

Minuta 10



Participantes:

- Afonso Ferreira
- João Neto
- Ricardo Antunes
- Pedro Ponte

Temas a discutir:

- Reformulação da arquitetura
- Reformulação dos models baseado nas mudanças na arquitetura
- Construção de endpoints e serviços
- Discussão sobre o serviço de envio de SMS para autenticação

Memorando:

Depois da apresentação de quinta-feira, ficou claro que a arquitetura tinha que ser reformulada, antes até que qualquer passo sobre os endpoints pudesse ser feito. Após explorarmos as timeseries DB's, chegámos à conclusão de que este modelo seria o mais adequado para guardar dados vitais dos nossos pacientes, especialmente após comparação direta com os Document-based DB's. Esta mudança é fundamentada pelo facto de que as timseries DB's serem adequadas para pesquisas sobre timestamp, e também bastante adequadas para análise analítica e construção de gráficos com os dados colecionados. Após uma pesquisa sobre as diferentes ferramentas que poderíamos utilizar, optámos por escolher o InfluxDB, tanto pelo facto de poder ser integrado no nosso projeto SpringBoot sem grandes problemas, mas também pelo facto de ter um atributo tag que se comporta como uma primary key (atributo indexado o que acelera pesquisas), o que nos permitiria inserir dados para cada cama separadamente. Para além desta mudança, introduzir um micro-serviço de autenticação, que faz uso da ferramenta twillio. Escolhemos esta ferramenta primariamente pelos recursos disponíveis na versão gratuita da ferramenta. Também criámos um novo serviço de notificações dado o número diferente de tipos de notificações que poderíamos ter, requerendo assim um diferente tratamento de dados no que diz respeito a estas notificações. Fizemos uma divisão dos micro-serviços Doctor / Nurse em 3 microserviços: Doctor, Nurse e Hospital Manager. Também passámos (quase) todos os dados anteriormente presentes na entidade Patient de um documento para uma entidade relacional, estabelecendo relações com outros aspetos do sistema. Também chegámos a acordo na utilização de uma API externa para gerar gráficos utilizando os dados vindos da timseries DB, nomeadamente a ferramenta Vital Graphics.

Como referido em cima, a maior reformulação em relação a models foi o ajuste da entidade Patient para um panorama relacional. Inserimos também mais relações entre entidades que achámos adequadas e que não estavam lá anteriormente, para além de também adicionarmos novas entidades, nomeadamente a entidade Bed (necessária para o acesso aos dados da timeseries DB, e a entidade Notifications, que irá ser utilizada pelo Notifications service.

Feito por: **Ricardo Antunes**

Revisto por: **Pedro Ponte; Afonso Ferreira; João Neto;**

Conseguimos adiantar bastantes endpoints e microserviços, nomeadamente o Nurse service, o Patient service e os endpoints relacionados com o paciente, adiantando assim algum trabalho que estava atrasado. Os endpoints e micro serviços relacionados com o Doctor não tiveram muito progresso já que estava *assigned* ao arquiteto que esteve focado em tomar as melhores decisões para a arquitetura do projeto (depois de ter tomado decisões não tão boas), prevenindo assim mais alterações no futuro.

Apesar de já não ser obrigatório para esta iteração, o DevOps está a tratar da Dockerização da aplicação. Este passo vai permitir que toda a gente comece a trabalhar num ambiente semelhante ao de produção e para além de promover Continuous Integration, vai também evitar muitos problemas futuros de sincronização.

Para terminar a reunião, aproveitámos para deixar em panos limpos o serviço que iremos utilizar para enviar SMS's aos visitantes. Acabámos por optar pelo twilio devido ao plano gratuito que possuía e as ferramentas disponíveis.

Próximo passo:

- Continuar o progresso nos microserviços e endpoints
- Revisão finalíssima da arquitetura
- Acabar docker-compose e Dockerfile
- Planeamento da próxima iteração

Feito por: **Ricardo Antunes**

Revisto por: **Pedro Ponte; Afonso Ferreira; João Neto;**