

Sistema para controle de calorias diárias

Desenvolvimento Móvel

Cristhian Nunes Dias, Matheus Assmann de Freitas

heycristhian@gmail.com, matheusassmann@hotmail.com

RESUMO: Este trabalho tem como objetivo principal, demonstrar uma aplicação de controle de calorias diárias, de forma que, o usuário possa fazer o apontamento dos alimentos consumidos, diariamente. O sistema calcula a quantidade de carboidratos, proteínas e gorduras em cada registro, para que no final do dia, seja exibido o valor total de calorias ingeridas. Com essa informação, o usuário pode escolher entre manter seu peso (igualar calorias diárias com a quantidade de calorias recomendadas), emagrecer (ficar em déficit com as calorias recomendadas) ou até para ganho de peso, se achar necessário (consumir mais calorias diárias em comparação com as calorias recomendadas). Todas essas funcionalidades do aplicativo, auxiliarão o usuário a ter total controle dos alimentos ingeridos, para que o mesmo possa alcançar seus objetivos.

TECNOLOGIAS ADOTADAS PARA O DESENVOLVIMENTO DA APLICAÇÃO

SPRING BOOT: Segundo Afonso (2017), Spring Boot é uma ferramenta para agilizar configurações iniciais e publicações de aplicações no ecossistema Spring, dito isso, havendo uma rapidez considerável em executar o projeto que é trabalhado. Para desenvolvimento desse projeto, foi utilizado na API Rest feita com a linguagem de programação Java, que tem como objetivo devolver informações para o aplicativo através de requisições HTTP.

GRAPHQL: É uma linguagem que tem como responsabilidade fazer consultas em ambientes de execução (API¹) com uma performance melhorada, mais flexível e intuitiva, Conforme Redhat.

IONIC: Uma SDK (Software Development Kit) de componentes visuais com foco em aplicações híbridas para dispositivos móveis, porém, é possível desenvolver para desktop. É de código aberto, então sempre tem melhorias da comunidade. Sendo assim,

¹ De acordo com Pires (2017), é um grupo de regras e padrões definidos e documentada por uma aplicação para que esta seja utilizada por outras.

facilitando a resolução de problemas. Para executar no dispositivo, o Ionic necessita do Cordova, pois é ele que tem a responsabilidade de instalar no aparelho e posteriormente ser executado. Outra característica boa do Ionic é que ele foi baseado no Angular e com isso traz semelhanças da ferramenta.

MONGODB: Conforme Soares (2016), MongoDB é um banco de dados orientado a documentos (NoSQL), que é de código aberto e que foi desenvolvido em C++. Por ser orientado a documentos, não há a obrigatoriedade de se preocupar com a estrutura de dados, como colunas e tipos de valores. Esse banco é muito semelhante a estrutura JSON, que acaba facilitando a leitura e escrita dos dados. Diferente de um banco de dados relacional, o MongoDB armazena documentos e coleções, onde documentos seriam as tabelas e as coleções seriam as bases de dados.

MongoDB vem com o objetivo de ter uma melhor performance para uma gama gigante de dados em comparação a banco de dados convencionais, porém tudo isso é possível por causa da desnormalização (dados redundantes).

ARQUITETURA DO TRABALHO

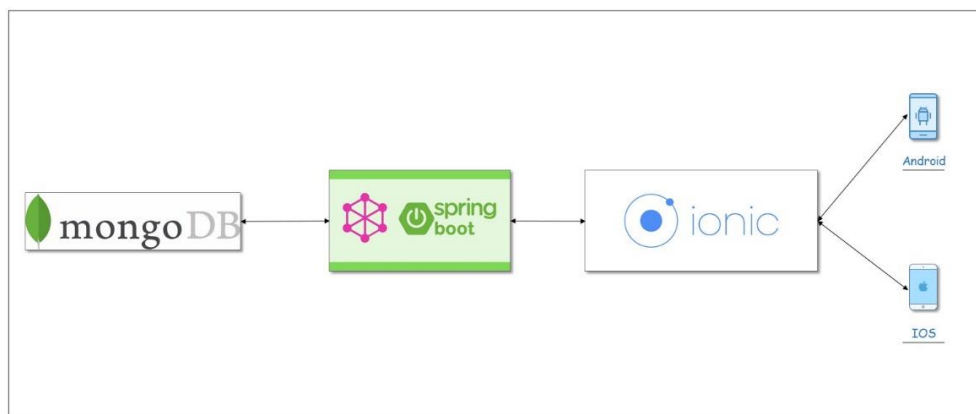


Figura 1: Arquitetura do trabalho

Na **Figura 1**, é exibida a arquitetura do trabalho, onde o MongoDB é onde ficará salvo todo registro que foi inserido no aplicativo. O Spring Boot juntamente com o GraphQL terá o trabalho de gerenciar toda comunicação que precisará ser feita entre o usuário, aplicativo e o bando de dados. Nele também, será armazenada toda regra de negócio, podendo ser utilizada em outras aplicações diferente do Ionic. Já o Ionic, terá responsabilidade de exibir para o usuário interfaces que o mesmo possa interagir. Essa

tecnologia é híbrida, dito isso, pode ser executada em dispositivos móveis com sistema operacional Android ou IOS.

DIAGRAMA DE CLASSE

O diagrama de classe é um diagrama que tem como público alvo pessoas que estão desenvolvendo o projeto, mais especificamente para o desenvolvedor, pois de acordo com Tybel (2016), o Diagrama de Classe é a exibição de uma estrutura do banco de dados, ou seja, estrutura das classes interligadas que são como se fossem modelos para a aplicação. Dito isso, facilita para o programador a estruturação do banco para o próximo passo, que seria o desenvolvimento do aplicativo. Na imagem a seguir, será exibido o diagrama de classe, que contém os principais atributos e métodos que será necessário em cada objeto para a aplicação móvel.

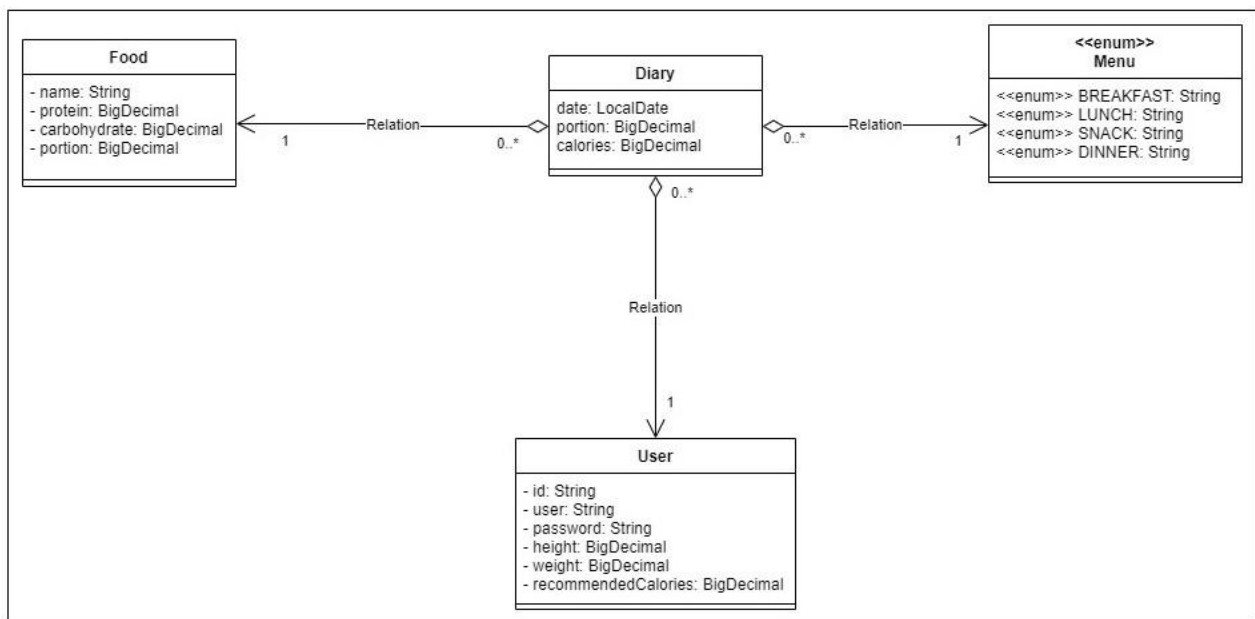


Figura 2: Diagrama de classe do trabalho

REFERÊNCIAS

SOARES, Jhonathan. **O que é MongoDB e porque usá-lo?** Disponível em <https://codigosimples.net/2016/03/01/o-que-e-mongodb-e-porque-usa-lo/>>. Acesso em: 21 jul. 2020.

AFONSO, Alexandre. **O que é Spring Boot?** Disponível em < <https://blog.algaworks.com/spring-boot/>>. Acesso em: 20 jul. 2020.

ANDRADE, Ana. **O que é Ionic?** Disponível em <https://www.treinaweb.com.br/blog/o-que-e-ionic/>>. Acesso em: 05 out. 2020.

PIRES. **O que é API? REST e RESTful? Conheça as definições e diferenças!** Disponível em < <https://becode.com.br/o-que-e-api-rest-e-restful/>>. Acesso em: 20 jul. 2020.

TYBEL, Douglas. **Orientações básicas na elaboração de um diagrama de classes.** Disponível em <<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>>. Acesso em: 02 mar. 2020.