integração

Para os **testes de integração** foi testado a componente do controller usando o MockMvc. Ao utilizar os endpoints de teste (@Test), podemos ter acesso aos comportamento dos testes de forma a termos a certeza que está tudo a trabalhar corretamente. Para isso acontecer, são feitas chamadas a API e a resposta fornecida por esta tem de ser semelhante à que a classe de testes do controller espera.

```
@Autowired
private MockMvc servlet;

@Test
@Order(1)
public void whenGetCacheBeforeRequests_thenReturnValidCache() throws Exception {
    servlet.perform( MockMvcRequestBuilders.get("/api/cache")
            .contentType(MediaType.APPLICATION_JSON) )
            .andDo(print())
            .andExpect(status().isOk())
            .andExpect(jsonPath("numRequests").value(0))
            .andExpect(jsonPath("numMisses").value(0))
            .andExpect(jsonPath("numHits").value(0));
}

@Test
@Order(2)
public void whenGetCacheAfterSomeRequests_thenReturnValidCache() throws Exception {
    servlet.perform( MockMvcRequestBuilders.get("/api/city/Aveiro") );
    servlet.perform( MockMvcRequestBuilders.get("/api/city/Aveiro") );
    servlet.perform( MockMvcRequestBuilders.get("/api/city/Lisboa") );


    servlet.perform( MockMvcRequestBuilders.get("/api/v1/cache")
            .contentType(MediaType.APPLICATION_JSON) )
            .andDo(print())
            .andExpect(status().isOk())
            .andExpect(jsonPath("numRequests").value(3))
            .andExpect(jsonPath("numMisses").value(2))
            .andExpect(jsonPath("numHits").value(2));
}
```

3.3 Análise da qualidade do código

A análise da qualidade de código foi feita através do SonarQube.



4 Referências e recursos

Project resources

Recurso:	URL/localização:
Git repository	https://github.com/PedroC55/tqs_104247
Video demo	Incluído no repositório.

Materiais de referência

API externa - https://www.weatherbit.io/api/airquality-current