

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**FITAPP - DESENVOLVIMENTO DE UM SISTEMA WEB PARA PERSONAL
TRAINERS**

LUCAS DE SOUZA FERREIRA

DIAMANTINA, MG

Janeiro de 2019

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS

**FITAPP - DESENVOLVIMENTO DE UM SISTEMA WEB PARA PERSONAL
TRAINERS**

LUCAS DE SOUZA FERREIRA

Orientador(a):

Luis Henrique Rodrigues

Trabalho de Conclusão de Curso apresentado ao
Curso de BACHARELADO EM SISTEMAS
DE INFORMAÇÃO, como parte dos requisitos
exigidos para a conclusão do curso.

DIAMANTINA, MG

Janeiro de 2019

FERREIRA, LUCAS DE SOUZA.

FitApp - Desenvolvimento de um sistema web para personal trainers /
LUCAS DE SOUZA FERREIRA - Janeiro de 2019.

47.p

1. Engenharia de Software; 2. Desenvolvimento Web; 3. Laravel; 4.
Métodos ágeis I. Título.

CDU xxx.xx

FITAPP - DESENVOLVIMENTO DE UM SISTEMA WEB PARA PERSONAL
TRAINERS

LUCAS DE SOUZA FERREIRA

Orientador(a):

Luis Henrique Rodrigues

Trabalho de Conclusão de Curso apresentado ao
Curso de BACHARELADO EM SISTEMAS
DE INFORMAÇÃO, como parte dos requisitos
exigidos para a conclusão do curso.

APROVADO em ____ / ____ / _____.

Prof. Dr. Marcus Vinícius Carvalho Guelpeli – UFVJM

Prof. Me. Átila Rocha Trindade – UFVJM

Prof. Me. Luis Henrique Rodrigues – UFVJM

Alguns sacrifícios são sólidos; o resto
são os meus.

Mikhail Nekhemievich Tal.

RESUMO

A dificuldade de gerir de forma eficiente um negócio autônomo somada a crescente demanda por cuidados com a saúde por meio de exercícios físicos, desafia personal trainers.

Este trabalho propõe uma análise do cenário desses profissionais visando identificar suas necessidades, propor uma solução envolta na metodologia atual de desenvolvimento de software, para o caso específico de um personal trainer na cidade de Diamantina/MG. O desenvolvimento deste sistema visa a melhoria da gestão dos processos recorrentes destes profissionais possibilitando o controle de fichas de avaliação física dos alunos, de suas fichas de treinamentos e um controle de pagamentos. Para o desenvolvimento do software, utilizou-se do framework para PHP, Laravel, banco de dados MySQL para composição do back end, e HTML somado ao JavaScript e ao CSS para parte de front end. A estrutura proposta para esse desenvolvimento foi a Scrum, sendo seus 4 artefatos gerados ao longo do texto. Ao fim o sistema foi testado, validando todas as funcionalidades propostas.

O sistema atingiu todos os requisitos propostos **Palavras-chave:** Laravel, Desenvolvimento web, Engenharia de Software.

ABSTRACT

The difficulty of efficiently managing an autonomous business coupled with the growing demand for physical exercise challenges personal trainers. A professional equipped with an information system that provides optimal management of resources, is able to deliver better results to its customers. This work proposes an analysis of the scenario of these professionals in order to identify their needs, to propose a solution involved in the current methodology of software development, for the specific case of a personal trainer in the city of Diamantina / MG. For software development, was used the framework for PHP, Laravel, MySql database for backend composition, and HTML added to JavaScript and CSS for front end part.

The difficulty of efficiently managing an autonomous business coupled with the growing demand for physical exercise challenges personal trainers. paragraph This work is an analysis made from a software support standard, for the specific case of a personal trainer in Diamantina / MG. The development of this system improves students performance, making it possible to control students' physical evaluation cards, their training records and a payment control. To do software development, use framework for PHP, database, MySQL database for backend composition, and HTML for JavaScript and CSS for front end part. The system meets all the proposed requirements and follows the agile model of development. The proposed version for this model was a scrum, being its 4 artifacts generated throughout the text. The system has been tested, validating all your needs and guaranteeing its qualit

Keywords: PHP, Information Systems, Laravel, Development, Software Engineering

SUMÁRIO

1 INTRODUÇÃO.	1
1.1 Justificativa	1
1.2 Objetivos	2
1.2.1 Objetivo Geral	2
1.2.2 Objetivos Específicos	2
1.3 Organização do texto	2
2 REFERENCIAL TEÓRICO	3
2.1 Engenharia de Software	3
2.1.1 Projetos de Software	4
2.1.2 Métodos Ágeis	5
2.1.3 Scrum	5
2.2 Engenharia Web	7
2.2.1 Cliente Servidor	7
2.2.2 Web 2.0	7
2.3 MVC (Model-View-Controller)	7
2.4 PHP 7	9
2.5 CSS	10
2.6 JavaScript	10
2.7 HTML 5	11
2.8 MySQL	11
2.9 Bootstrap 4.0	12
2.10 Laravel	12
2.11 Framework	12
3 PROJETO DO SOFTWARE	14
3.1 Product Backlog	14
3.2 Descrição do cenário	14
3.2.1 Entrevista	14
3.3 Proposta de Software	15
3.3.1 Proposta de Software FitApp	15
3.3.2 Apreciação da Proposta	16
3.4 Requisitos Funcionais	17

3.4.1	Casos de Uso	17
3.4.1.1	Login	19
3.4.1.2	Cadastrar	19
3.4.1.3	Listar Ficha de treino	20
3.4.1.4	Listar Avaliações	20
3.4.1.5	Listar Pagamentos	21
3.4.1.6	Cadastrar Ficha de Treino	21
3.4.1.7	Cadastrar Pagamento	22
3.4.1.8	Cadastrar Dados Pessoais	23
3.4.1.9	Cadastrar Planos	23
3.4.1.10	Cadastrar Avaliação	24
3.4.1.11	Excluir Ficha de Treino	24
3.4.1.12	Excluir Pagamento	25
3.4.1.13	Excluir Dados Pessoais	25
3.4.1.14	Excluir Planos	26
3.4.1.15	Excluir Avaliação	26
3.5	Ferramentas de desenvolvimento	27
3.5.1	Versionamento GIT	27
3.5.2	Mysql Workbench	27
3.5.3	Microsfot Visual Code	27
3.5.4	Gitkraken	27
3.5.5	Trello	27
3.5.6	Composer	28
3.5.7	Atlassian	28
3.6	Product Backlog	29
3.6.1	Sprint Backlog	29
3.6.2	Sprint 1 - Configuração do Ambiente	29
3.6.3	Sprint 2 - Criação do Banco de dados (Models)	30
3.6.4	Sprint 3 - Criação dos controladores (Controllers)	32
3.6.5	Sprint 4 - Criação das telas (views)	34
3.6.6	Sprint 5 - Criação das permissões e ajustes finais	35
4	RESULTADOS	36
4.1	Site	36

4.2	Sistema FitApp	38
4.3	Testes	39
5	CONSIDERAÇÕES FINAIS	43
5.1	Trabalhos Futuros	43
5.2	Principais Desafios	44
	REFERÊNCIAS	45

1 INTRODUÇÃO.

As aplicações web se integraram à vida moderna de tal forma que se tornaram inerentes ao cotidiano. O processo de desenvolvimento dessas, deve ser empregado de maneira a favorecer a relação dos utilizadores da tecnologia e seus idealizadores, a fim de sempre produzir soluções com mais qualidade, em menor custo e tempo. A era da informação traz consigo a possibilidade de controlar e verificar resultados a partir de dados e métricas atuais de avaliação das atividades físicas, bem como confrontar no decorrer desse período os exercícios realizados frente a seus resultados.

Os personal trainers trabalham de maneira individual e personalizada com cada cliente, sendo o atendimento a seus alunos sua principal atividade. Ao acompanhar a rotina e gestão das atividades profissionais é constatado que a gerência dos artefatos gerados durante o processo de avaliação e acompanhamento tem fundamental importância para o êxito da tarefa. A forma de se manter estes registros pode ser otimizada quando aplicadas as técnicas modernas de sistemas de informação, para se extrair destes, a eficiência gerada na gestão de dados e consequente transformação em informação.

O ramo de atividades de personal trainers, se caracteriza como um mercado de profissionais autônomos que além de exercer seu trabalho devem cuidar e promover sua imagem, como fator para destacar o diferencial competitivo do seu serviço. A gestão dos documentos gerados pelos atendimentos aos alunos, como ficha de avaliações físicas, de treinos e pagamentos recebidos, quando bem gerida agrega valor a seu labor.

Com a percepção desse cenário, uma solução possível é avaliá-lo amplamente e propor, dentro das métricas da metodologia de software, a entrega de uma aplicação que atenda as necessidades diretas do profissional, seguido de uma proposta de solução ágil dentro no universo web que supra as demandas elegidas.

1.1 Justificativa

A quantidade de dados gerados pelas atividades profissionais na vida moderna, e a necessidade de obter destes os mais diversos tipos de informação, compõem um desafio aos personal trainers. A falta de gestão, perda de informação e consequente não gerência das atividades de acompanhamento de seus alunos e devidos registros, resultam em impactos negativos frente as expectativas de resultados.

A conjectura atual do mercado desses profissionais é de um ambiente altamente competitivo. Uma gestão eficiente do negócio, fator indispensável para aquisição de clientes, viabiliza uma administração mais profissional. Aliado a apresentação, um diferencial na comunicação com seus alunos, a aplicação, associa satisfação a responsabilidade de exposição dos dados de maneira transparente e acessível.

Agrupando estas necessidades e desafios, um sistema de informação adequado fornecerá os resultados desejados para boa gestão destes dados, aliado a uma forma profissional no tratamento de dados e sua consequente transformação em informação. Bem como pode fornecer uma maneira estruturada de se proceder as atividades profissionais de um personal trainer, agregando também a disponibilidade e acessibilidade aos documentos gerados durante o processo.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver uma solução web com portal e sistema de gerenciamento para um personal trainer, que considere o cenário deste profissional, apresente soluções para aprimorar a gestão de seu trabalho, e contribua para a sua atividade.

1.2.2 Objetivos Específicos

- Facilitar a gestão das Fichas de Treinamento e Avaliações.
- Organizar e centralizar a informação da gestão Financeira.
- Disponibilizar um site para a divulgação dos serviços de Personal Trainer.
- Estudar o processo de desenvolvimento de Software.

1.3 Organização do texto

O texto deste trabalho será dividido em um capítulo para o levantamento do referencial teórico, Capítulo 2, onde serão apresentados e definidos os conceitos pertinentes ao trabalho. No Capítulo 3, será detalhado o projeto do software proposto. Em sequência, no Capítulo 4, são apresentados os resultados, testes e telas do sistema. O Capítulo 5, apresenta as principais telas do sistema. O Capítulo 6 traz as considerações finais. O Capítulo 7, finaliza o trabalho ao elencar possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este estudo aplica os conceitos da Engenharia Web utilizando as linguagens PHP 7, CSS e Javascript. O sistema gerenciador de banco de dados (SGBD) utilizado foi o MySQL. O framework Laravel, juntamente com o padrão Model Viewer Controller (MVC) foi aplicado para otimizar o tempo de desenvolvimento do software e facilitar sua manutenção e adição de novos módulos. A linguagem de marcação HTML e a biblioteca JQuery foram utilizados para gerar a parte de front end da aplicação e telas de interação com o usuário. O processo de engenharia segue os modelos ágeis do scrum adaptados para equipe de uma pessoa.

2.1 Engenharia de Software

Softwares, continuam a ser a tecnologia mais importante no cenário mundial e também são um ótimo exemplo da lei das consequências não intencionais. Há cinquenta anos, ninguém poderia prever que o software iria se tornar uma tecnologia indispensável para negócios, ciência e engenharia. Em um período inferior a apenas meio século, viabilizaram a criação de novas tecnologias como, por exemplo, engenharia genética e nanotecnologia, a extensão de tecnologias existentes tal como telecomunicações, impactaram indústrias seculares como a indústria gráfica e tornaram-se a força motriz por trás da revolução do computador pessoal (PRESSMAN, 2011, p,30).

A definição de engenharia de software padronizada pela IEEE é um processo de aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software, ou ainda, o processo de estudo de abordagens que segue estes preceitos (IEEE, 1990).

O processo de desenvolvimento de software em seu início, tem como função entender o domínio da aplicação. Deve-se começar com a identificação de uma necessidade a ser atendida, geralmente, advinda de um problema a ser resolvido. Para tal é necessário fazer uma coleta inicial de requisitos, parte onde a conversa com os stakeholders acontece, e a aplicação começa a tomar forma pelos objetivos a serem alcançados. Após esse levantamento, os requisitos são agrupados de maneira lógica e coerentes em grupos classificados por sua semelhança de maneira ainda não estruturada. (SOMMERVILLE, 2011).

Ainda segundo Sommerville (2011) a obtenção e avaliação de requisitos contêm as seguintes atividades:

- Entender o domínio da aplicação: processo no qual as características são isoladas e agrupadas em sistemas que contêm características semelhantes.
- Coleta de requisitos: É o processo de interagir com os stakeholders do sistema para descobrir seus requisitos. A compreensão do domínio se desenvolve mais durante essa atividade;
- Classificação: Essa atividade considera o conjunto não estruturado dos requisitos e os organiza em grupos coerentes.
- Resolução de conflitos: Quando múltiplos stakeholders estão envolvidos, os requisitos apresentarão conflitos. Essa atividade tem por objetivo solucionar esses conflitos.
- Definição das prioridades: Em qualquer conjunto de requisitos, alguns serão mais importantes do que outros. Esse estágio envolve interação com os stakeholders para a definição dos requisitos mais importantes.
- Verificação de requisitos: Os requisitos são verificados para descobrir se estão completos e consistentes e se estão em concordância com o que os stakeholders desejam do sistema.

2.1.1 Projetos de Software

Ao que tange as características da engenharia de software, o software se tornou peça fundamental da vida moderna e há a tendência de crescimento do número de usuários. Ainda concordam que as demandas e requisitos que a sociedade necessita, criam uma pressão por um software com cada vez mais qualidade e agilidade em seu processo de desenvolvimento. Todos esses fatores consideram desempenho e satisfação por parte de seus usuários. SOMMERVILLE (2011) e PRESSMAN (2011).

Para Paula Filho (2013) uma parte do modelo de ciclo de vida de um software é o projeto, sendo definido por uma data de início, fim e uma equipe. Dentro dos modelos modernos de software outros recursos podem aparecer. Ainda segundo Paula Filho os projetos devem se dividir em etapas chamadas marcos, que vão servir para o controle de seu andamento, sendo o próprio produto a ser entregue um marco, de conclusão do projeto.

Pelas constatações de PRESSMAN (2011) a um projeto software tem como conceito um olhar sistematizado, padronizado que demanda disciplina e pode ser quantificado. Seu objetivo é a produção de software bem como sua manutenção e operação. (SOMMERVILLE, 2011) complementa que softwares são abstratos e intangíveis não sendo portanto, regidos pelas leis da física ou por processos comuns de manufatura. Seguindo Pressman (2011), os softwares modernos assumem dois papéis, o papel de produto onde disponibiliza o poder dos computadores numa rede ou de maneira local e também o papel de veículo, onde se distribui o produto. Aliada a sua natureza dual, tem-se a visão de que um software além de cumprir seu papel de sistematização de informações também pode servir para a divulgação de si próprio. O software de maneira geral é o ativo com o maior valor, ainda justifica, fazendo o paralelo, entre a era da informação na qual vivemos e os sistemas de informação que abordam os softwares (PRESSMAN, 2011).

2.1.2 Métodos Ágeis

Baseado no Manifesto Ágil, documento descreve suas principais características, pode-se afirmar que a prioridade se dá aos indivíduos e interações em conjunto com o software em pleno funcionamento, traz para a rotina de trabalho a colaboração com o cliente e torna a equipe, e conseqüentemente o software, apto a responder às mudanças. (BECK, 2019). Sendo estes princípios norteadores do desenvolvimento das aplicações que seguem este modelo.

2.1.3 Scrum

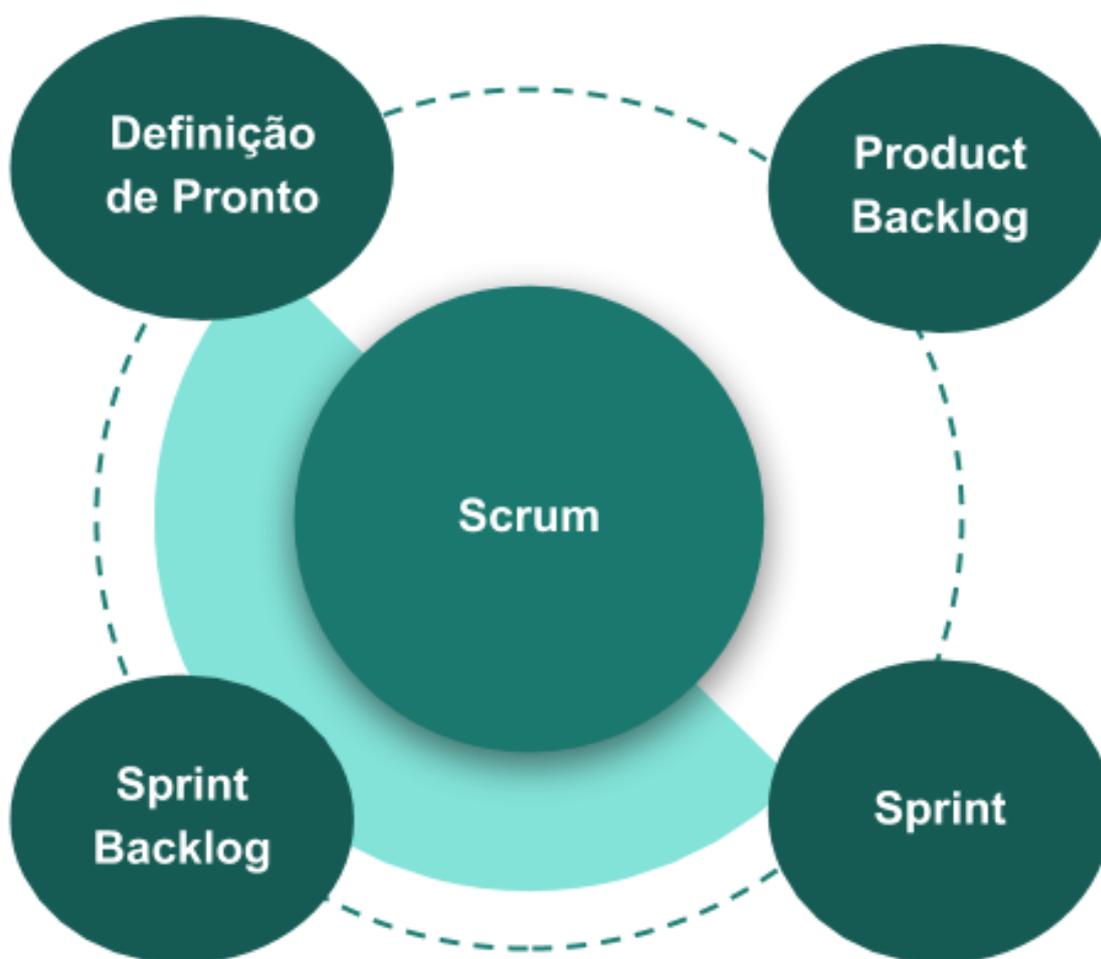
A estrutura de desenvolvimento scrum possibilita o desenvolvimento ágil de software sendo seus pilares a transparência, inspeção e adaptação (DUARTE, 2019). A transparência para o autor coincide com o conceito proposto de que para se conhecer e saber se as atividades estão sendo feitas de maneira apropriada, é preciso reforçar o sentimento de pertencimento diariamente, bem como que todos se posicionem a respeito das atividades. Já a inspeção serve para identificar as ameaças e oportunidades presentes em cada relato. Por fim, a adaptação que se inicia ao se identificar ameaça ou oportunidade, todos colaboram para se adaptar e empenhar pelo sucesso da atividade. (AUDY, 2013).

Ainda segundo o autor os 4 artefatos básicos propostos pelo scrum, e ilustrados na figura 2.1.3, são:

1. Product Backlog: Lista de requisitos mínimos que vão cumprir a tarefa desejada, agregando valor ao produto.

2. **Sprint Backlog:** Lista de requisitos desejáveis para as próximas etapas e semanas, que advém do product backlog com uma organização lógica do do que vai ser feito e a expectativa de quando.
3. **Incremento:** Resultado obtido após cada Sprint concluído. O incremento é a forma e o resultado da próxima entrega a ser efetuada dentro do ciclo de sprints.
4. **Definição de "Pronto":** Para cada funcionalidade requerida é necessário acordar entre as partes envolvidas o conceito de pronto, pois para um programador da equipe, "pronto" pode significar o código rodando em sua máquina, enquanto para o cliente o mesmo conceito pode ser definido como a aplicação em produção respondendo de maneira adequada, como ilustra a figura 2.1.3:

Figura 2.1.3 - Artefatos do Scrum



Fonte: O autor.

2.2 Engenharia Web

2.2.1 Cliente Servidor

A arquitetura cliente servidor é definida como uma forma de padronização para web, visando acesso de recursos remotos em servidores/provedores por parte da aplicação do cliente, que pode ser um browser ou um cliente de e-mail, que é responsável pelos procedimentos de requisições e recebimento de respostas, e pode ser diretamente controlado no lado do cliente pela interação do usuário. Ainda segundo o autor esse modelo de arquitetura, este, é usado na web para aplicações multi usuários.(BERTOCCO et al., 1998).

2.2.2 Web 2.0

A web 2.0 surgiu como uma evolução natural da web e uma tentativa de oferecer serviços mais dinâmicos interativos (CUNHA; JESUS, 2012). A interação é o novo paradigma da web com aplicações que permitem níveis cada vez mais dinâmicos de influência recíproca de dois ou mais usuários e está cada vez mais presentes na vida das pessoas (TORRES-SALINAS; DELGADO-LÓPEZ-CÓZAR, 2009). Ainda seguindo os autores, definem as principais aplicações que a web 2.0 pode trazer:

- Aplicações para depósito: Definição geral de aplicação para repositórios ou ferramentas de compartilhamento de apresentações, vídeos e documentos.
- Canal Central: Representado pelos blogs onde serão relacionados as aplicações de depósito com uso de ferramentas para redes sociais.
- Canal de difusão: Definidos pelo uso dos sites de ferramentas de rede sociais e microblogs.

2.3 MVC (Model-View-Controller)

A forma de se desenvolver códigos aliando código em html, que serve para a construção da visualização da página (front-end), juntamente com códigos de lógica e inserção do banco de dados no mesmo arquivo com as regras de negócio e o códigos de programação, ainda é muito utilizada (HANSEN; FOSSUM, 2005).

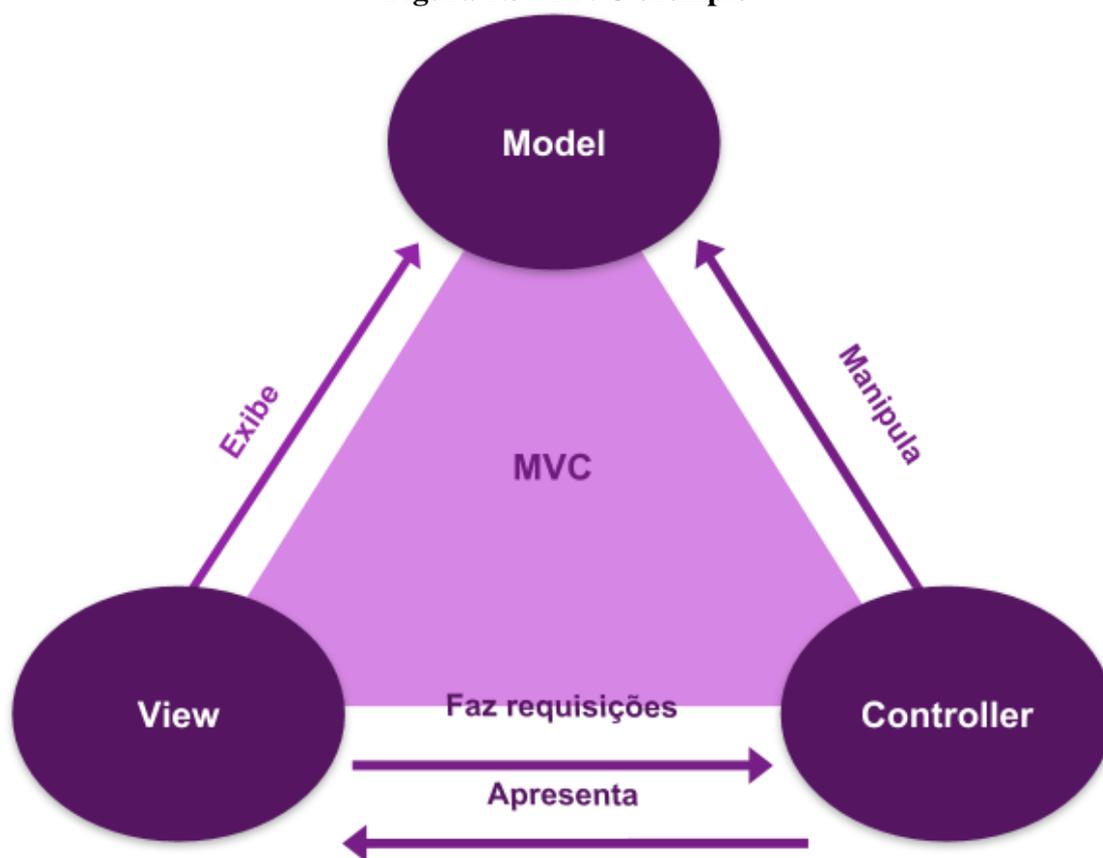
Como uma alternativa a esse método, a arquitetura em 3 camadas tipicamente conhecida como MVC, propõe uma forma de se reestruturar aplicações web, o que permite sua

manutenção e modularização do código gerado. Para (W3C, 2018) e (HANSEN; FOSSUM, 2005) o modelo pode ser conceituado como :

- **Model:** Representa um modelo de dados que geralmente possui uma classe e uma tabela no banco de dados associada, fornece abstração para se manipular os dados.
- **View:** Reapresenta a tela que será exibida ao usuário com os elementos de interação e dados solicitados.
- **Controller:** Contém a lógica da aplicação e todas as regras para se manipular os dados e gerar as telas advindas das requisições feitas.

A organização típica de pastas e estrutura de arquivos criadas pelo MVC, para (SANCHEZ; ALTHMANN, 2017) é composta de: uma pasta models contendo todos os modelos, uma pasta views contendo todas as telas para o usuário e uma pasta controllers contendo os controladores, assim como ilustra a figura 2.3. Ainda segundo o autor, e em concordância com os conceitos apresentados por (HANSEN; FOSSUM, 2005), essa organização facilita a criação de rotas para o acesso aos recursos pela aplicação.

Figura 2.3 - MVC exemplo



Fonte: O autor

2.4 PHP 7

A sigla PHP significa em tradução livre “pré-processador de hipertexto” e é uma linguagem de programação criada em 1994 disponibilizada em 1995 por Rasmus Lerdorf (Lerdorf 2019) e é predominante na web, com atuação no backend, ao lado do servidor. Para Converse e Park (2003) o PHP é uma linguagem que tem compatibilidade com o lado do servidor e serve para se criar scripts interpretados a medida que os usuários fazem solicitações, e, simultaneamente incorpora trechos de códigos HTML normais. Segundo os desenvolvedores, o PHP Group, em sua página oficial <http://php.net/>, as duas maiores vantagens de se usar o PHP são : a facilidade de uso e a possibilidade dos comandos rodarem no servidor e serem devolvidos em HTML diretamente para o cliente. Outro fator mencionado em sua documentação, esta, não se limita a funções simples, tendo recursos suficientes para rodar aplicações web complexas e atender às necessidades dos desenvolvedores, (PHP Documentação,2019). A linguagem conta com módulos que trabalham diretamente com banco de dados e facilitam as operações e persistência dos dados. A versão mais recente disponível é a 7.3, disponibilizada em seu site oficial e distribuída de

forma gratuita, assim como, também é, uma iniciativa de código livre. A linguagem tem hoje uma predominância de 78.9% da fatia do mercado web segundo a W3techs, empresa que se propõe a fazer estatísticas das aplicações e tecnologias presentes na internet.

Tais fatores aliados ao conhecimento do autor sobre a tecnologia, adquirido em projetos passados e da versatilidade que uma linguagem com alta taxa de utilização no mercado, sua gratuidade e o código livre somam os norteadores para escolha dessa tecnologia.

2.5 CSS

Em tradução livre CSS significa, "folhas de estilos em cascata" definidos por (SIKOS, 2011) como uma especificação da W3C para arranjo de estilos visuais de uma página. Tem como características típicas fontes, cores, planos de fundo, margens, bordas e camadas. (SIKOS, 2011) e (W3C, 2019b) concordam na definição de que o CSS tem seu uso comumente em páginas HTML na web possibilitando dividir os estilos das informações em arquivos separados mantendo a hierarquia entre eles e também que o CSS descreve elementos de forma textual por seus atributos e características.

Os principais benefícios do uso dessa linguagem são listados por (W3C, 2018)

- Controle de interface em diferentes documentos em um único arquivo;
- Controle de diferentes interfaces para diferentes dispositivos ;
- Precisão para manter a mesma interface para diferentes navegadores;
- Melhorias na acessibilidade com a possibilidade de “esconder” elementos da tela para usuários sem problemas de visão, mas manter os mesmos elementos acessíveis para leitores de tela;
- Formulários com aparência diferente do padrão do sistema operacional;
- Menor consumo de banda para usuário e servidor;
- Inúmeras técnicas dinâmicas que não poderiam ser utilizadas em tabelas;

2.6 JavaScript

Linguagem de programação multi paradigma, dinâmica e interpretada. Para definir a linguagem, (FLANAGAN, 2013) esclarece que JavaScript é a linguagem da web que

integra sites e navegadores modernos. Também ressalta que, pela sua característica multiplataforma que abrange computadores de mesa, consoles de jogos, tablets, smartphones entre outros dispositivos que possuem interpretadores JavaScript, esta se tornou essencial ao desenvolvedor. Para os navegadores (GRILLO, 2019) elucida que o JavaScript abre a possibilidade de se criar programas que podem ser embutidos em uma página HTML e funcionar como uma linguagem de programação capaz de alterar a própria estrutura do HTML, o que gera uma forma do usuário interagir com a página sem necessariamente gerar tráfego na rede, ou depender da resposta de um servidor. Com a abordagem de (CROCKFORD, 2019), elucida-se que o nome JavaScript é uma marca da corporação Oracle e que a tecnologia se chama formalmente ECMA Script. Ainda destaca-se que a orientação à objetos e funções de primeira classe são suportadas. A linguagem foi inventada por Brendan Eich, quando era funcionário da Netspace tendo no mesmo ano, sua primeira aparição. O desenvolvimento de sua especificação atualmente é regido pelas publicações de novos padrões da ECMA Script e se encontra na revisão 262.

2.7 HTML 5

Linguagem de marcação de hipertexto, padrão utilizado para se criar páginas em navegadores modernos oferecendo suporte à criação de botões, parágrafos e estrutura básica de um site. Teve como seu uso original uma forma de se descrever semanticamente documentos científicos. Segundo (W3C, 2019c) que oferece referência de todas as versões do HTML, a definição que se aplica é de uma linguagem de uso geral e é justamente sua finalidade genérica oferecer facilidade para se desenvolver páginas modernas e adaptáveis à diversas realidades, sendo seu uso indispensável para páginas web, principalmente quando aliada à outras tecnologias.

2.8 MySQL

Para desenvolver uma base de dados sólida, um SGBD (sistema gerenciador de banco de dados) é indispensável, com a propriedade de um software livre e de bom reconhecimento no mercado, o Mysql supre todas as características desejadas. Sendo uma plataforma que usa a SQL (Linguagem de Consulta Estruturada) torna-se de fácil acesso e sintaxe de simples uso. Seu desempenho em bases de dados pequenas é altamente satisfatório. Outro fator importante segundo (Oracle, 2019), é que a plataforma implementa o modelo de entidade relacionamento, o que é ideal para se trabalhar com aplicações que utilizam o MVC e orientação à objetos.

2.9 Bootstrap 4.0

Segundo a própria equipe de desenvolvimento via página oficial da tecnologia (W3C, 2019a), é o framework mais popular para projetos na web e possibilita a criação de modelos de referência para web sites. Sua gama de conteúdo fornece botões, estilos, menus, parágrafos e o que mais for necessário para estilos visuais e responsivos de um web site. O framework é mantido de forma aberta e colaborativa pelo GitHub, e fornece padronização de estilos de código aberto para criação rápida e responsiva de sites.

2.10 Laravel

O framework Laravel atualmente na versão 5.7, fornece um padrão para se criar aplicações web no modelo MVC (Otwell, Taylor B., 2019). Ainda segundo seus desenvolvedores, este possui uma filosofia de uso e implementação própria. Para eles, o Laravel tenta tornar o processo de criação de aplicações web agradável, favorecendo a parte criativa do desenvolvedor, que se referem como artesãos, e oferece ferramentas de uso comum em aplicações web como, rotas, cache, autenticação e conexão com banco de dados. Além de fornecer uma interface de comunicação via CLI (command line interface) para executar diversas ações integradas entre o Laravel e a tela de comandos.

2.11 Framework

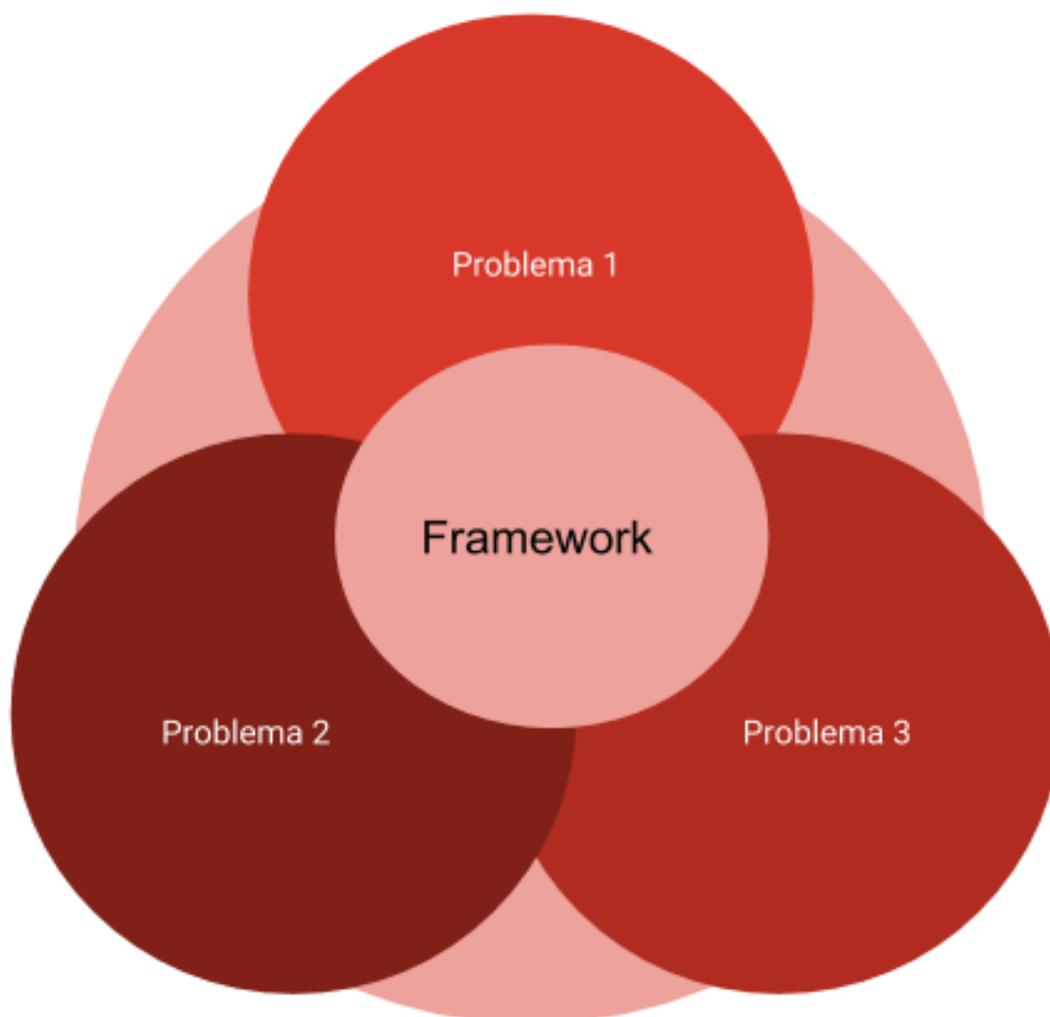
Frameworks, de maneira ampla, reúnem conceitos usados para se resolver problemas comuns a uma área de conhecimento, como ilustra a figura 2.11, sendo no caso específico de um software, uma estrutura reutilizável com diversos aspectos semi prontos que podem ser aprofundados para adaptar ao problema abordado (FAYAD; SCHMIDT, 1997). Descrições de frameworks são encontradas de maneira vasta em descrições de interfaces para o usuário, e tem como exemplo o framework MVC. Seu detalhamento contém instruções para especificar, implementar, usar e reutilizar código, bem como sua apresentação e padrões de implementação (CAMPBELL; ISLAM, 1993). Para (FAYAD; SCHMIDT, 1997), as características contidas em um framework que favorecem o reuso e aproveitamento das funções semi prontas implementadas são:

- Padrões de projeto: Os padrões representam soluções contornando os problemas do desenvolvimento do software dentro de um contexto.
- Frameworks definem aplicações semiprontas enquanto que bibliotecas algoritmos, Os frameworks estendem os benefícios de bibliotecas ao inverter a estrutura de

controle que redireciona para o framework o controle da aplicação, enquanto a biblioteca e funções devolvem o controle para o objeto , ou método que a chamou.

- Componentes: São um dos tipos abstratos de dados que podem ser plugados para dar forma a aplicações completas. O relacionamento entre estruturas e componentes é altamente sinérgico, com nenhum subordinado ao outro.

Figura 2.11 - Modelo de Framework



Fonte: O Autor

3 PROJETO DO SOFTWARE

3.1 Product Backlog

Essa seção irá descrever e especificar o processo. Cada um dos passos, divididos em tópicos, que seguem o modelo básico do scrum apoiado pelo Kanban, também são demonstrados, além de outras ferramentas para documentação do desenvolvimento.

3.2 Descrição do cenário

A fim de melhor descrever o cenário em que a aplicação se encontra, foi feito o primeiro passo para gerar o product backlog, artefato do scrum e uma entrevista com o cliente, um personal trainer que atua na cidade de Diamantina/MG. Nessa entrevista foi gerado um documento com as principais características do ambiente descrito pelo profissional. Como a proposta de software será feita diretamente para sua área de atuação, foi explicado que para fins do desenvolvimento ele assumirá o papel de cliente, onde, além de demandar por recursos e elencar problemas, ficará a disposição para retirar possíveis dúvidas e promover esclarecimentos quanto a pontos que fossem de suma importância ao software, o que atende a estrutura ágil que integra o cliente no processo de desenvolvimento de software.

3.2.1 Entrevista

O cliente, 30 anos, educador físico há oito com ensino superior completo, trabalha como personal trainer em academias da região, possui, durante a realização deste trabalho, doze alunos entre fixos oito que são mensalistas e variáveis quatro que fazem aulas esporádicas durante a semana. Os atendimentos acontecem nas academias onde ele orienta, acompanha e corrige a execução de exercícios de seus alunos. Para gerir seus clientes ele usa uma agenda física de papel, onde controla sua agenda de atendimento. O controle e a gestão financeira é feita por aplicativo de comunicação WhatsApp por onde são feitas as cobranças, negociações de preço e eventuais comprovantes de depósitos e transferência bancária. Demais assuntos profissionais são, em sua maioria, tratados de forma direta e pessoal quando abordado por um cliente.

O histórico de avaliações físicas de seus alunos fica salvo em papel sob a tutela do aluno, tal como histórico de treinos conhecidos como “ficha de treino”. O marketing é feito na rede social Instagram, onde após acordo verbal com o aluno, são postados conjuntos de fotos com o tema "antes e depois", evidenciando as mudanças de peso e forma física.

O cliente relata sua insatisfação com o tempo dispendido com respostas a perguntas frequentemente feitas e com aquisição de novos alunos. Outro importante fator informado é a recorrência os resultados que os históricos e a evolução dos treinamentos é perdida ou não vira uma informação clara. Dado o fato de ser um profissional autônomo e único responsável pelas atividades, o controle financeiro não é eficiente.

Em função dos problemas da gestão de tempo demais recursos necessários para melhor condução dos serviços prestados, o educador físico, não é capaz de expandir sua base de clientes.

3.3 Proposta de Software

Após a primeira entrevista com o cliente em que o cenário foi descrito e detalhado, foi feita uma releitura do documento procurando evidenciar as características do ambiente, do profissional e seu negócio, enumeradas abaixo:

1. Cliente possui 30 anos, formação de nível superior exerce a profissão a 8.
2. Trabalha como personal trainer atendendo em academias da região de Diamantina.
3. Atende a clientes mensalistas e clientes recorrentes.
4. Ausência de controle, métodos de arquivo ou criação de histórico das ações, avaliações, pagamentos ou mesmo fichas de treino de seus alunos.
5. Possui smartfone e algum domínio de aplicativos comuns na web.
6. Marketing realizado através de fotos no Instagram pessoal.
7. Ingerência do tempo para levantamento de informações dos alunos.
8. Perda de informação dos alunos.

Com base numa análise das relações entre as características do cenário bem como suas limitações foi elaborada uma primeira proposta de Software.

3.3.1 Proposta de Software FitApp

O software Fitapp propõe automatizar, informatizar e sistematizar a rotina de trabalho de um educador físico. Uma vez que no mundo moderno onde o tempo e a informação são

ativos preciosos, o programa Fitapp coloca a disposição do educador físico uma solução para suprir suas necessidades de gestão de alunos, financeira e profissional. Conta com funcionalidades como: Módulo avaliações: que gerencia os dados da avaliação corporal e de composição física do aluno, calculando a partir desses o IMC, taxa metabólica, composição corporal e índice de gordura; Módulo ficha de treinos: uma ficha personalizada onde o educador pode cadastrar cada exercício separado por dia e podendo evidenciar número de repetições, séries, carga e tempo de execução. Portal: site com diversas informações sobre a forma como o trabalho acontece, preços, planos, horários, relatos de alunos, parceiros, dicas de musculação e uma região de perguntas frequentes (FAQ) que servirá como apoio, para seus atuais alunos e futuros clientes. Para os alunos, o aplicativo oferece todo o suporte de visualização e acompanhamento de avaliações físicas, fichas de treino, evoluções e objetivos. Adequando-se a variedade de dispositivos que é ofertada ao mercado o programa suporta variados tamanhos de tela e é compatível com os navegadores modernos, Chrome, Firefox, Opera, Safari.

Custo do software: Gratuito sob condição de aceitar usar como meu trabalho de conclusão de curso, sem divulgação do código ou nome dos envolvidos, apenas o trabalho científico de desenvolvimento. Tempo previsto : Será adicionada uma previsão de tempo após realizar a análise de requisitos.

3.3.2 Apreciação da Proposta

Após enviada a proposta de software inicial contendo os detalhamentos dos módulos e funcionalidades associadas, foi feito um novo encontro com o cliente, onde foi lida a proposta, e explicou-se melhor alguns termos. Em sequência começou a discussão sobre essas expressões e expectativas em relação à proposta. O cliente aceitou a proposta de desenvolvimento, concordou com todas as funcionalidades levantadas e adicionou os seguintes itens:

- Em função de sua escolha, ele prefere lançar os pagamentos no sistema manualmente, para melhor gestão da informação do meio de recebimento dos valores.
- Deve-se gerar gráficos e relatórios básicos para facilitar o entendimento dos dados lançados e seus históricos.
- Modificar a parte de agenda pelo atendimento a consultar e disponibilidade na semana.

- Os planos vão conter valores de referência por hora/aula dada para que os alunos possam estimar os custos. A negociação do valor final é feita pelo profissional.

O desenvolvedor concordou com os pontos e acordou-se a adição dos pontos mencionados na documentação do Software assim como a implementação das funcionalidades. Foi estabelecido pelas partes, para fins de referencia, um prazo inicial de 3 meses para entrega do primeiro protótipo, sendo portanto, a previsão de entrega para janeiro 2019. O cliente será isento de custos com o desenvolvedor. Apenas arcará com os custos de hospedagem e domínio, que são requisitos mínimos de aplicações web em ambiente de produção. Os valores durante a realização do trabalho foram estimados em R\$120,00 por ano pagos diretamente para empresa. A escolha desta, será definida em comum do desenvolvedor com o cliente na etapa de implantação.

Todos os textos, slogan e marca do site, são responsabilidade do cliente. O sistema será entregue com textos de teste para que as funcionalidades possam ser aferidas. O cliente então, entregará os textos e logomarca para serem adicionados no site.

O protocolo utilizado para calcular os índices especificados no sistemas seguem o me todo proposto pelo Protocolo de Faulkner (1968) que se utiliza de 4 dobras cutâneas sendo elas Tríceps; subescapular; supra-ilíaca e adome para o cálculo da gordura corporal. Outro índice calculado é a massa corporal abreviado em IMC que efetua a razão do peso pelo quadrado da altura de um individuo.

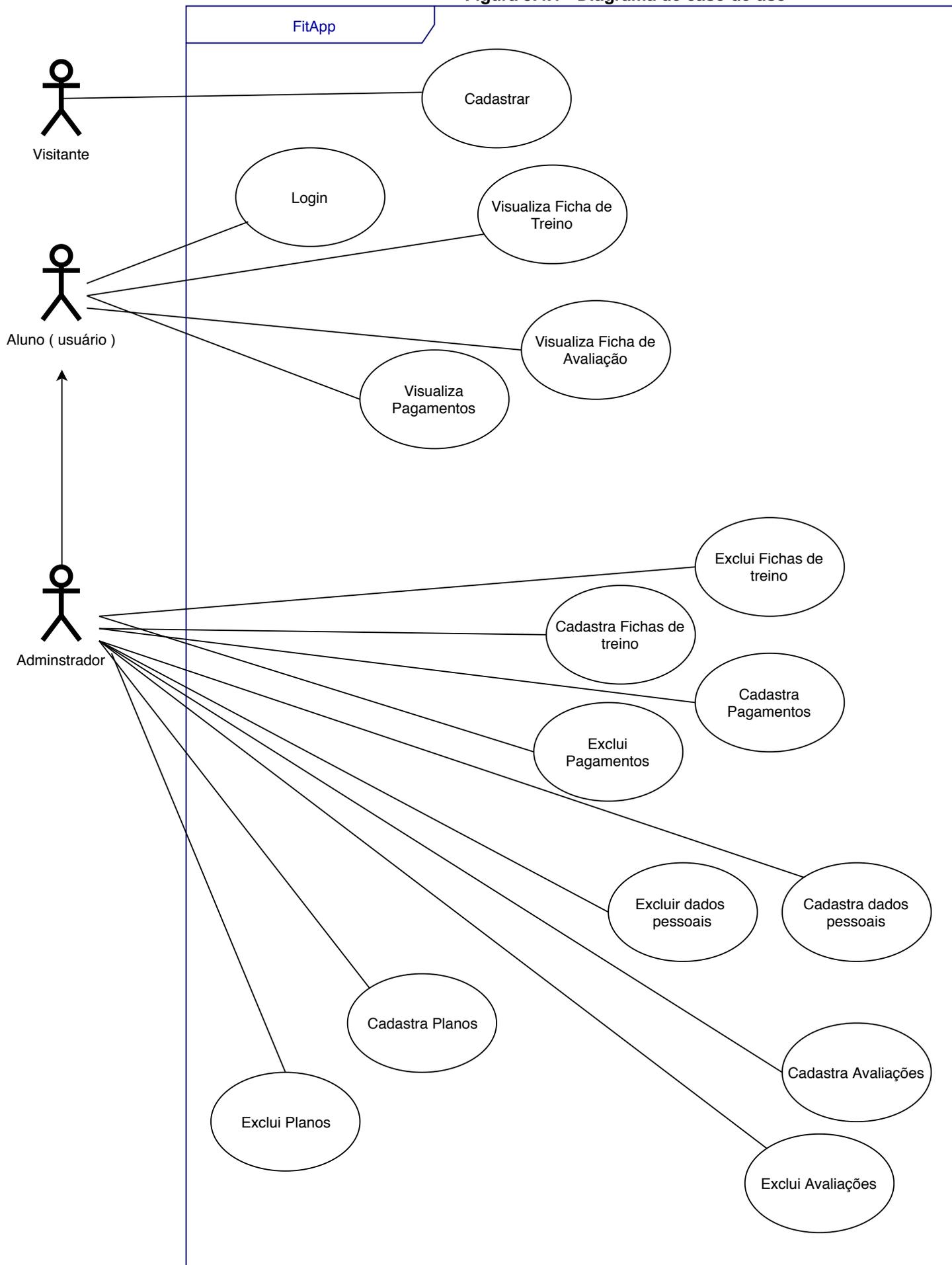
3.4 Requisitos Funcionais

Um conjunto de requisitos, é composto pela soma de todas as demandas e expectativas a serem atendidas. Dentro do contexto de requisitos de software, os requisitos funcionais compõem a parte dos requisitos ligadas diretamente a uma função do software.

3.4.1 Casos de Uso

Para se descrever as principais funcionalidades do sistema, foi feito uma modelagem via diagrama UML (Unified Modeling Language), onde são descritos os casos de uso, figura 3.4.1. Com o objetivo de se definir os tipos de usuários conhecidos no diagrama como atores e quais funções devem estar disponíveis para estes. A partir desse diagrama todas os comportamentos necessários do software podem ser visualizadas.

Figura 3.4.1 - Diagrama de caso de uso



Fonte: O Autor

Para cada caso de uso verificado, foi feito seu detalhamento, tópicos 3.4.1.1 até 3.4.1.15, em termos de: seus atores, breve descrição, pré-condições, processo executado, Saída e a descrição do fluxo principal junto ao alternativo quando necessário. Este detalhamento visa descrever de maneira textual como o sistema deve se comportar tomando por base ações dos usuários descritas no diagrama de caso de uso.

3.4.1.1 Login

- **Ator(es):** Aluno e administrador.
- **Descrição do caso de uso:** Descreve o processo de entrada no sistema.
- **Pré-condições:** O usuário precisa estar cadastrado e com a tela de login aberta.
- **Processo:** O sistema verifica se as credenciais de acesso são aceitas.
- **Saída:** Mensagem de sucesso e redirecionamento para pagina inicial.
- **Fluxo Principal:**
 1. O usuário insere os dados de login solicitados
 2. O sistema verifica as credenciais - Fluxo alternativo
 3. O sistema redireciona o usuário para página inicial
- **Fluxo Alternativo:** e-mail ou senha incorretos
 1. O sistema verifica as credenciais.
 2. O sistema exibe uma mensagem de erro apropriada.
 3. O caso de uso volta ao passo 1 do fluxo principal.

3.4.1.2 Cadastrar

- **Ator(es):** Visitante.
- **Descrição do caso de uso:** Descreve o processo de cadastro no sistema.
- **Pré-condições:** O visitante precisa acessar a página de cadastro.
- **Processo:** O sistema cria um novo usuário e insere no banco.
- **Saída:** Mensagem de sucesso, e redirecionamento para pagina inicial.

- **Fluxo Principal:**

1. O usuário insere os dados de cadastro solicitados.
2. O sistema valida os dados - Fluxo alternativo .
3. O sistema cria o novo usuário e o insere na base de dados.
4. O sistema redireciona o usuário para página inicial.

- **Fluxo Alternativo:** e-mail ou senha incorretos

1. O sistema verifica as credenciais.
2. O sistema exibe uma mensagem de erro apropriada.
3. O caso de uso volta ao passo 1 do fluxo principal.

3.4.1.3 Listar Ficha de treino

- **Ator(es):** Administrador e Aluno.
- **Descrição do caso de uso:** Descreve o processo de visualizar fichas de treino.
- **Pré-condições:** O usuário precisa acessar a página de visualizar fichas de treino.
- **Processo:** O usuário carrega as informações devidas e encaminha para o usuário.
- **Saída:** Todas as informações sobre a ficha de treino.

- **Fluxo Principal:**

1. O usuário acessa a página de visualizar fichas.
2. O sistema redireciona o usuário para página, apresentando as informações apropriadas.

3.4.1.4 Listar Avaliações

- **Ator(es):** Administrador e Aluno.
- **Descrição do caso de uso:** Descreve o processo de visualizar avaliações.
- **Pré-condições:** O usuário precisa acessar a página de visualizar avaliações.
- **Processo:** Carrega as informações devidas e encaminha para o usuário.

- **Saída:** Todas as informações sobre as Avaliações (no caso do administrador visualiza de todos os alunos).
- **Fluxo Principal:**
 1. O usuário acessa a página visualizar avaliações.
 2. O sistema redireciona o usuário para página, apresentando as informações apropriadas.

3.4.1.5 Listar Pagamentos

- **Ator(es):** Administrador e Aluno.
- **Descrição do caso de uso:** Descreve o processo de visualizar Pagamentos.
- **Pré-condições:** O usuário precisa acessar a página de visualizar Pagamentos.
- **Processo:** Carrega as informações devidas e encaminha para o usuário.
- **Saída:** Todas as informações sobre Pagamentos (no caso do administrador visualiza de todos os alunos).
- **Fluxo Principal:**
 1. O usuário acessa a página Pagamentos.
 2. O sistema redireciona o usuário para página, apresentando as informações apropriadas.

3.4.1.6 Cadastrar Ficha de Treino

- **Ator(es):** Administrador.
- **Descrição do caso de uso:** Descreve o processo de cadastro da Ficha de Treino .
- **Pré-condições:** O administrador precisa acessar a página de cadastro.
- **Processo:** O sistema cria uma nova Ficha de Treino e insere no banco.
- **Saída:** Nova ficha de treino.
- **Fluxo Principal:**
 1. O usuário escolhe o aluno que receberá a ficha.

2. O usuário insere os dados de cadastro Ficha de Treino.
3. O sistema valida os dados - Fluxo alternativo.
4. O sistema cria a Ficha de Treino e o insere na base de dados.
5. O sistema redireciona o usuário para página visualizar treino(com o novo treino aberto).

- **Fluxo Alternativo:** Dados fora do parâmetro

1. O sistema verifica os dados, validação simples de números e strings.
2. O sistema exibe uma mensagem de erro apropriada.
3. O caso de uso volta ao passo 2 do fluxo principal.

3.4.1.7 Cadastrar Pagamento

- **Ator(es):** Administrador.

- **Descrição do caso de uso:** Descreve o processo de cadastro do Pagamento .

- **Pré-condições:** O administrador precisa acessar a página de adicionar Pagamento.

- **Processo:** O sistema cria um novo Pagamento e insere no banco.

- **Saída:** Novo Pagamento adicionado.

- **Fluxo Principal:**

1. O usuário escolhe o perfil do aluno que realizou o pagamento.
2. O usuário insere os dados de Pagamento.
3. O sistema valida os dados - Fluxo alternativo
4. O sistema cria o Pagamento e o insere na base de dados.
5. O sistema redireciona o usuário para página visualizar Pagamento (com o novo Pagamento aberto).

- **Fluxo Alternativo:** Dados fora do parâmetro

1. O sistema verifica os dados, validação simples de números e strings.
2. O sistema exibe uma mensagem de erro apropriada.
3. O caso de uso volta ao passo 2 do fluxo principal.

3.4.1.8 Cadastrar Dados Pessoais

- **Ator(es):** Administrador.
- **Descrição do caso de uso:** Descreve o processo de cadastro dos Dados Pessoais.
- **Pré-condições:** O administrador precisa acessar a página de adicionar Dados Pessoais.
- **Processo:** O sistema cria um novo conjunto de Dados Pessoais e insere no banco.
- **Saída:** Dados Pessoais adicionados.
- **Fluxo Principal:**
 1. O usuário escolhe o aluno que receberá o conjunto de Dados Pessoais.
 2. O usuário insere os Dados Pessoais.
 3. O sistema valida os dados - Fluxo alternativo.
 4. O sistema Dados Pessoais e o insere na base de dados.
 5. O sistema redireciona o usuário para página visualizar Dados Pessoais(com os novos Dados Pessoais aberto).
- **Fluxo Alternativo:**

Dados fora do parâmetro

 1. O sistema verifica os dados, validação simples de números e strings.
 2. O sistema exibe uma mensagem de erro apropriada.
 3. O caso de uso volta ao passo 2 do fluxo principal.

3.4.1.9 Cadastrar Planos

- **Ator(es):** Administrador.
- **Descrição do caso de uso:** Descreve o processo de cadastro de Plano.
- **Pré-condições:** O administrador precisa acessar a página de adicionar Plano.
- **Processo:** O sistema cria um novo Plano e insere no banco.
- **Saída:** Avaliação adicionada.
- **Fluxo Principal:**

1. O usuário insere os Dados do Plano.
2. O sistema valida os dados - Fluxo alternativo
3. O sistema cria uma nova Avaliação e a insere na base de dados.
4. O sistema redireciona o usuário para página visualizar Avaliação (com os novos Dados Avaliação aberta).

3.4.1.10 Cadastrar Avaliação

- **Ator(es):** Administrador.
- **Descrição do caso de uso:** Descreve o processo de cadastro da Avaliação.
- **Pré-condições:** O administrador precisa acessar a página de adicionar Avaliação.
- **Processo:** O sistema cria uma nova Avaliação e insere no banco.
- **Saída:** Avaliação adicionada.
- **Fluxo Principal:**
 1. O usuário escolhe o aluno que receberá a Avaliação.
 2. O usuário insere os Dados da Avaliação.
 3. O sistema valida os dados - Fluxo alternativo.
 4. O sistema cria uma nova Avaliação e a insere na base de dados.
 5. O sistema redireciona o usuário para página visualizar Avaliação (com os novos Dados Avaliação aberta).
- **Fluxo Alternativo:** Dados fora do parâmetro
 1. O sistema verifica os dados, validação simples de números e strings.
 2. O sistema exibe uma mensagem de erro apropriada.
 3. O caso de uso volta ao passo 2 do fluxo principal.

3.4.1.11 Excluir Ficha de Treino

- **Ator(es):** Administrador.
- **Descrição do caso de uso:** Descreve o processo de Excluir da Ficha de Treino .
- **Pré-condições:** O administrador precisa acessar a página de Excluir Ficha de Treino.

- **Processo:** O sistema deleta uma Ficha de Treino e atualiza o banco.
- **Saída:** Nova ficha de treino.
- **Fluxo Principal:**
 1. O usuário escolhe o aluno e posteriormente a Ficha de Treino que será apagado pelo código.
 2. O usuário clica na caixa do código e confirma a exclusão.
 3. O sistema aplica a rotina de soft delete no item
 4. A tela é atualizada sem o Plano deletada.

3.4.1.12 Excluir Pagamento

- **Ator(es):** Administrador.
- **Descrição do caso de uso:** Descreve o processo de Excluir Pagamento.
- **Pré-condições:** O administrador precisa acessar a página de Excluir Pagamento.
- **Processo:** O sistema deleta um Pagamento e atualiza o banco.
- **Saída:** Pagamento deletado.
- **Fluxo Principal:**
 1. O usuário escolhe o Pagamento que será apagado pelo código.
 2. O usuário clica na caixa do código e confirma a exclusão.
 3. O sistema aplica a rotina de soft delete no item
 4. A tela é atualizada sem o Pagamento deletado.

3.4.1.13 Excluir Dados Pessoais

- **Ator(es):** Administrador.
- **Descrição do caso de uso:** Descreve o processo de Excluir Dados Pessoais.
- **Pré-condições:** O administrador precisa acessar a página de Excluir Dados Pessoais.
- **Processo:** O sistema deleta e atualiza o banco.

- **Saída:** Dados Pessoais deletados.
- **Fluxo Principal:**
 1. O usuário escolhe o aluno que terá os Dados Pessoais apagados pelo código.
 2. O usuário clica na caixa do código e confirma a exclusão.
 3. O sistema aplica a rotina de soft delete no item
 4. A tela é atualizada sem os Dados Pessoais deletados.

3.4.1.14 Excluir Planos

- **Ator(es):** Administrador.
- **Descrição do caso de uso:** Descreve o processo de excluir um Plano.
- **Pré-condições:** O administrador precisa acessar a página de adicionar Plano.
- **Processo:** O sistema deleta o plano e atualiza o banco.
- **Saída:** Avaliação deletada.
- **Fluxo Principal:**
 1. O usuário escolhe o Plano pelo código.
 2. O usuário clica na caixa do código e confirma a exclusão.
 3. O sistema aplica a rotina de soft delete no item
 4. A tela é atualizada sem o Plano deletado.

3.4.1.15 Excluir Avaliação

- **Ator(es):** Administrador.
- **Descrição do caso de uso:** Descreve o processo de Excluir a Avaliação.
- **Pré-condições:** O administrador precisa acessar a página de Excluir Avaliação.
- **Processo:** O sistema deve Excluir uma Avaliação e atualizar o banco.
- **Saída:** Avaliação Excluída.
- **Fluxo Principal:**

1. O usuário escolhe a avaliação pelo código .
2. O usuário clica na caixa do código e confirma a exclusão.
3. O sistema aplica a rotina de soft delete no item
4. A tela é atualizada sem a avaliação deletada.

3.5 Ferramentas de desenvolvimento

3.5.1 Versionamento GIT

Protocolo de versionamento de texto utilizado para gerenciar código, criar versões do sistema e permitir o trabalho sempre do ponto mais atual do desenvolvimento (GIT, 2018). Possui diversas opções de código aberto e de gratuita utilização.

3.5.2 Mysql Workbench

Ferramenta de interface gráfica amigável para manipulação de dados e tabelas de banco de dados (Oracle, 2018b), oferecida junto ao pacote padrão de instalação do MySQL.

3.5.3 Microsot Visual Code

Editor de código leve e simples da Microsot com diversas funcionalidades desejáveis ao desenvolvimento em PHP. Destaque para o plug-in de integração com HTML e sintaxe Blade para criação das views e da integração com o PHP para recursos de auto completar nomes de classes, variáveis e namespaces do Laravel (Microsoft, 2018).

3.5.4 Gitkraken

Ferramenta de interface gráfica de código aberto para uso do protocolo GIT que facilita o gerenciamento utilizando de elementos visuais, para controle das versões de software e melhor visualização do seu andamento (Oracle, 2018a).

3.5.5 Trello

Ferramenta que implementa a técnica de Kanban para gerenciamento de atividades, online e gratuita (Trello, 2018). Dispõe de um quadro onde serão dispostas e controlados todos os sprints do software.

3.5.6 Composer

Gerenciador de dependências PHP que auxilia a instalação de módulos que adicionam funcionalidades a linguagem, ferramenta gratuita disponibilizada para PHP versão 5 ou maior (Composer, 2018).

3.5.7 Atlassian

Ferramenta online para armazenamento de código compartilhado em tempo real compatível com protocolo GIT (Atlassian and Bitbucket, 2018). Permite que o código fique acessível de maneira privada e fornece controle de versão e histórico de mudanças do código.

3.6 Product Backlog

No contexto deste trabalho, o autor promoveu adaptações no método, o único responsável por todas as tarefas relativas à produção dos artefatos e também o contato com o cliente. A comunicação será dividida em entrevistas iniciais e uma entrega final para o cliente, de forma que as atualizações incrementais dos sprints serão alvo de testes por parte do desenvolvedor. A tabela 3.6 contém a separação dos sprints detalhada.

Tabela 3.6 - Sprint Backlog

Controle	Tarefa	Importância(%)	Estimativa (dias)	Atividade
Sprint 1	Configuração do Ambiente	90	2	Instalar as aplicações e configuração básica de funcionamento, criar contas nas ferramentas online
Sprint 2	Criação do Banco de dados	70	10	Criar todos os modelos no Laravel e no banco de dados
Sprint 3	Criação dos controladores	60	10	Criar todos os controladores para receber as requisições com a lógica necessária
Sprint 4	Criação das telas	50	10	Criar todas as telas necessárias para apresentação e interação com o usuário
Sprint 5	Criação das permissões, e ajustes finais	50	5	Configurar a autenticação e nível de acesso dos usuários

Fonte: O autor.

O tempo para a execução do trabalho foi estimado em função dos dias necessários, com a restrição de um expediente de quatro horas diárias de trabalho.

3.6.1 Sprint Backlog

3.6.2 Sprint 1 - Configuração do Ambiente

O primeiro passo para realização do sprint foi a separação de todos os requisitos das ferramentas instaladas e configuração para uso. O ambiente de desenvolvimento utilizado

foi um computador de mesa com 16GB de memória RAM, processador de 4 núcleos e armazenamento SSD com o sistema MacOS 10.13.6.

Após todos os softwares instalados e configurações básicas, o Laravel foi testado para garantir que suas funcionalidades estivessem ativas como sugere sua documentação. A tela inicial do Laravel deve aparecer no navegador.

Para executar o Laravel, utilizou-se o comando "php artisan serve" e seu ambiente foi iniciado na porta 8000 da máquina local.

Ao finalizar essa parte, as contas no Trello e Atlassian foram criadas e configuradas para trabalhar com o controle do código e das tarefas.

Com as ferramentas habilitadas, verificou-se que a previsão de 2 dias foi ideal para a realização da tarefa, sendo entregue no período estipulado e todas as ferramentas necessárias ao projeto ativadas.

3.6.3 Sprint 2 - Criação do Banco de dados (Models)

O segundo sprint tem como foco garantir a implementação do banco de dados. Em acordo com o modelo MVC, os modelos são as representações das tabelas do banco de dados. Todos foram criados com o auxílio da ferramenta Artisan, inclusa no Laravel que tem o comando específico para a criação destes e migrações para o banco de dados sendo o comando: "php artisan make: model "nome do modelo-m".

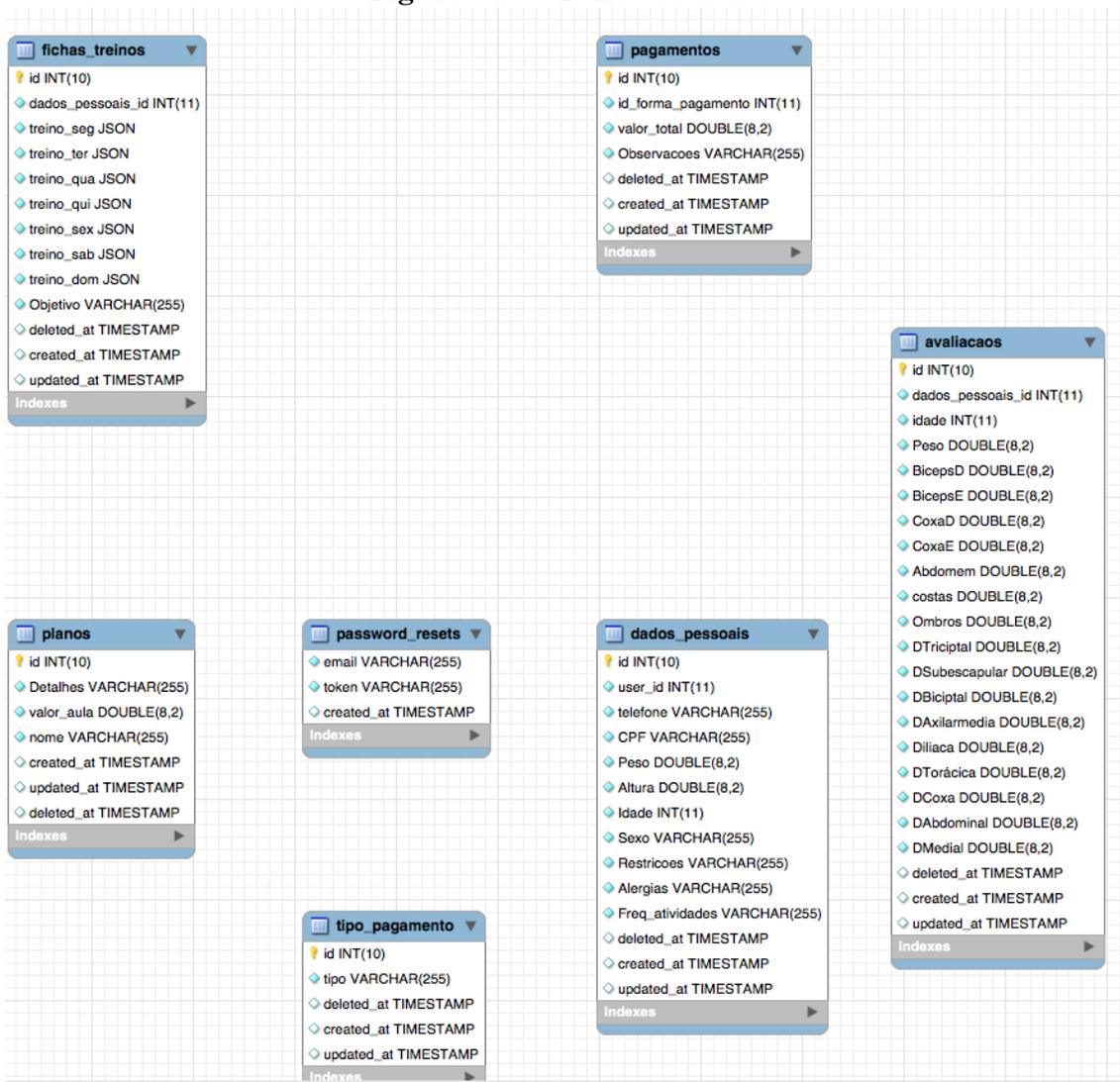
As informações persistentes para cada objeto, foram consideradas para a adição dos campos. Resultou-se portanto, nos modelos criados no Laravel e nas tabelas configuradas do banco de dados. O segundo sprint foi finalizado dentro do prazo com todas as ações previstas concluídas, bem como todos os requisitos atingidos. A tabela 3.6.3 ilustra a relação das tarefas com sua importância em porcentagem e a estimativa em dias de sua conclusão, em seguida a figura 3.6.3 ilustra o modelo de tabelas do banco de dados criado e por fim a figura 3.6.3.1 mostra os arquivos criados na estrutura de pastas do laravel.

Tabela 3.6.3 - Tarefas Sprint 2

Tarefa	Importância	Estimativa
1 Identificar cada entidade	70	1 Dias
2 Criar código de criação de cada tabela	50	7 Dias
3 Adicionar relacionamentos	70	2 Dias

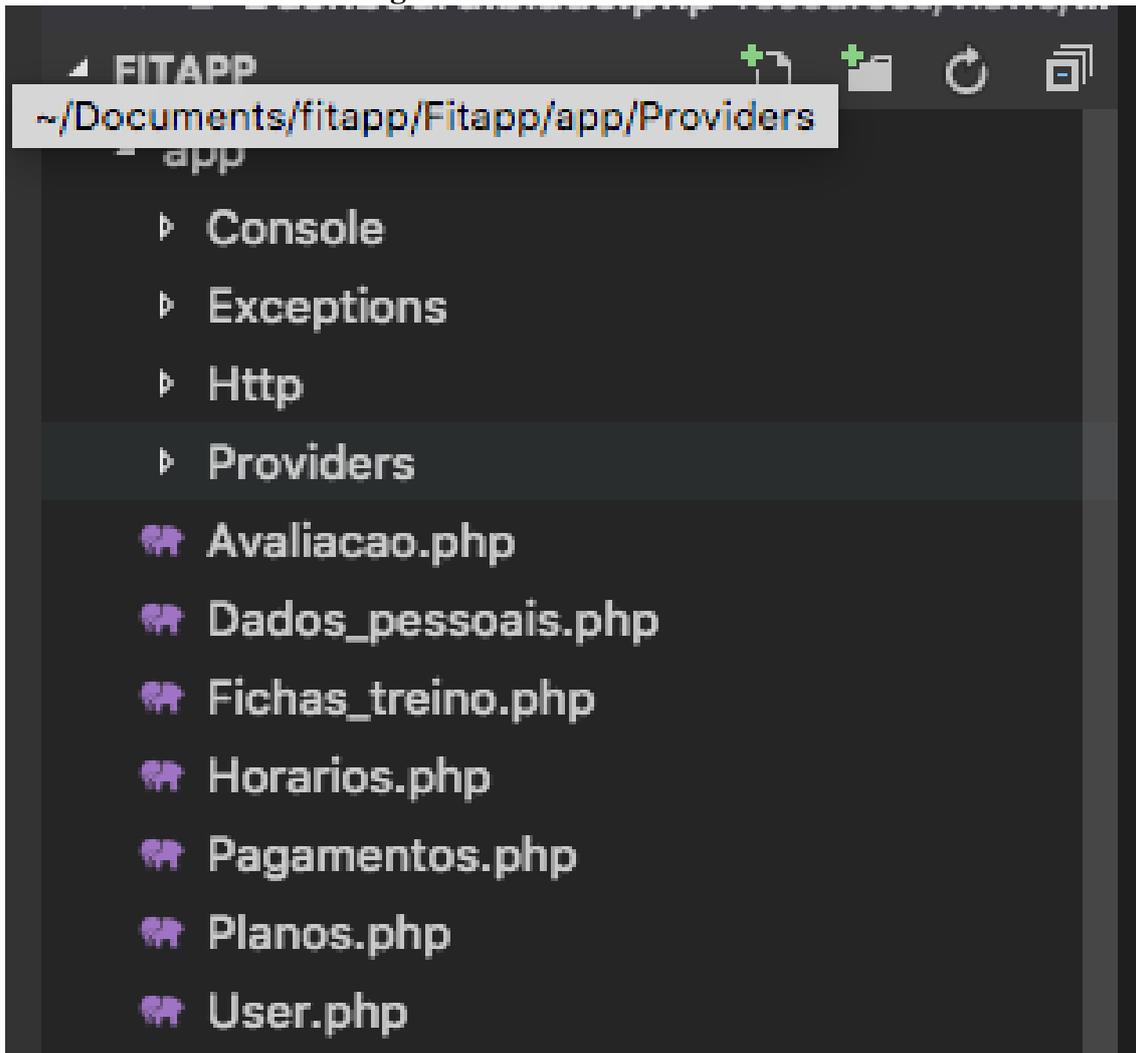
Fonte: O autor.

Figura 3.6.3 - Banco de dados



Fonte: O autor.

Figura 3.6.3.1 - Models Laravel



Fonte: O autor.

3.6.4 Sprint 3 - Criação dos controladores (Controllers)

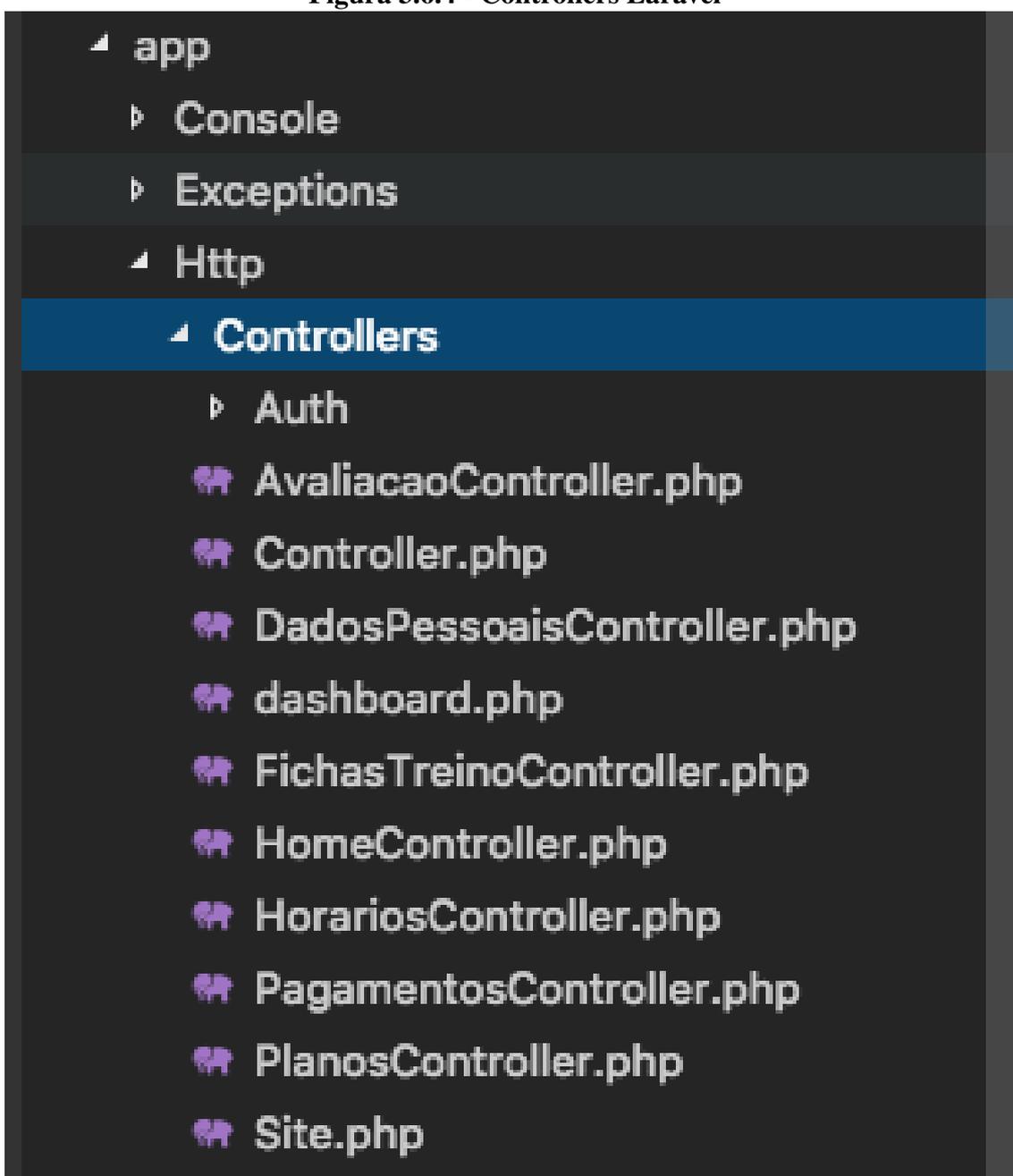
Para criar os controladores, o Laravel possui um comando acessível via CLI : `php artisan make:controller "Nome do controller"`. O padrão MVC define que os controllers vão conter a lógica de acesso as funcionalidades e vão manipular os modelos para conseguir as informações necessárias. A tabela 3.6.4 ilustra a relação das tarefas com sua importância em porcentagem e a estimativa em dias de sua conclusão, em seguida a figura 3.6.4 demonstra os arquivos criados para os controladores na estrutura de pastas do laravel.

Tabela 3.6.4 Tarefas Sprint 3

Tarefa	importância	Estimativa
1 Criar um controller por modulo	80	3 Dias
2 Adicionar regras de negocio	80	7 Dias

Fonte: O autor.

Figura 3.6.4 - Controllers Laravel



Fonte: O autor.

3.6.5 Sprint 4 - Criação das telas (views)

Para a criação das views, foi concebido um arquivo para cada tela com nome que segue o padrão do CRUD (Create, Read, Update, Delete) sendo cada um, relativo a sua função e organizado dentro de pastas conforme a figura abaixo. Ao final do prazo estipulado foi verificado que todas as telas estavam criadas e exibiam seus dados corretamente.

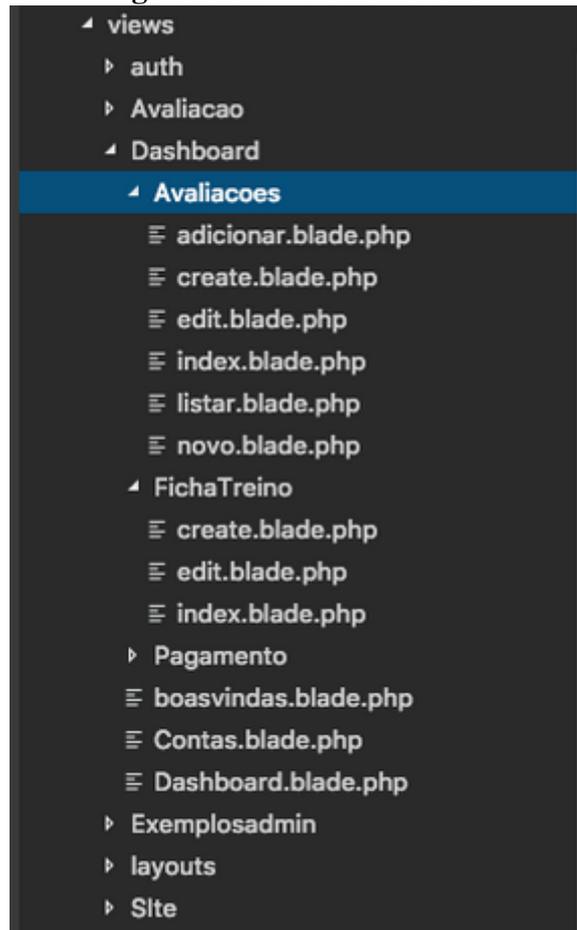
A tabela 3.6.5 ilustra a relação das tarefas com sua importância em porcentagem e a estimativa em dias de sua conclusão, em seguida a figura 3.6.5 mostra os arquivos criados para as views na estrutura de pastas do laravel.

Tabela 3.6.5 - Tarefas Sprint 4

	Tarefa	Importância	Estimativa
1	Cria um view para cada função por modulo	70	3 Dias
2	Padronizar estilos e elementos visuais na view	50	7 Dias

Fonte: O autor.

Figura 3.6.5 - Views Laravel



Fonte: O autor.

3.6.6 Sprint 5 - Criação das permissões e ajustes finais

O último sprint objetiva criar as permissões e autenticação assim como fazer os ajustes necessários nas telas e controllers a fim de adequar as suas funcionalidades.

O Laravel dispõe de um comando para ativar a autenticação : PHP Aritsan Make: auth; este, cria a autenticação para os usuários, para então, se alterar as rotas e telas tal que suportem a nova funcionalidade e definir o e-mail utilizado pelo administrador.

Após os ajustes, verificou-se que o prazo foi suficiente para realização da tarefas findo o último sprint do desenvolvimento.

4 RESULTADOS

Os resultados obtidos estão dispostos em duas seções separados pelo portal apresentado na seção 4.1. Este, foi desenvolvido para a divulgação de informações profissionais e em seguida o sistema FitApp, na seção 4.2. Nesta última, inclui-se a gestão da rotina profissional e alunos. No parâmetro dos resultados constatados, estão elencadas as características desejáveis do sistema frente aos seus requisitos e propósitos bem como testes adequados para evidenciar as circunstâncias que a característica/funcionalidade se comportou como deveria e atendeu as expectativas.

4.1 Site

A aplicação de técnicas modernas de desenvolvimento de software apoiada pelas ferramentas tecnológicas que implementam os modelos ágeis de desenvolvimento, se mostram efetivas e suficientes para criação de aplicações web. A aplicação desenvolvida atendeu e foi aprovada pelo cliente. Apenas sua implantação e real utilização ainda devem ser implementadas.

O site se encontra com fotos e textos de teste, para demonstrar o funcionamento pleno das funções e recursos visuais/ textuais. Os dados reais, como telefone, endereço planos e preços o cliente se acordou em entregar junto com os dados de acesso da hospedagem que será contratada por ele.

A análise do log em tempo real oferecido pelo servidor http embutido do Laravel, não aponta erros ou falhas durante a navegação pelo site. Destaca-se também que os níveis de acesso se consolidaram, onde somente o administrador pode manusear os dados de forma que os alunos acessam apenas sua visualização.

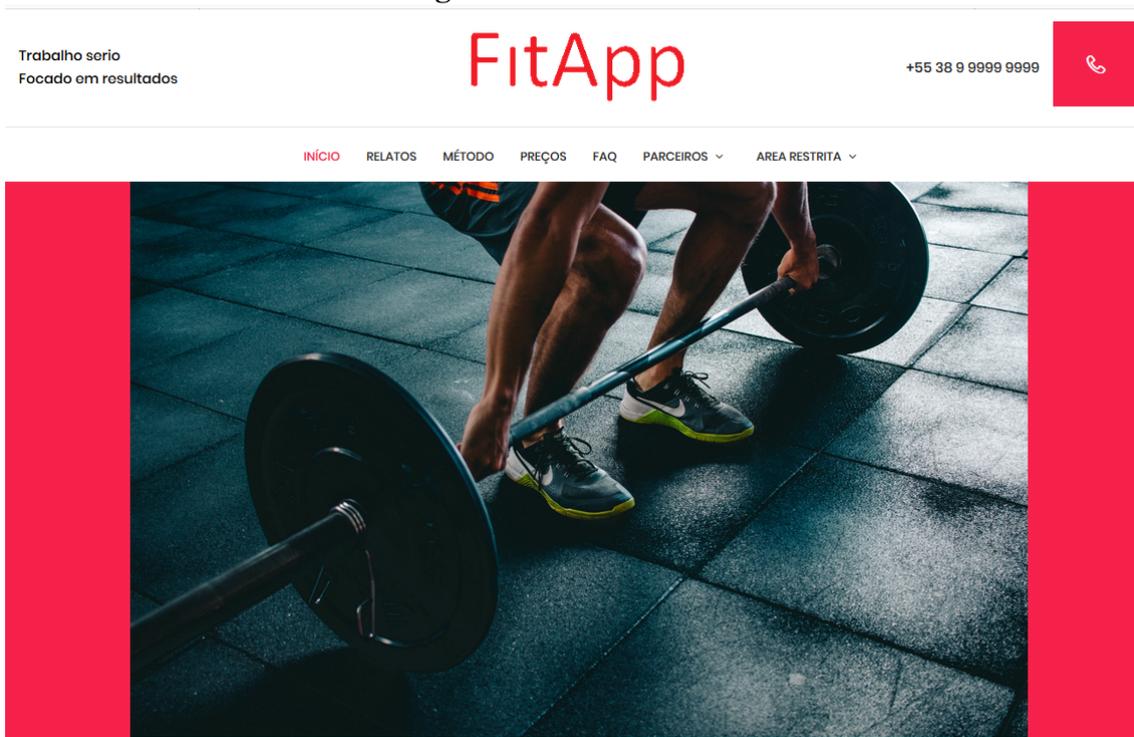
A adequação a diversos tamanhos de tela foi objeto de testes. Constatou-se que o sistema responde prontamente ao ajustar os textos e diminuir os menus e conteúdos estáticos tal que as principais informações do site foram priorizadas e atendeu a variadas resoluções dentro da faixa de 220X180 a 1920X1080. Pequenas quebras de espaçamento surgem em resoluções maiores em função da área de visualização. No entanto, não há interferência que inviabilize o uso da aplicação.

Outro aspecto a ser considerado, é o custo de utilização e desenvolvimento. Não houve gasto significativo para se desenvolver a aplicação. Não há um valor expressivo para a utilização do site em produção, uma vez que o custo de um domínio na internet tem valor fixado pelo RegistroBR, site com o preço de referência para registros ".com.br". Atualmente, o valor cobrado é de R\$40,00 por ano. Complementar ao domínio, o servidor onde o site será hospedado será compartilhado com o do sistema FitApp em uma única máquina, o que contribui consideravelmente para a redução do investimento nessa área.

Ao se realizar avaliações de conformidades com os requisitos iniciais, foi verificado que todos foram atingidos de maneira satisfatória. Os testes de desempenho em ambiente local, não apresentaram quaisquer falhas durante a navegação pelas páginas do sistema, bem como não apontaram lentidão ou travamentos.

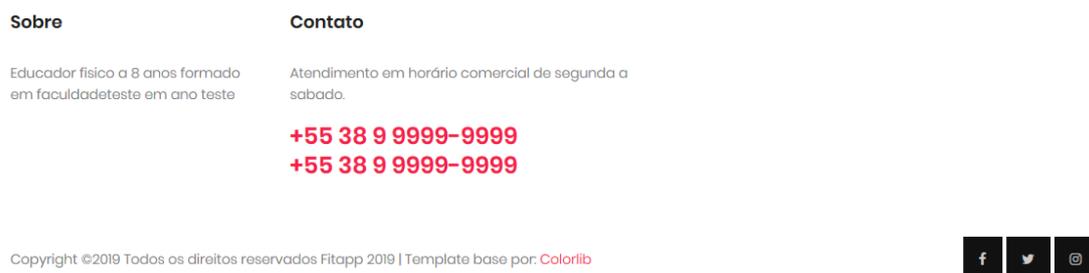
Para ilustrar seguem as telas, "página inicial", figura 4.1.1, da parte de "contato", figura 4.1.2 e "método", figura 4.1.3:

Figura 4.1.1 - Tela Home



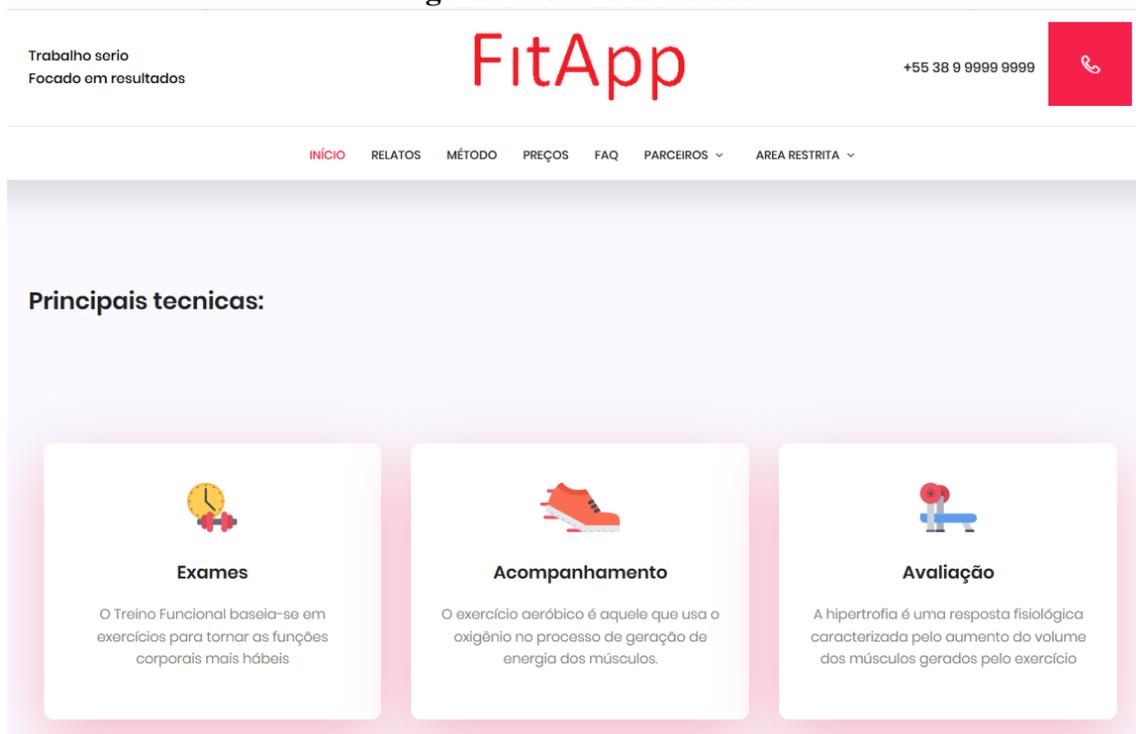
Fonte: O autor.

Figura 4.1.2 - Tela Contato



Fonte: O autor.

Figura 4.1.3 - Tela Método



Fonte: O autor.

4.2 Sistema FitApp

A primeira parte do sistema a ser avaliada foi o banco de dados. A rotina de testes foi apoiada pela ferramenta MySQL Workbench, que permite a manipulação de bancos de dados via interface, com recursos adequados para a realização destes. Primeiramente, foram testados individualmente cada tabela do banco inclusão, alteração e remoção. Nesta etapa todas as funcionalidades funcionaram como previsto, e em tempo adequado. Os comandos foram executados em frações de segundo. A segunda parte avaliada é relativa aos relacionamentos das tabelas. Foram feitos comandos de pesquisa do tipo "join": em que

se relacionam tabelas com o uso de chaves de identificação para o cruzamento de dados. Novamente, todos os comandos foram executados em frações de segundos e retornaram os resultados esperados sem erros ou travamentos.

Findo o teste do banco de dados, foi feita uma extensa navegação por todas as páginas do sistema com o log do servidor HTTP, embutido do Laravel, ativado. Essa etapa de avaliação, visa encontrar erros do protocolo http comuns, como por exemplo, página não encontrada 404. Realizou-se uma inclusão e uma remoção de cada um dos itens: Avaliação, Ficha de treino, Pagamento, Dados Pessoais e criação de um usuário. Ao final, inspecionou-se o log e apenas foram apresentados resultados normais das páginas e scripts disponíveis de maneira pública ao usuário. Todos os códigos encontrados no log de número 200, "sucesso", indicam que todas as requisições do protocolo foram adequadamente respondidas.

4.3 Testes

As tabelas a seguir documentam os testes utilizados para validar as funções do software, assim como tornam explícito o estado de pronto de cada um dos sprints. As mesmas também informam o método usado para a avaliação de cada funcionalidade, bem como seus estados encontrados e propõe a forma de se restaurar em caso de falha. Cada parâmetro que não for necessário será identificado por um N/A, adicionalmente, quando necessário a rotina de soft delete que, altera o processo de exclusão definitiva de um registro no banco trocando seu efetivo descarte por adicionar um campo deletado em contendo a data e hora da solicitação de exclusão, será mencionada. Todas estas considerações estão presentes nas tabelas de 4.3.1 até 4.3.5, presentes nesta seção.

Após a rotina de testes pode se afirmar que todas as funcionalidades implementadas encontram-se prontas e com requisitos atendidos. As avaliações se procederam sem interrupções de erros ou falhas intermitentes o que demonstra a solidez do método e das tecnologias.

Tabela 4.3.1 - Testes no portal(site)

CASO DE TESTE	
Código	1 - Teste navegação portal (site)
Finalidade	Verificar as funcionalidades atreladas as paginas do site.
Entradas	Dados de teste
Resultados Esperados	Páginas, abrindo rapidamente e sem erros
Dependências	N/A
PROCEDIMENTOS	
Preparação	Instalar o Laravel no ambiente de testes
Inicialização	Iniciar o servidor http
Execução	Abrir todas as telas acessáveis pelo menu
Reorganização	N/A
Interrupção	Restaurar a ultima versão do código .
Recursos Específicos	N/A
AVALIAÇÃO DO CASO DE TESTE	
Resultados Obtidos	Todos os resultados dentro do esperado.
Observações	N/A

Fonte: O autor.

Tabela 4.3.2 - Teste do módulo Pagamentos

CASO DE TESTE	
Código	2 - Teste do módulo Pagamentos
Finalidade	Verificar as funcionalidades atreladas a Pagamentos
Entradas	Dados de teste
Resultados Esperados	Adicionar, Excluir e visualizar pagamentos
Dependências	N/A
PROCEDIMENTOS	
Preparação	Instalar o Laravel no ambiente de testes
Inicialização	Iniciar o servidor http
Execução	Criar um pagamento, visualizar como aluno e admin, e excluir ao final
Reorganização	N/A
Interrupção	Restaurar a ultima versão do código .
Recursos Específicos	N/A
AVALIAÇÃO DO CASO DE TESTE	
Resultados Obtidos	Todos os resultados dentro do esperado.
Observações	N/A

Fonte: O autor.

Tabela 4.3.3 - Teste do módulo Ficha de Treino

CASO DE TESTE	
Código	3 - Teste do módulo Ficha de Treino
Finalidade	Verificar as funcionalidades atreladas a Fichas de treino
Entradas	Dados de teste
Resultados Esperados	Adicionar, Excluir e visualizar Fichas de Treino.
Dependências	N/A
PROCEDIMENTOS	
Preparação	Instalar o Laravel no ambiente de testes
Inicialização	Iniciar o servidor http
Execução	Criar uma Ficha, visualizar como aluno e admin, e excluir ao final
Reorganização	N/A
Interrupção	Restaurar a ultima versão do código .
Recursos Específicos	N/A
AVALIAÇÃO DO CASO DE TESTE	
Resultados Obtidos	Todos os resultados dentro do esperado.
Observações	N/A

Fonte: O autor.

Tabela 4.3.4 - Teste do módulo Avaliação

CASO DE TESTE	
Código	4 - Teste do módulo Avaliação
Finalidade	Verificar as funcionalidades atreladas a avaliações físicas
Entradas	Dados de teste
Resultados Esperados	Adicionar, Excluir e visualizar avaliações.
Dependências	N/A
PROCEDIMENTOS	
Preparação	Instalar o Laravel no ambiente de testes
Inicialização	Iniciar o servidor http
Execução	Criar uma avaliação, visualizar como aluno e admin, e excluir ao final
Reorganização	N/A
Interrupção	Restaurar a ultima versão do código .
Recursos Específicos	N/A
AVALIAÇÃO DO CASO DE TESTE	
Resultados Obtidos	Todos os resultados dentro do esperado.
Observações	N/A

Fonte: O autor.

Tabela 4.3.5 - Teste do Banco de dados

CASO DE TESTE	
Código	5 - Teste do Banco de dados
Finalidade	Verificar a estrutura do banco suporte as pesquisas necessárias
Entradas	Dados de teste
Resultados Esperados	Relacionamento entre tabelas feitos de maneira adequada. Inserção e updates sem erros ou inconsistências
Dependências	N/A
PROCEDIMENTOS	
Preparação	Migrar o banco para o servidor
Inicialização	Iniciar a conexão com o banco
Execução	Inserir dados;Pesquisar,alterar dados, Join entre tabelas
Reorganização	N/A
Interrupção	Restaurar a ultima versão do banco
Recursos Específicos	Conexão suportada com Laravel e php
AVALIAÇÃO DO CASO DE TESTE	
Resultados Obtidos	Todos os resultados dentro do esperado.
Observações	N/A

Fonte: O autor.

5 CONSIDERAÇÕES FINAIS

O presente texto relata todo o caminho percorrido dentro da engenharia de software e suas áreas correlatas para desenvolver um software. A aplicação que é resultado deste trabalho, FitApp, permitirá a seus usuários uma grande facilidade na gestão de seus alunos e características físicas importantes. A organização proposta pelo scrum como estrutura ágil de desenvolvimento foi produzida juntamente com a aplicação.

Todas as ferramentas e tecnologias propostas para ajudar o desenvolvimento mostram seu valor e utilidade ao longo do trabalho, principalmente o framework Laravel ,que foi o responsável por coordenar as principais partes e módulos e sua interação.

A evolução da engenharia web e de técnicas de desenvolvimento, bem como das tecnologias associadas, para que sistemas de informação tornem-se cada vez mais integrados permitindo expandir suas funcionalidades, sem perder o foco inicial e suas características. Com o uso constante da aplicação será possível conseguir cada vez mais efetividade para controlar e gerir os históricos dos documentos gerados. Além disso vai permitir ao administrador conseguir uma gestão eficiente de suas atividades profissionais. Outro aspecto é o desenvolvimento com a separação proposta pelo MVC, o que permite a fácil adição de novos módulos e expansões futuras. Os alunos também ao utilizarem o software, vão conseguir portar seus treinos e entender melhor os resultados alcançados, fazendo um paralelo entre os dados apresentados, potencializando assim seus resultados, e ajudando no cumprimento de metas.

5.1 Trabalhos Futuros

Mesmo com o software tendo mostrado resultados a contento, dentro das funcionalidades e requisitos propostos, há espaço para adição de melhorias e ferramentas, uma vez que as expectativas anteriores a concepção do software, evoluem com o seu uso.

Dentre as possibilidades para incrementos futuros, destaca-se:

- Aplicativo mobile integrado que consuma a API do site e mantenha os dados sincronizados e disponíveis off-line.
- Integração com sistemas de pagamentos online e bancários.
- Entrevistar os alunos e fazer uma análise da interação com o software.
- Quantificar o número de acessos e alcance do site junto as mídias sociais.

- Expandir o modelo para aceitar e separar novos avaliadores físicos e seus alunos, propondo um modelo de negocio mais genérico.

5.2 Principais Desafios

Ao se desenvolver uma aplicação que se propõe utilizar técnicas modernas de desenvolvimento ágil, com equipe pequena e uma grande variedade de ferramentas, se destacam vários pontos desafiadores:

- Gerir de maneira efetiva o tempo para cada tarefa ser executada.
- Produzir todos os artefatos necessários.
- Produzir um código reutilizável, para permitir a evolução do sistema.
- Definir as tecnologias adequadas.

REFERÊNCIAS

- Atlassian and Bitbucket. **Criado para equipes profissionais**. 2018. Disponível em: <<https://br.atlassian.com/software/bitbucket>>, Acesso em: 29 Dez. 2018.
- AUDY, J. **Scrum 360: Um guia completo e prático de agilidade**. 2013. 153 p.
- BECK, K. et. al. **Manifesto para Desenvolvimento Ágil de Software**. 2019. Disponível em: <<http://agilemanifesto.org/iso/ptbr/manifesto.html>>, Acesso em: 05 jan. 2019.
- BERTOCCO, M. et al. A client-server architecture for distributed measurement systems. **IEEE Transactions on Instrumentation and Measurement**, v. 47, n. 5, p. 1143–1148, 1998. ISSN 00189456.
- CAMPBELL, R. H.; ISLAM, N. A Technique for Documenting the Framework of an Object-Oriented System. **Computing Systems**, v. 6, n. 4, p. 363–389, 1993. ISSN 17448360.
- Composer. **Introduction**. 2018. Disponível em: <<https://getcomposer.org/doc/00-intro.md>>, Acesso em: 29 Dez. 2018.
- CROCKFORD. **How JavaScript Works**. 2019. Disponível em: <<https://www.crockford.com/>>, Acesso em: 01 jan. 2019.
- CUNHA, M. B.; JESUS, D. L. Produtos e serviços da web 2 . 0 no setor de referência das bibliotecas Products and services of web 2 . 0 in the reference sector of libraries. p. 110–133, 2012.
- DUARTE, L. **Os artefatos do Scrum**. 2019. Disponível em: <<https://escritoriodeprojetos.com.br/artefatos-do-scrum>>, Acesso em: 5 jan. 2019.
- FAYAD, M.; SCHMIDT, D. C. Object-Oriented Application Frameworks. **Communications of the ACM**, v. 40, n. 10, p. 32–38, 1997. Disponível em: <<http://www.cs.wustl.edu/~schmidt/CACM-frameworks.ht>>.
- FLANAGAN, D. **JavaScript O guia definitivo**. 6. ed. [S.l.: s.n.], 2013. 1062 p. ISBN 9788565837194.
- GIT. **Documentation**. 2018. Disponível em: <<https://github.com/features>>, Acesso em: 29 Dez. 2018.
- GRILLO. et. al. **Aprendendo Javascript**. 2019. Disponível em: <<http://www.amazon.com/dp/B0026OR2PY>>, Acesso em: 05 jan. 2019.

- HANSEN, S. A.; FOSSUM, T. V. Refactoring model-view-controller. **Journal of Computing Sciences in Colleges**, v. 21, n. 1, p. 120–129, 2005.
- IEEE. **IEEE standard Glossary of Software Engineering Terminology**. [S.l.], 1990. 84 p. Disponível em: <www.mit.jyu.fi/ope/kurssit/.../IEEE{_}SoftwareEngGlossary>
- Microsoft. **Getting Started**. 2018. Disponível em: <<https://code.visualstudio.com/docs>>, Acesso em: 29 Dez. 2018.
- Oracle. **About Us**. 2018. Disponível em: <<https://www.gitkraken.com/about>>, Acesso em: 29 Dez. 2018.
- Oracle. **MySQL Workbench**. 2018. Disponível em: <<https://dev.mysql.com/doc/workbench/en/>>, Acesso em: 29 Dez. 2018.
- Oracle. **MYSQL 5.5 REFERENCE MANUAL**. 2019. Disponível em: <<https://dev.mysql.com/doc/refman/5.5/en>>, Acesso em: 02 jan. 2019.
- Otwell, Taylor B. **Laravel A php Framework For Web Artisans**. 2019. Disponível em: <<https://laravel.com/docs/5.7>>, Acesso em: 01 jan. 2019.
- Paula Filho, W. d. P. Engenharia de. **XI Simpósio Brasileiro de Automação Inteligente (SBAD)**, p. 1–39, 2013.
- PRESSMAN, R. S. **Egenharia de Software: Uma Abordagem Profissional**. 7ª edição. ed. [S.l.]: AMGH Editora Ltda., 2011. 780 p. ISBN 9788580550443.
- SANCHEZ, F.; ALTHMANN, M. **Desenvolvimento web com: ASP.NET MVC**. [S.l.: s.n.], 2017. 399–404 p.
- SIKOS, L. F. **Web Standards:Mastering HTML,CSS e XML**. [S.l.]: Apress, 2011. ISBN 978-1-4302-4041-9.
- SOMMERVILLE, I. **Software Engineering**. [s.n.], 2011. 241–249 p. ISSN 0014-2972. ISBN 9780137035151. Disponível em: <<http://mnras.oxfordjournals.org/content/291/1/241.short>>.
- TORRES-SALINAS, D.; DELGADO-LÓPEZ-CÓZAR, E. Estrategia para mejorar la difusión de los resultados de investigación con la Web 2.0. **El Profesional de la Informacion**, v. 18, n. 5, p. 534–539, 2009. ISSN 1386-6710. Disponível em: <<https://recyt.fecyt.es/index.php/EPI/article/view/epi.2009.sep.07>>.
- Trello. **Simple à primeira vista, mas com muitas surpresas**. 2018. Disponível em: <<https://trello.com/tour>>, Acesso em: 29 Dez. 2018.

W3C. **World Wide Web Consortium**. 2018. Disponível em: <<https://www.w3.org/>>, Acesso em: 28 Dez. 2018.

W3C. **BOOTSTRAP 4 TUTORIAL**. 2019. Disponível em: <<https://getbootstrap.com/docs/4.2/about/overview>>, Acesso em: 05 jan. 2019.>.,.

W3C. **CSS TUTORIAL**. 2019. Disponível em: <<https://www.w3schools.com/css/>>, Acesso em: 08 jan. 2019.

W3C. **HTML 5 TUTORIAL**. 2019. Disponível em: <<https://www.w3.org/TR/html52/>>, Acesso em: 06 jan. 2019.

AUTORIZAÇÃO

Autorizo a reprodução e/ou divulgação total ou parcial do presente trabalho, por qualquer meio convencional ou eletrônico, desde que citada a fonte.

DIAMANTINA, MG, ____ / ____ / _____.

LUCAS DE SOUZA FERREIRA

lucasdtna@yahoo.com.br

Universidade Federal dos Vales do Jequitinhonha e Mucuri

Campus J.K. - Alto da Jacuba, S/N.