Reunião 5

01 de novembro de 2024, 22:30

Presentes : Diogo Domingues, Gabriel Silva, Sebastião Teixeira, Martim Santos
Não Presentes : NA

Ordem de trabalhos:

1) Estabelecimento de uma estratégia para desacoplar os micro serviços uns dos outros;
2) Discussão acerca de políticas para preservar a segurança de chaves, senhas e outros dados confidenciais;
3) Levantamento do trabalho a realizar até ao final da iteração atual;
4) Atribuição dos trabalhos planeados para próxima iteração aos respectivos membros e verificação de tarefas em falta para a mesma.

Apontamentos:

1) Após conversarmos com os professores das aulas práticas, percebemos que tínhamos muitas dependências entre serviços que nos iria obrigar a adotar estratégias para que estes se pudessem comunicar.
2) Um dos pontos de maior acoplamento está na validação da autenticação do utilizador ao aceder à *API* dos diversos serviços. Para resolver este problema, propôs-se a transferência da responsabilidade da validação dos tokens de autenticação (*JWT*) para o *API Gateway*. Para tal, deverá ser gerado um par de chaves público-privada, onde a chave privada ficará sob a custódia do *Users Service* e a chave pública no *API Gateway*. O token deve conter o *id* do utilizador.
3) *AuthService* muda de nome para *Users Service* na arquitetura.
4) Após todos os ajustes à arquitetura, conseguimos desacoplar quase todos os serviços. No entanto, o *Animals Service* e o *Reports Service* continuam a ler dados comuns. Na realidade, ambos os serviços fazem operações idênticas (leem os dados dos animais), no entanto, diferem na forma como os disponibilizam para o utilizador. O primeiro expõe o acesso aos dados através da *API*, enquanto que os segundo utiliza-os para gerar um relatório, que é indexado na base de dados relacional, permitindo ao utilizador partilhá-lo com terceiros (veterinário por exemplo). Desta forma, para evitar duplicação de estruturas no código, o que dificultaria consideravelmente a manutenção do projeto, consideramos que a criação de uma biblioteca partilhada entre os 2 serviços seria a melhor opção. Esta biblioteca é responsável apenas por interagir com a entidade *Animal* na base de dados relacional (classes *entity* e *repository*) e ler dados (brutos ou agregados) armazenados na base de dados temporal. Os módulos que importem esta devem implementar os *services* e *controllers* que forem necessários.
5) É feita a junção de *Animals Service* com *Animals Data Service* no repositório.
6) Define-se que as credenciais das bases de dados deverão estar localizadas num ficheiro .env que deve ser importado no docker compose e nos módulos que de tal necessitarem.

7) No caminho da *API* exposta para o utilizador deve estar contida a versão da *API* (ex.: */api/v1/user*).
8) Fez-se uma análise das tarefas levantadas para a iteração seguinte. A acrescentado *Implement Reports Generation*. Foram abertos *issues* para as tarefas que estavam como rascunho.
9) Estabeleceu-se como tarefa prioritária a elaboração de descrições de *issues* que se encontrem vazias ou consideravelmente incompletas.
10) Realizou-se um brainstorm sobre ferramentas e bibliotecas para delimitar a área em que o animal deve estar, a partir da qual deve ser enviado um alerta caso o animal se afaste desta.
11) Foram atribuídas de as tarefas da próxima iteração aos respectivos membros tendo em conta os pontos de esforço estimados para as mesmas.
12) A reunião terminou às 23:40.