GymBit

Disciplina:	IES - Introdução à Engenharia de Software
Data:	Aveiro, 17 de dezembro de 2019
Alunos:	88864: Gonçalo Passos 89236: Diogo Andrade 89251: Gonçalo Freixinho 89265: João Dias
Tema:	GymBit consiste numa aplicação web que permite monitorizar os treinos realizados no ginásio e as suas componentes estatísticas de saúde.

1 Introdução

2 Conceito do produto

- Visão
- Personas
- Cenários Principais

3 Arquitetura

- Requerimentos chave
- Vista arquitetural
- Interação entre módulos

4 Perspetiva da Informação

5 Bibliografia e referências

1 Introdução

No âmbito da cadeira de Introdução à Engenharia de Software foi-nos proposto o desenvolvimento de um projeto no qual se criasse um produto, usando conteúdos lecionados nas aulas tais como as ferramentas, o software e técnicas de trabalho colaborativo.

O objetivo do projeto em causa é implementar um fluxo de dados contínuo através do uso de sensores e/ou simuladores de dados, uma REST api para a gestão dos dados entre a interface e base de dados, o uso de Docker containers e também VM (Virtual Machine) para fazer o depoimento contínuo ao longo do desenvolvimento do projeto.

Foi com base nestes pré-requisitos que nos foram apresentados que surgiu o *GymBit*, uma aplicação cujo intuito é realizar a monotorização de pessoas a fazer exercício físico.

2 Conceito do produto

Visão

O objetivo principal da nossa aplicação web é facilitar a monotorização das pessoas enquanto realizam exercício físico, dando acesso, à pessoa em questão e ao personal trainer que a acompanha, aos dados vitais e informação dos exercícios.

Esta aplicação é uma vantagem da qual as pessoas podem tirar grande partido, visto que lhes permite monitorizar o exercício físico e os dados vitais a qualquer distância, coisa que nos dias de hoje não é comum e, ainda que já existam soluções, não são as mais eficazes.Inicialmente o plano envolvia muito trabalho estatístico com base nos dados recebidos.

No entanto, isto acabou por se tornar num problema devido à relação entre o tempo que ocupava e o prazo final, pois exigia demasiado do grupo e dos recursos disponibilizados. O personal trainer teria muito mais poder na aplicação, porque esta lhe permitiria, por exemplo, enviar notificações quando algo estivesse errado e alterar o plano de treino de um cliente. Assim, era possível poupar muito tempo e, em consequência, aumentar a produtividade e eficácia do trabalho.

Personas

Alberto é um ex-banqueiro, já reformado, com 70 anos e viúvo. Tem uma vida bastante sedentária, algo que não mudou desde os seus dias como banqueiro. Não se sente confortável a fazer exercício físico sozinho devido a possuir conhecimentos muito básicos no que toca a exercício físico, nutrição e fisiologia. A vida sedentária que levou ao longo da sua vida originou lesões que dificultam certos movimentos.

Motivação: Devido às suas lesões e falta de atividades aeróbicas, o Sr. Alberto necessita de um sítio onde possa fazer exercício físico enquanto está a ser monitorizado e vigiado por

profissionais da área.

José Luís tem 31 anos, casado, com dois filhos, *personal trainer* de profissão. Leva uma vida normal, gosta do seu trabalho. O José Luís, visto que é *personal trainer* num ginásio, deve ter acesso acerca dos treinos dos seus clientes, podendo observar o tipo de treino de cada cliente, bem como recomendá-los a outro tipo de treino. Pode também verificar se algum cliente apresenta alguma anomalia, podendo assim agir rápido para que nada de mal aconteça.

Motivação: Zé Luís pretende monitorizar os treinos dos seus clientes no ginásio.

Cenários Principais

O Sr. Alberto vê os exercícios que realizou no dia anterior. Após o login, o Sr. Alberto pode deslocar-se até ao seu perfil e ver informação sobre os seus treinos prévios, desta maneira pode saber o que treinos das últimas vezes e se a nível biológico estava tudo bem durante os mesmos.

O José Luís pode avisar o Pedro devido a treino incorreto. O José Luís quando se encontra na página do Pedro repara que repara que o batimento cardíaco está demasiado baixo enquanto ele está a correr, logo o Zé sabe que ou se passa algo com a saúde do Pedro ou ele não está a exercitar o suficiente para cumprir os objetivos pretendidos.

O José Luís pode avisar a Sara devido à falta de treinos. O José Luís quando está a ver os últimos treinos dos seus clientes repara que a Sara já não tem registos de treinos há mais do que uma semana e decide contactar a Sara para saber o que se passa.

3 Arquitetura

Requerimentos chave

Começando pela *web framework*, optámos por usar *Django*, isto porque é escrito em *python*, que facilita a construção de uma *RESTapi*, possui uma interface de administração que nos facilita a gestão dos modelos e é bastante escalável que, dado o tema, é algo de tamanha importância no nosso projeto.

Quanto ao armazenamento de dados, decidimos usar *MongoDB*, tem uma performance bastante elevada, grande escalabilidade, muito flexível usa representação de dados em *JSON*. Todos estes factores levaram a que o *MongoDB* fosse o tipo de base de dados escolhida.

Usamos um *Arduino UNO* com um sensor de movimento PIR. Escolhemos usar *Arduino* em vez de *Raspberry Pi* pela sua simplicidade dado o facto de só usármos um único sensor de movimento.

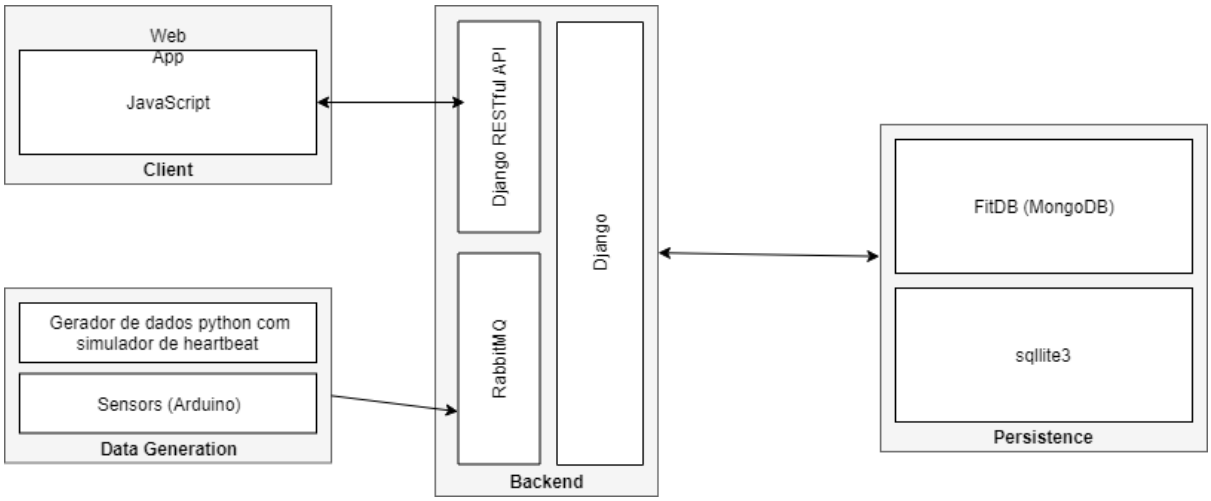
Usamos o Docker, que com o uso de containers, tornou mais fácil a criação e o desenvolvimento da nossa aplicação e dando a garantia que corre em qualquer outro computador. Ao longo do desenvolvimento tivemos problemas que levaram a que

tivéssemos de trabalhar a maior parte do tempo em local host

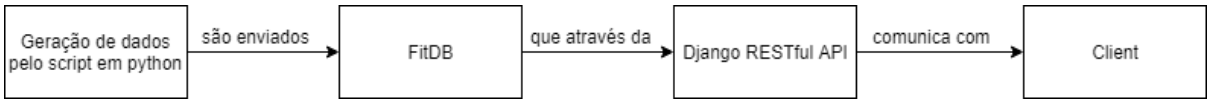
A *Django RESTful API* é uma *API* que usa os métodos *GET*, *PUT*, *POST* e *DELETE*. Tem uma maior flexibilidade visto que permite vários tipos de informação, ao contrário do SOAP que apenas permite XML, e teoricamente é mais simples de usar.

Quanto ao client, decidimos usar JavaScript porque consome menos tempo na resolução de erros e bugs em comparação com, por exemplo, *AngularJS*.

Vista arquitetural



Interação entre módulos

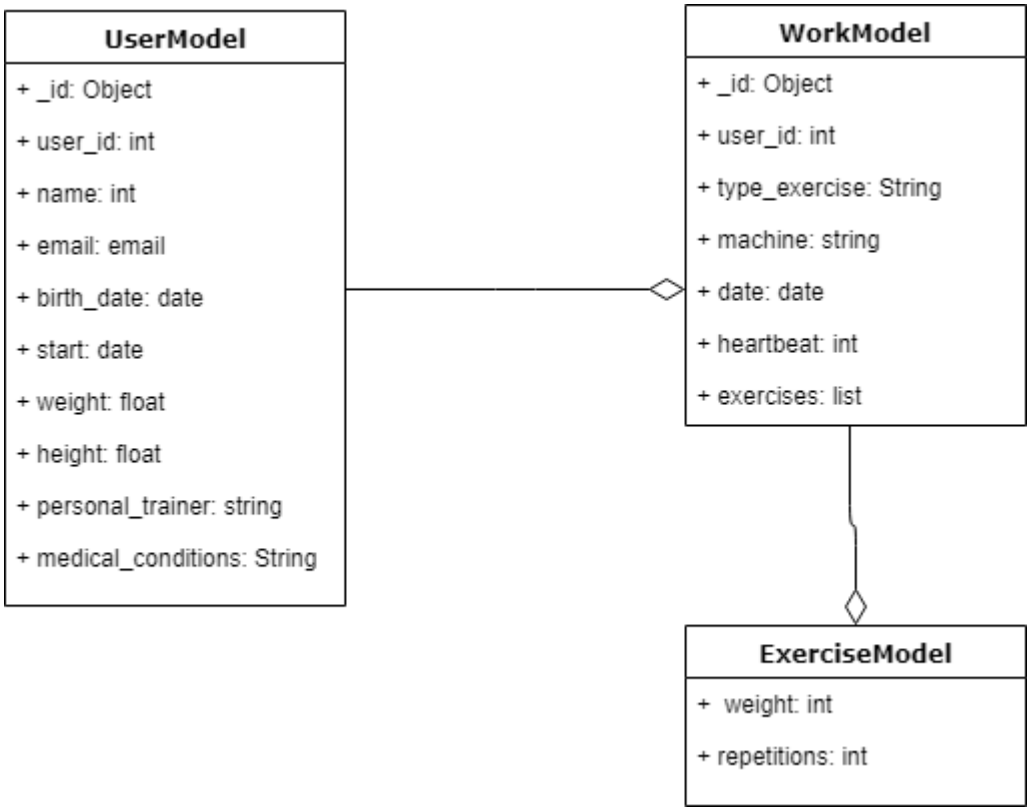


Apesar de na iteração 3 termos um sensor arduino de movimento PIR montado e funcional, optamos por não o utilizar devido à incoerência de dados que ia causar e à dificuldade de simular casos reais. Sendo assim, também não usamos nenhum message broker, neste caso, o RabbitMQ, visto que não usamos o sensor.

4 Perspetiva da Informação

Os conceitos que foram geridos são o UserModel, o WorkModel e o ExerciseModel.

Eles estão relacionados entre eles por relações de composição. O UserModel precisa do WorkModel para saber que exercício vai realizar, e o WorkModel precisa do ExerciseModel para saber como vai ser realizado o exercício (peso, numero de repetições).



5 Bibliografia e referências

https://docs.mongodb.com
https://docs.djangoproject.com/en/3.0/
https://stackoverflow.com
https://devdocs.io/javascript/