

DOCUMENTO DE ARQUITETURA DE SOFTWARE

Versão 1.4



Universidade Joaquim Nabuco Bacharelado em Sistemas de Informação

Equipe de Desenvolvimento

Claudiomildo Ventura

Leonardo Belas

Lucas Wallis

William Andrey

SOFTWARE / 2019

Paulista – PE

Histórico de Revisões

Versão	Data	Autor	Descrição
1.0	28/04/2019	Lucas Wallis	Elaboração do documento
1.1	01/05/2019	Lucas Wallis	Avaliações e modificações gerais do documento
1.2	20/05/2019	Claudiomildo Ventura	Avaliações e modificações gerais do documento
1.3	20/06/2019	Claudiomildo Ventura	Avaliações e modificações gerais do documento
1.4	20/09/2019	Lucas Wallis	Avaliações e modificações gerais do documento

Índice

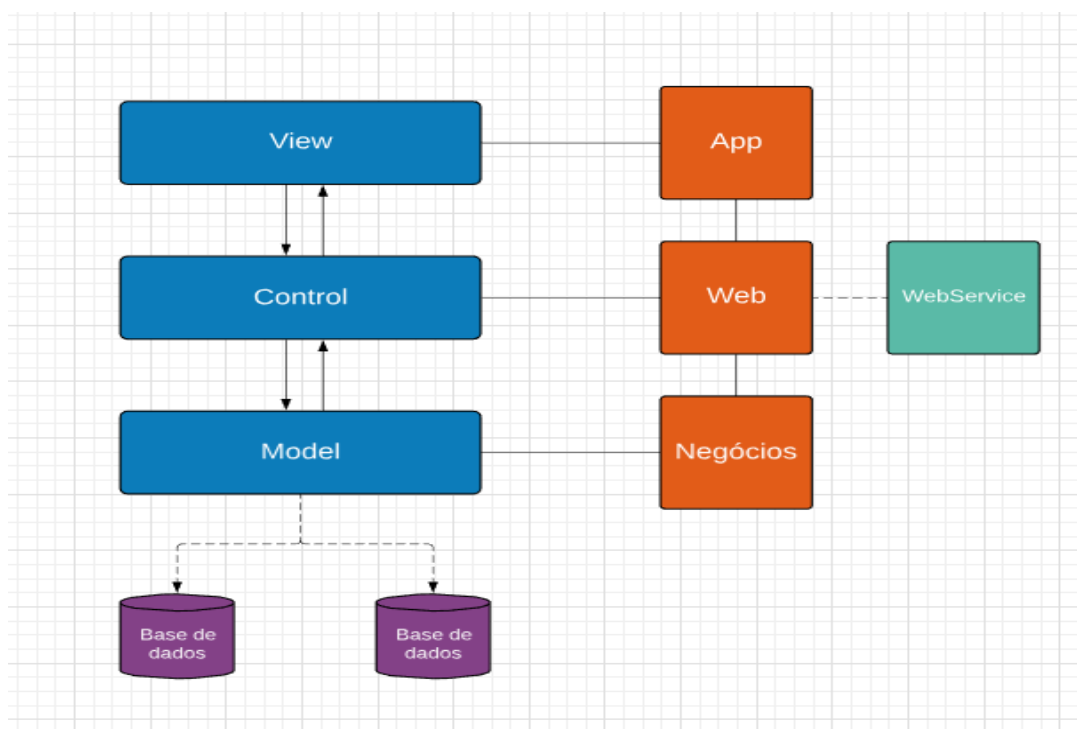
1. Introdução	1
2. Representação Arquitetural	1
2.1. View	2
2.2. Control	2
2.3. Negócios	2
2.4. Dados	2
3. Metas E Restrições Da Arquitetura	3
3.1. Usabilidade	3
3.2. Segurança	3
3.3. Desempenho	3
3.4. Manutenibilidade	4
3.5. Portabilidade	4
4. Visão Da Arquitetura	4
4.1. Visão Lógica	4
4.2. Visão Do Caso De Uso	4
4.3. Visão Geral	5
5. Visão Lógica	5
6. Visão De Implantação	5
7. Visão Do Diagrama De Classe	6
8. Visão De Dados	6
9. Tamanho E Desempenho	6
10. Qualidade	7

1. INTRODUÇÃO

Este documento detalha a arquitetura adotada para o desenvolvimento do projeto Staff Fitness. Este projeto tem a finalidade de possibilitar ao personal trainer, o monitoramento no acompanhamento do cronograma de treino dos seus alunos, na gestão da rotina de treinamento, a exemplo da execução e intensidade, e terá facilidade em monitorar, de forma dinâmica, as atividades para treinamentos livres (funcionais). Desta forma, é descritas as metas, restrições, padronização e visão geral da arquitetura deste sistema.

2. REPRESENTAÇÃO ARQUITETURAL

O padrão escolhido para o desenvolvimento do projeto foi o modelo em camadas. O estilo em camadas organiza o sistema em camadas hierárquicas.



2.1. View

Nela serão processadas entradas e saídas das operações realizadas pelo usuário. Além disso, esta camada mantém uma relação com o usuário e a camada de comunicação. É responsável por usar as informações modeladas para produzir interfaces de apresentação conforme a necessidade.

2.2. Control

Esta camada tem o objetivo de prover a comunicação entre a interface gráfica e as operações relacionadas com o usuário. Nela o usuário funcionará como uma porta de acesso que autentica e autoriza o usuário a estabelecer uma comunicação com o sistema. Interpreta as requisições submetidas pelo usuário e traduz em comandos que são enviados para o (Model) e/ou para a (View).

2.3. Negócios

Esta camada promoverá a interação do usuário com o sistema no sentido executar as atividades disponíveis ao usuário. Desta maneira, proporcionará a modificação e recuperação de informações pertinentes aos processos de negócio. Notifica a view e controls associados quando há uma mudança em seu estado.

2.4. Dados

Esta camada promoverá a interação do usuário com o sistema no sentido de executar as atividades disponíveis ao usuário. Desta maneira, proporcionará a modificação e recuperação de informações pertinentes aos processos de negócio. Notifica a view e controls associados quando há uma mudança em seu estado.

3. METAS E RESTRIÇÕES DA ARQUITETURA

As metas e restrições estabelecidas para o projeto Staff Fitness se encontram no documento de requisitos, no que se refere aos requisitos não funcionais. Desta forma, visando à qualidade do sistema, foram estabelecidos os seguintes critérios, especificados logo abaixo.

3.1. Usabilidade

O sistema deve ser de fácil compreensão e utilização por parte dos usuários.

3.2. Segurança

O sistema deve ser confiável. Sendo assim, deve oferecer proteção de tal maneira que não cause danos aos clientes. Desta forma, deverá ser resistente a intrusões acidentais ou intencionais. Além disso, deve oferecer persistência dos dados armazenados no sistema. Por tanto, é fundamental o comprometimento com a integridade dos dados.

3.3. Desempenho

O sistema deverá estar disponível 24 horas, 7 dias por semana, capaz de fornecer serviços úteis a qualquer instante que seja solicitado. deverá garantir o fornecimento correto dos serviços esperado pelo usuário. Além disso, o tempo de processamento das operações não poderão ultrapassar 30 segundos.

3.4. Manutenibilidade

O sistema deve oferecer facilidade de ser mantido através de manutenções. Sendo assim, serão seguidos padrões durante a sua produção. Como também, garanta que novos requisitos sejam incluídos quando forem necessários.

3.5. Portabilidade

O sistema será capaz de rodar em plataformas que os seus usuários estão acostumados a utilizar. Atualmente, a aplicação mobile está sendo desenvolvido na versão Android.

4. VISÃO DA ARQUITETURA

Os tipos de visões aqui descritas servem para esclarecer quais ferramentas e métodos serão utilizados no desenvolvidos do projeto Staff Fitness.

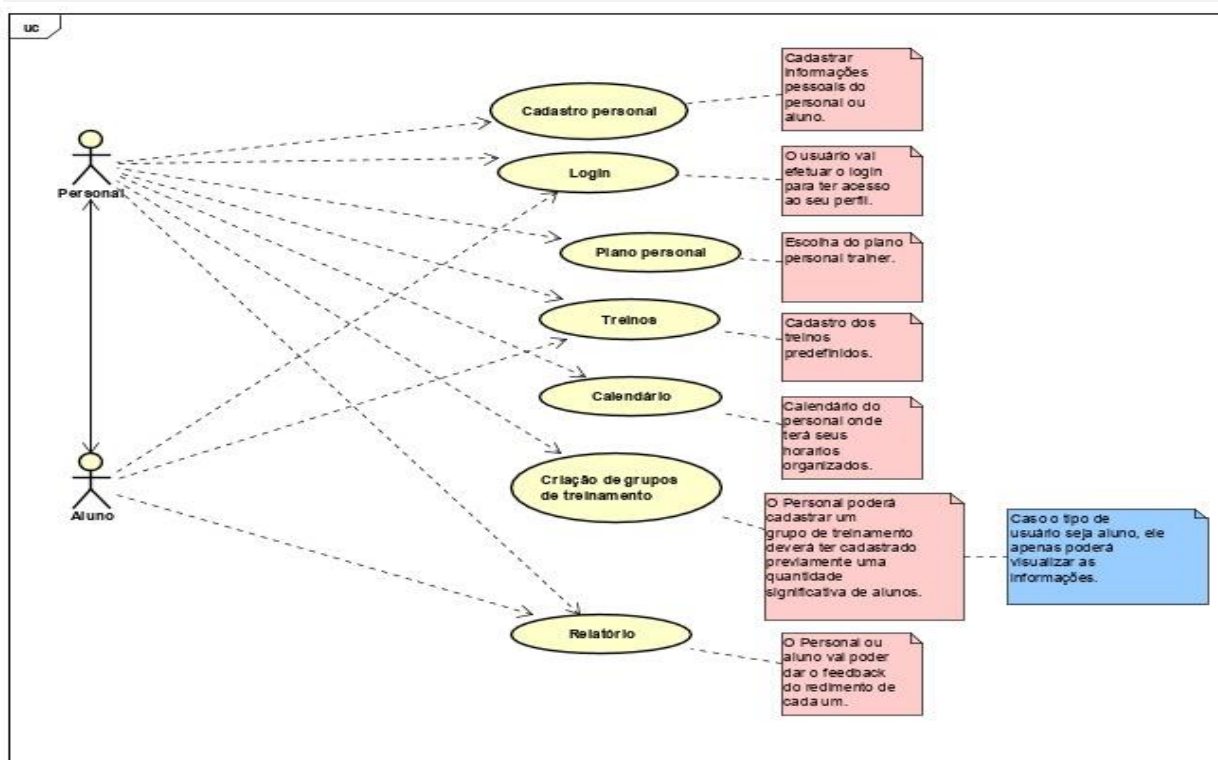
4.1. Visão Lógica

A aplicação Staff Fitness será desenvolvida utilizando as ferramentas React-Native e o Adonis JS para o desenvolvimento da aplicação mobile. e o React JS e o Adonis JS para o desenvolvimento da aplicação Web, o banco de dados que será utilizado será o PostgreSQL.

4.2. Visão do Caso de Uso

As especificações dos casos de uso estão descritas no documento de caso de uso.

4.3. Visão Geral



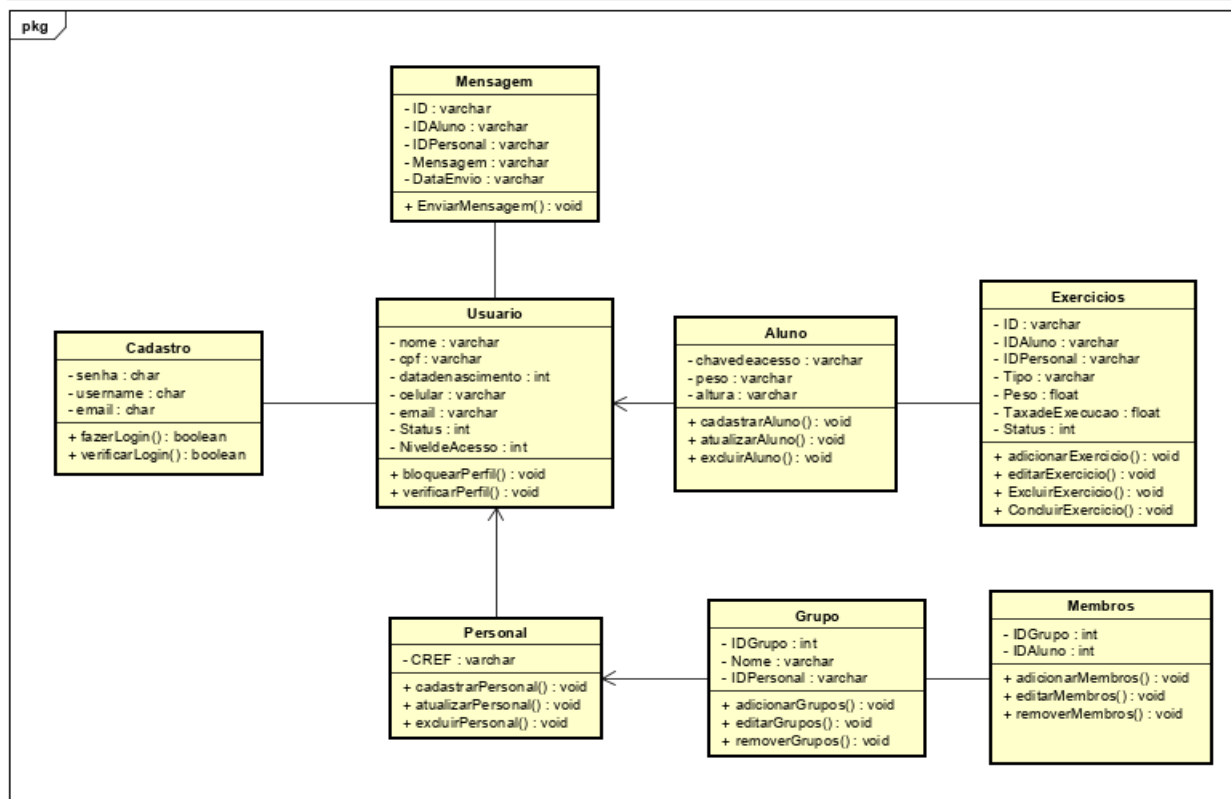
5. VISÃO LÓGICA

A aplicação será executada para os alunos em dispositivos móveis, e para o personal trainer por meio de um browser, com acesso à internet. O programa responderá às requisições feitas pelos usuários cadastrados no sistema, obedecendo ao processamento das regras de negócios, realizando a coleta, a manipulação e a exibição dos dados, que serão enviados e/ou recebidos pelo banco de dados, e retornará aos usuários respostas referentes às suas solicitações.

6. VISÃO DE IMPLANTAÇÃO

Para utilizar a aplicação Staff Fitness é necessário possuir dispositivo com acesso à internet e navegador que ofereça condições de acesso, sendo eles (Mozilla, Internet Explorer e Chrome).

7. VISÃO DO DIAGRAMA DE CLASSE



8. VISÃO DE DADOS

O repositório das informações do sistema será instalado e executado na mesma máquina que ele estiver rodando a fim de manter a persistência e integridade dos dados.

9. TAMANHO E DESEMPENHO

A complexidade do software pode ser considerada mediana devido à necessidade de interação com o ambiente web.

O desempenho poderá ser considerado relativamente baixo devido ao uso de cálculos matemáticos simples e desempenho gráfico pouco elevado, ocasionando um baixo consumo de memória. Foi definido um tempo de resposta máximo como restrição do desempenho desejado.

10. QUALIDADE

A arquitetura permite que novas funcionalidades sejam desenvolvidas e adicionadas ao modelo atual de arquitetura, garantindo a sua extensibilidade.