

# Desenvolvimento de sistema para relacionamento entre academias e seus clientes

**Matheus de Lima Gutierrez, Felipe Martin Sampaio**

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)  
Campus Farroupilha  
95.174-274 – Farroupilha – RS – Brasil

maatzg@gmail.com, felipe.sampaio@farroupilha.ifrs.edu.br

***Abstract.** This work presents the development of a platform of transparency and relationship between client and gym. It provides the management of gym cards; the monitoring of the evolution of the athletes; and the dissemination of institutional informations, such as the curriculum of professionals and datas of the equipments of the gym. The functionalities of the proposed work were compared with similar applications in the market, presenting different characteristics. The Laravel framework was used as the basis for the application development, providing high productivity and organization of the different parts of the system. With the responsive Web application proposed in this work, it is possible for the athlete to consult his training and his evolution remotely from different devices.*

***Resumo.** Este trabalho apresenta o desenvolvimento de uma plataforma de transparência e relacionamento entre cliente e academia. Ela proporciona o gerenciamento de fichas de treino; o acompanhamento da evolução do aluno; e a divulgação de informações da instituição, como formação de profissionais e dados de aparelhos e equipamentos que possuem na academia. As funcionalidades do trabalho proposto foram comparadas com aplicações similares no mercado, apresentando características diferentes. O framework Laravel foi utilizado como base para o desenvolvimento da aplicação, proporcionando alta produtividade e organização das diferentes partes os sistema. Com a aplicação Web responsiva proposta neste trabalho, torna-se possível ao aluno consultar seu treino e sua evolução de forma remota a partir de diferentes dispositivos.*

## **1. Introdução**

Com uma nova ordem da comunicação organizacional, surge a necessidade das empresas se comunicarem de forma permanente, com todos os seus públicos estratégicos. Dessa forma, as organizações modernas passam a ter um perfil institucional com comportamento mais adequado, aumentando sua responsabilidade frente às transformações mundiais (RUÃO, 2016). Às empresas atribui-se um compromisso público cada vez maior, e passam a agir de acordo com novas demandas,

competindo e gerando qualidade. A comunicação institucional é diretamente responsável pela construção e formação da identidade corporativa de uma organização, feita no sentido de divulgar e valorizar a organização. Uma característica marcante dos modelos organizacionais é a preocupação com o bem-estar social, pois dessa forma atraindo-se mais clientes e gera-se mais lucro. Como as organizações estão cada vez mais cientes do seu papel social, uma boa comunicação institucional passa a ser pensada como indispensável na obtenção de bons resultados. Desta forma, os gestores organizacionais passam a não se preocupar apenas com a qualidade total de seus serviços e produtos, mas também começam a sentir-se obrigados a buscar excelência na comunicação com seus públicos estratégicos, como funcionários e clientes. (ALBÉ, 2007).

As academias são um exemplo de organizações que estão buscando, cada vez mais, o aprimoramento da relação com seus clientes. A saúde é algo conquistado através de cuidados com o corpo, como com a prática de exercícios físicos e uma boa alimentação. A prática de atividades físicas vem se tornando cada vez mais comum entre os indivíduos, por conta da preocupação principalmente com a saúde, e também com questões de estética (BRITO, 2017).

Mesmo assim, sistemas para acompanhamento de atividades físicas ainda são raros em academias. Em 2013, uma pesquisa foi realizada na cidade de Curitiba, tendo como base de amostra 15 academias da região, cujos resultados estão apresentados na Figura 1. Como principais aspectos, têm-se: 73% das academias ainda não têm um *software* para controlar e acompanhar os exercícios praticados por seus alunos (Figura 1a); 95% dos alunos entrevistados têm interesse em um sistema para acompanhamento (Figura 1b); e 93% dos alunos não costumam fazer nenhum tipo de acompanhamento (Figura 1c) (BENTO, 2013).

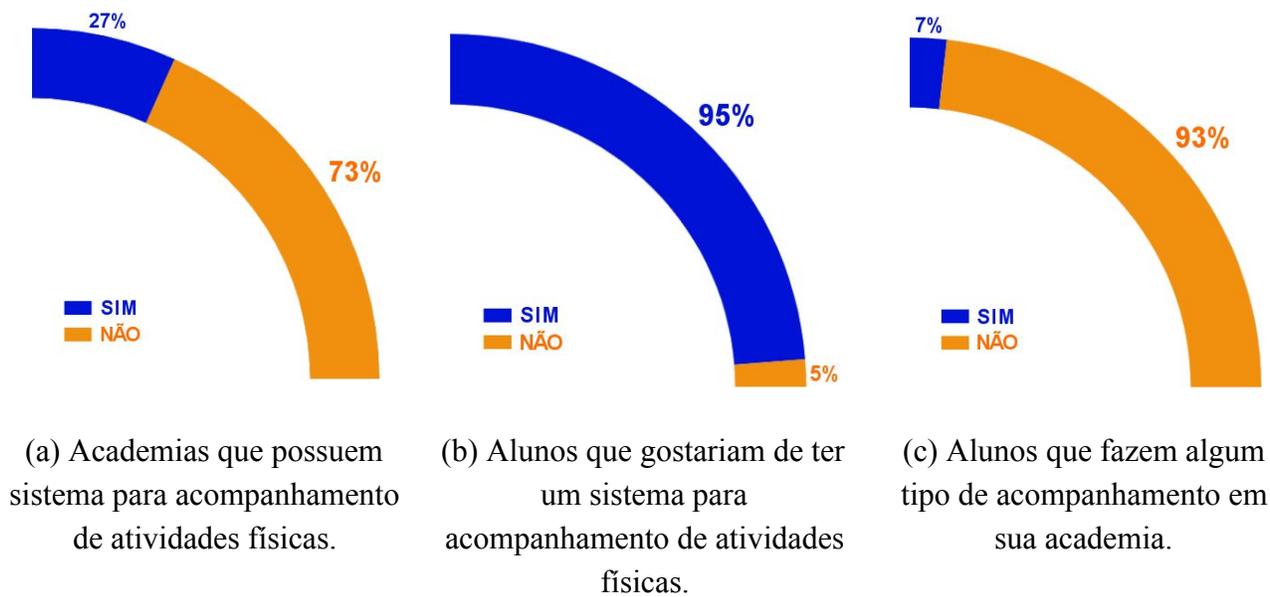


Figura 1: Pesquisa sobre a existência de sistemas informatizados para acompanhamento de atividades físicas.

Fonte: BENTO, 2013

Ilustração: Autor

Sendo assim, um investimento em tecnologias para este tipo de controle é um diferencial para a academia agregar qualidade ao serviço oferecido. Dessa forma, também, matricular mais alunos e aumentar o seu retorno financeiro, tornando um sistema de acompanhamento de atividades físicas algo comum entre as academias, visto que conforme os resultados da pesquisa apresentada, alunos de academias demandam por este tipo de sistema. Além disso, é inevitável a integração de uma melhora na comunicação institucional dessas empresas, visto a importância deste relacionamento entre instituição, funcionários e clientes.

Observada a necessidade de um sistema para o relacionamento entre academias e seus clientes, o objetivo deste trabalho é desenvolver um sistema Web institucional padrão para academias, a fim de auxiliar no acompanhamento das atividades físicas de seus alunos. Os objetivos específicos a serem contemplados no desenvolvimento são: auxiliar na comunicação institucional de academias, para um melhor relacionamento com o seu público estratégico, através de um sistema Web personalizável; proporcionar às academias um sistema de gerenciamento de informações relativas ao funcionamento das mesmas, pautado em diferentes níveis de privilégios de uso, para a manutenção dos dados atualizados; disponibilizar um mecanismo em que o aluno e o instrutor possam gerenciar a ficha de suas atividades físicas; oferecer o acompanhamento da evolução de alunos, mostrando a variação de

medidas corporais importantes dentro de um espaço de tempo; apresentar um guia, em que o cliente consiga observar todos os equipamentos e aparelhos da academia, bem como os exercícios que podem ser praticados em cada um deles, informando quais músculos são afetados; transparecer currículo dos profissionais que trabalham na academia, e detalhes dos aparelhos utilizados, como por exemplo a data da última manutenção feita.

Como forma de atingir os objetivos propostos, este trabalho oferece uma plataforma para o aluno acessar sua conta no sistema, consultar e alterar dados da sua ficha de treino, registrar e monitorar sua evolução, estudar exercícios através do guia, além de conferir a chegada de um equipamento ou um instrutor novo na academia. Sem focar especificamente nas necessidades de uma única empresa, pretende-se alcançar pequenas e médias empresas com um custo reduzido — já que o mesmo sistema contemplaria várias empresas, e cada uma delas teria a aplicação Web com sua *interface* personalizada —, e assim fazer com que elas estejam disponíveis na internet para seus clientes e demais público, tornando a instituição transparente através da exposição de suas informações.

## **2. Referencial teórico**

Nesta seção é abordada a fundamentação teórica, e são apresentados os seguintes conceitos: os principais benefícios da tecnologia na saúde e estética corporal; e a escolha do Laravel como *framework* para o desenvolvimento Web.

### **2.1. Benefícios da tecnologia na saúde e estética corporal**

A atividade física moderada e regular é a principal forma de melhorar a saúde e qualidade de vida (COBRA, 2003). O crescente aumento do sedentarismo, observado na espécie humana, parecem exercer um papel fundamental para a perda de desempenho orgânico. A ausência de atividade física traz um número elevado de efeitos prejudiciais ao ser humano (PRAZERES, 2007).

A evolução tecnológica oferece conforto para o homem moderno. A tecnologia otimiza tarefas do dia-a-dia, mas em contrapartida diminui as exigências de movimentação do corpo no cotidiano, o que traz consigo o aumento do sedentarismo, um dos principais inimigos de uma boa qualidade de vida (COBRA, 2003). Porém, para tentar mudar este quadro, a tecnologia pode funcionar como algo a auxiliar a atividade física, e assim incentivar pessoas a ingressar a este meio. A tecnologia pode auxiliar na gestão do acompanhamento preventivo da saúde dos alunos (PINOCHET, 2011).

No ambiente estético, a musculação modela o formato do corpo, enfatizando um trabalho de aumento do volume muscular e do alcance de simetria corporal. Essa atividade deve ser realizada dentro dos limites do indivíduo, considerando todos os princípios do treinamento físico, com uma maior ênfase sobre o princípio da individualidade biológica. Dentre os principais benefícios oferecidos

pela musculação estética, e que ajudam a obter uma melhor qualidade de vida dos praticantes, temos o aumento da massa corporal metabolicamente ativa e a melhora na auto-imagem (PRAZERES, 2007).

### 2.3. Laravel

O *framework* Laravel foi escolhido para realização do trabalho por ser empregado pela comunidade de desenvolvimento de aplicações Web e por, assim como alguns outros *frameworks*, trabalhar com uma estrutura padronizada e uma boa organização de diretórios. A Figura 2 apresenta um diagrama ilustrando a estrutura proposta pelo *framework*. O Laravel é um *framework* PHP utilizado para desenvolvimento Web, que utiliza a arquitetura *MVC* — arquitetura que divide uma aplicação em três itens: a representação dos dados na linguagem de programação (*Model*), a apresentação desses dados (*View*) e a ligação e tratamento das informações antes da apresentação (*Controller*) (BARROS, 2007). A arquitetura *MVC* tem como principal característica auxiliar a desenvolver aplicações seguras e performáticas de forma rápida, com código limpo e simples. Este *framework*, ainda, utiliza uma *Engine* de *template* chamada *Blade* — um pré-processador de *template* que combina todos os elementos de visualização para formar uma determinada página Web, com o principal objetivo de separar a lógica das *views* (NGUYEN, 2015) —, que traz uma série de ferramentas que ajudam a criar *interfaces* gráficas organizadas e funcionais de forma rápida, e que acaba evitando a duplicação de código (SILVA, 2015).

Para comunicar-se com um Banco de Dados, o Laravel utiliza uma implementação chamada de *Eloquent* — uma ORM (*Object-Relational Mapping*) capaz de interagir com o banco de dados de forma simplificada (NGUYEN, 2015) —, que trata-se de uma ferramenta com funcionalidades para facilitar o cadastro, a modificação, a busca e a exclusão de registros. Com um código simples pode-se configurar uma conexão com banco de dados (SILVA, 2015).

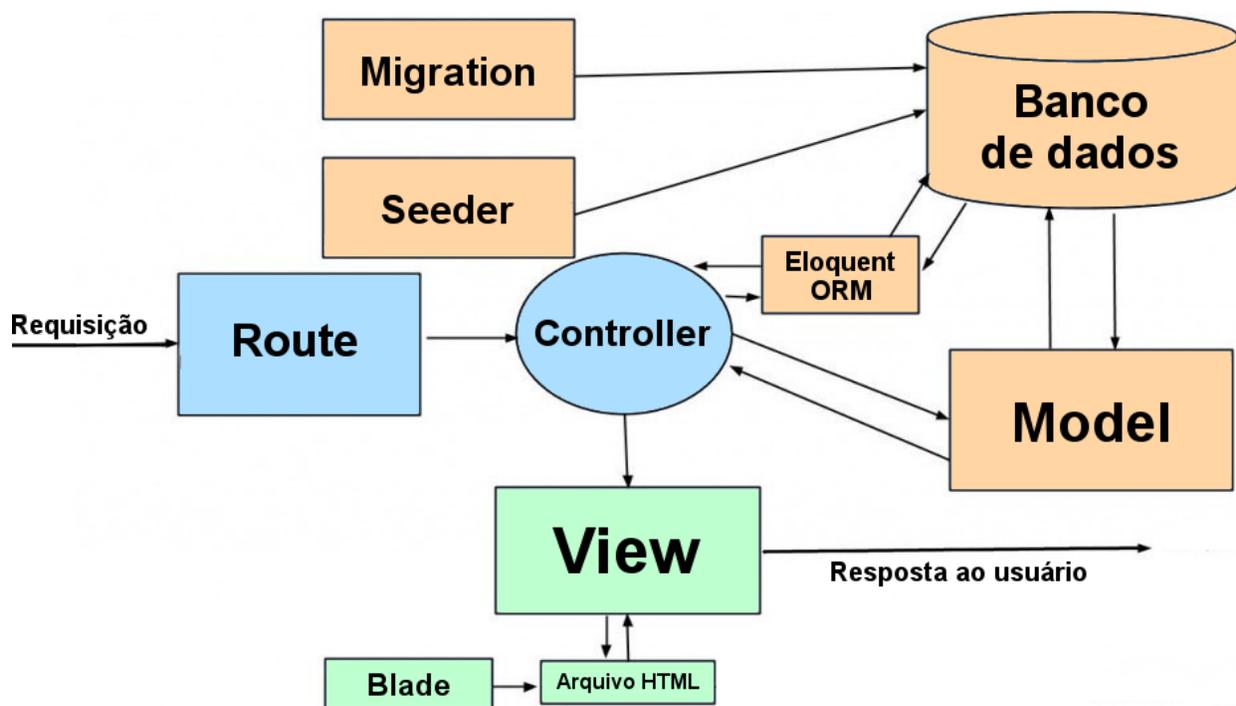


Figura 2: Estrutura do *framework* Laravel

Fonte: SILVA, 2015

Ilustração: Autor

### 3. Aplicações relacionadas

Foi realizada uma pesquisa sobre tecnologias semelhantes no mercado. Dentro de uma lista de sistemas, encontrou-se três *softwares* similares ao trabalho proposto. São eles: *Pro-Treino*<sup>1</sup>, *iFitness*<sup>2</sup> e *GO!Fit*<sup>3</sup>.

O *Pro-Treino* é um sistema multiplataforma (*desktop*, *web*, *app*) capaz de montar fichas automaticamente, além de oferecer a impressão ao chegar na academia, através de uma impressora semelhante às de nota fiscal, imprimindo fichas de forma compacta e visando diminuir o uso do papel. Essa ferramenta também permite o aluno fazer avaliação física e acompanhamento da evolução. Já o *iFitness* possui semelhanças, porém não deixa explícito em seu site oficial as plataformas em que pode ser operado. Por fim, assim como o primeiro sistema citado, o *GO!Fit* é multiplataforma, podendo ser utilizado tanto na versão *desktop*, quanto na versão *web* e *app*.

<sup>1</sup> Disponível em: [www.pro-treino.com](http://www.pro-treino.com)

<sup>2</sup> Disponível em: [www.ifitness.com.br](http://www.ifitness.com.br)

<sup>3</sup> Disponível em: [www.gofitweb.com](http://www.gofitweb.com)

A Tabela 1 apresenta uma comparação qualitativa entre os sistemas relacionados e a aplicação desenvolvida neste trabalho. Como critérios para comparação, foram definidos: (1) o suporte à criação de fichas de treino; (2) a disponibilização de um guia com a organização dos exercícios, equipamentos utilizados e músculos atingidos; (3) o monitoramento da evolução das medidas corporais dos alunos; (4) a transparência sobre os equipamentos e instrutores; (4) uma área para informações institucionais; (5) a disponibilização do sistema por uma aplicação Web; e, por fim, (6) a padronização do sistema de forma a contemplar diferentes perfis de academias.

Tabela 1: Comparação do sistema proposto neste trabalho com aplicações relacionadas.

Funcionalidade / Aplicação	Pro-Treino	iFitness	GO!Fit	Trabalho proposto
Fichas de treino	V	V	V	V
Guia	X	X	X	V
Evolução	V	V	V	V
Transparência sobre equipamentos e instrutores	X	X	X	V
Informações institucionais	X	X	X	V
Versão Web	V	X	V	V
Padronização do sistema	X	X	X	V

Ao contrário do trabalho proposto, todos os sistemas citados têm uma preocupação maior com o controle financeiro, entre outras questões diversas, como o acesso da academia. Além disso, nenhum deles oferece: (1) um guia personalizável, (2) uma área para a empresa gerenciar informações institucionais que serão lidas por seus clientes, (3) questões como transparência da academia sobre equipamentos e instrutores, ou (4) um sistema padrão com *interface* customizável, como sistematizado na Tabela 1.

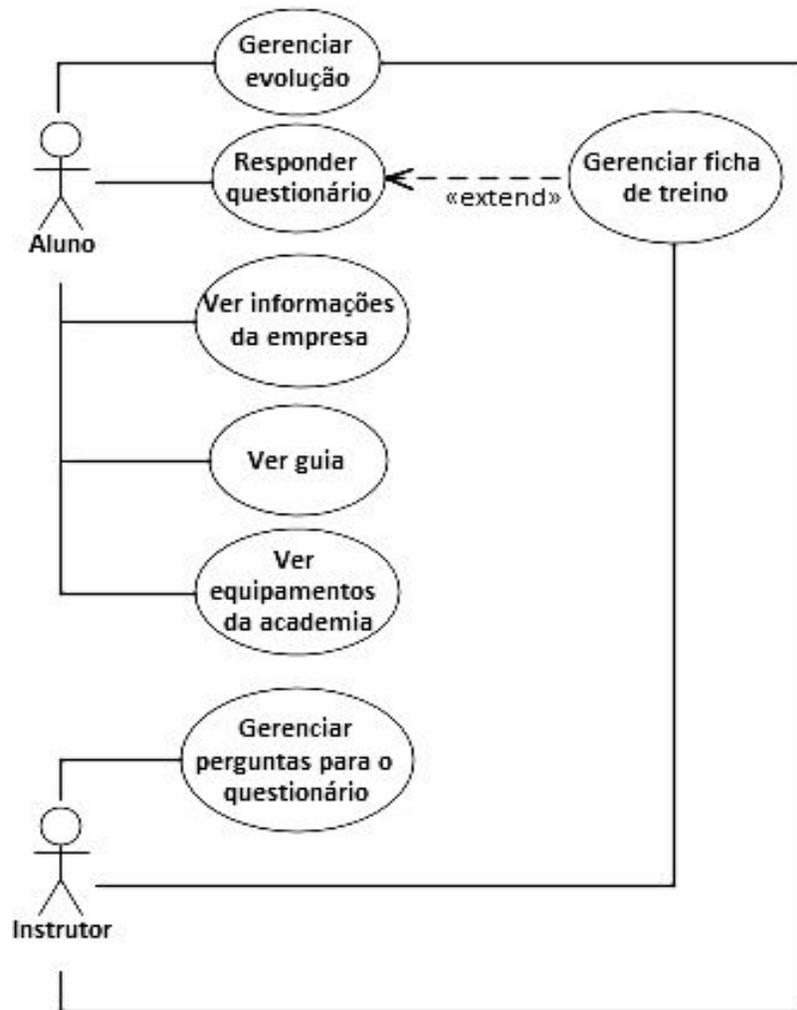
#### 4. Procedimentos metodológicos

Nesta seção são abordados os métodos utilizados na implementação do sistema, para: (1) descrever como foi realizada a configuração do ambiente de trabalho; (2) expor como foi projetado e desenvolvido o banco de dados, suas tabelas e colunas auxiliares, e a modularização do sistema; (3) relatar como foi feita a interação entre aplicação e banco de dados utilizando *migrations*, *seeders* e *factories* (disponíveis através do *framework* Laravel); (4) explicar a criação e o funcionamento de

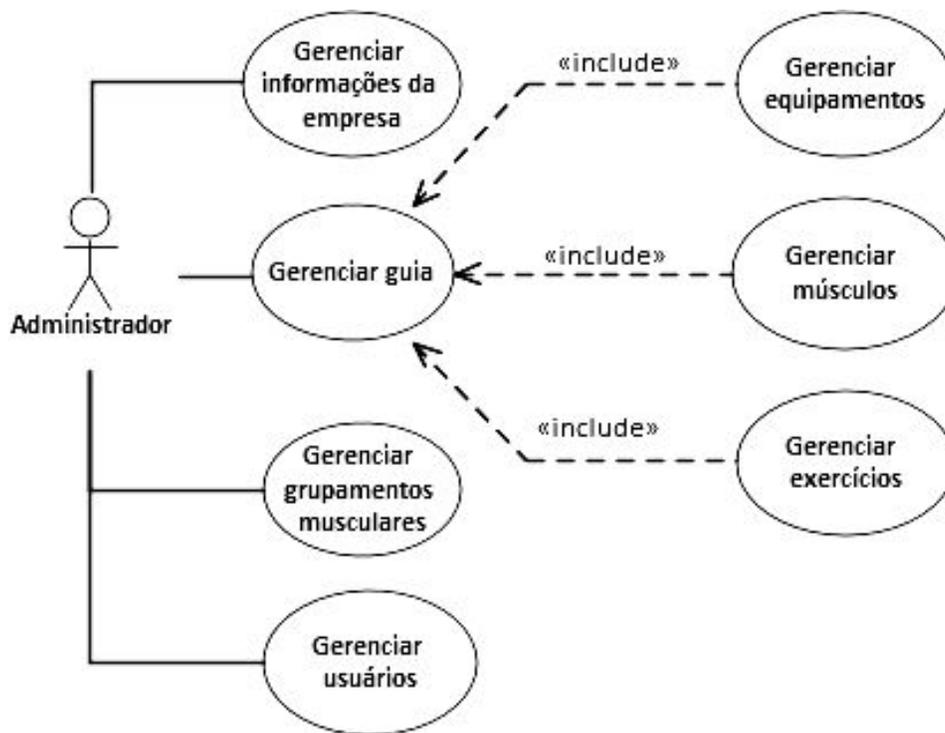
*routes*, *controllers* e *models*; e (5) mostrar detalhes do desenvolvimento da *interface* gráfica das páginas, bem como a construção de um dicionário de idiomas oferecido pelo *framework*. Além disso, pode-se observar no subtópico a seguir o diagrama de casos de uso, demonstrando o que cada ator do sistema pode fazer, mesmo que as permissões sejam alteráveis por cargos hierárquicos superiores.

#### **4.1. Diagrama de casos de uso**

O aluno da academia (Figura 3a) pode: gerenciar sua evolução, gerenciar sua ficha de treino (caso responda o questionário), visualizar as informações institucionais da academia, bem como o guia de exercícios e os equipamentos da academia. O instrutor (parte inferior da Figura 3a) pode: gerenciar as perguntas para o questionário, as fichas de treino de todos os alunos, e a evolução dos mesmos. Já o administrador do sistema (Figura 3b), tem a possibilidade de: gerenciar os itens do guia (equipamentos, exercícios, músculos e grupamento musculares) as informações institucionais da empresa e os usuários cadastrados no sistema.



(a) Casos de uso correspondentes ao aluno e ao instrutor



(b) Casos de uso correspondentes ao administrador do sistema

Figura 3: Diagrama de casos de uso do sistema desenvolvido.

## 4.2. Configuração do ambiente de trabalho

Inicialmente, criou-se um repositório utilizando o serviço *GitLab*<sup>4</sup>, o qual utiliza o sistema de versionamento de arquivos *Git*<sup>5</sup>. Feito isso, clonou-se o mesmo em um diretório do computador a ser usado para a desenvolvimento do trabalho. Com isso, o projeto está iniciado no *GitLab*, pronto para receber arquivos.

Foi montado um ambiente de desenvolvimento apropriado para a implementação de sistemas para web, com *Wamp Server 3.1* — um servidor local *Windows* para desenvolvimento de sistemas *web* (WAMP, 2019) —, *PHP 7.2* — uma linguagem de script open source adequada para o desenvolvimento *web* e que pode ser embutida dentro do *HTML* (PHP, 2019) —, *MySQL 5.7* — popular sistema de gerenciamento de banco de dados (MYSQL, 2018a). Com isso, é possível iniciar

<sup>4</sup> Disponível em: <https://gitlab.com/>

<sup>5</sup> Disponível em: <https://git-scm.com/>

qualquer sistema para internet através das linguagens e programas explicitados. Logo depois, foi gerado o projeto com Laravel 5.6.

Para isso, primeiramente foi necessário fazer a instalação do *Composer* — uma ferramenta para gerenciamento de dependências em PHP, que permite a declaração de bibliotecas das quais um projeto depende e as gerencia (COMPOSER, 2019) — no computador, para assim conseguir iniciar um projeto com o *framework*. Passando o comando de criação do projeto através do *prompt* de comando, foi criado o projeto, sendo possível, assim, começar a implementar o sistema.

A partir disso, foi necessária a configuração do arquivo *.env* na raiz do diretório gerado pelo comando anterior, como demonstrado na Figura 4. Nesse arquivo é necessário colocar o nome e a URL da aplicação, o banco de dados, o host, a porta, o nome do banco de dados, o usuário e sua senha, entre outros itens.

```
1 APP_NAME=Vex
2 APP_ENV=local
3 APP_KEY=base64:qxPYkYDXDr4mwQs2QB/mD5XKRr
4 APP_DEBUG=true
5 APP_LOG_LEVEL=debug
6 APP_URL=http://academia.imaatz.com
7
8 DB_CONNECTION=mysql
9 DB_HOST=127.0.0.1
10 DB_PORT=3306
11 DB_DATABASE=nome_do_banco
12 DB_USERNAME=usuario
13 DB_PASSWORD=senha
```

Figura 4: Trecho de código do arquivo *.env*

### 4.3. Módulos do sistema

A modelagem foi dividida em dois módulos: o núcleo do sistema; e dados referentes à academia. O *núcleo do sistema* é o módulo onde são guardados os seguintes dados: usuários, tipos de usuários e suas autorizações, informações referentes à empresa (linha do tempo da empresa, parceiros ou patrocinadores, serviços, personalização e configurações dos dados), informações de rodapé das páginas, textos da página inicial, e itens para contato com a empresa (telefones, e-mails e endereços). As autorizações, para cada tipo de usuário, foram elaboradas através de uma tabela no banco de dados com campos do tipo lógico, em que assinalam o que o usuário pode ver, cadastrar, modificar ou apagar

no sistema. Nas configurações do sistema, o administrador pode personalizá-lo, alterando cores e fontes, além de alterar o nome da empresa, logo, ícone da aba do navegador, suas redes sociais (*Facebook, Instagram, Twitter e LinkedIn*), seus números de telefone, e-mails, e endereço de todas as suas filiais.

O *módulo da academia* é onde estão presentes os itens referentes à academia, como guia, acompanhamento ao aluno e gerência de equipamentos e suas manutenções. Neste módulo foram envolvidos os dados de: equipamentos da academia; músculos e exercícios para o guia; questionário para geração de fichas de treino; fichas de treino; e medidas para acompanhamento da evolução do cliente. Com isso, supre-se as necessidades de transparência para o estabelecimento e de acompanhamento para o aluno.

#### **4.4. Modelagem de dados**

Foi realizada a modelagem do *software* através de um modelo lógico de banco de dados, pois este ajuda na manutenibilidade, na caracterização da construção de um *software* em camadas e no mapeamento de dados entre os modelos orientados a objetos (ESPÍNDOLA, 2017). Este modelo oferece um maior detalhamento sobre os requisitos do sistema, obtidos através de pesquisas por melhores formas de projetar cada parte do sistema.

Como pode ser observado no diagrama lógico de banco de dados (Apêndice A), o sistema possui 36 tabelas, sendo que as tabelas que estão representadas pela cor laranja são do módulo da academia, as representadas pela cor azul são do núcleo do sistema, e as demais (cinzas e amarela) são tabelas auxiliares.

A Figura 5 ilustra o banco de dados expondo as tabelas referentes ao módulo da academia. Observando o diagrama, conclui-se que: (1) um usuário pode ter várias medidas sobre um item o qual esteja monitorando; (2) um equipamento possui vários registros de manutenção; (3) um músculo pertence a um grupamento muscular; (4) um usuário gerencia uma lista de perguntas; (5) cada resposta é atribuída a um usuário e a uma pergunta; (6) um item da ficha de treino pertence a um usuário, e contém um exercício; (7) uma tabela com relação tripla registra uma ligação entre equipamento, exercício e músculo; (8) medidas, equipamentos e itens da ficha de treino possuem uma tabela para registro de histórico, ligada ao usuário que promoveu a alteração.

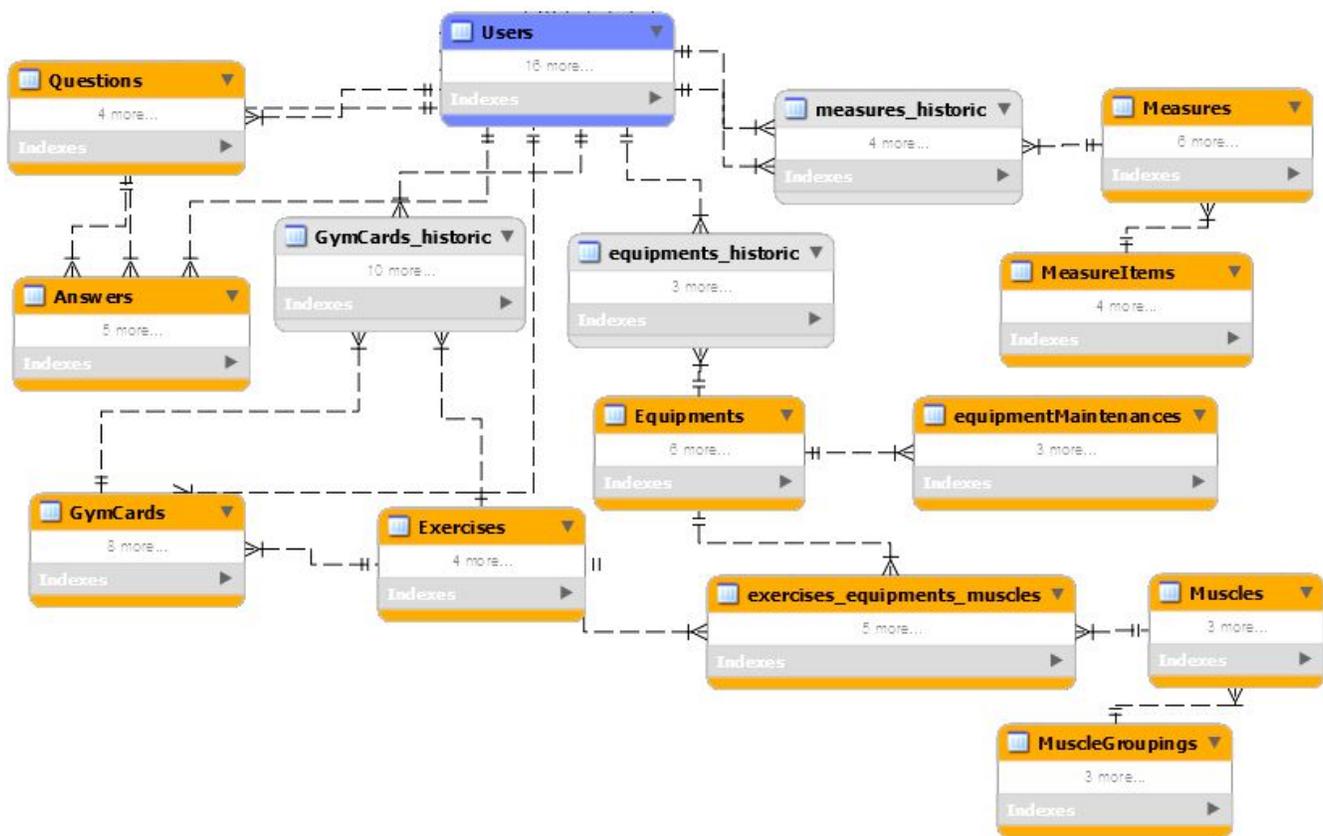


Figura 5: Diagrama lógico de banco de dados do módulo da academia

Foram criadas as tabelas auxiliares para o registro do histórico de acontecimentos em cada tabela importante para o sistema. Julgam-se importantes os registros de: configurações, usuários, informações da empresa, informações de rodapé, informações da página inicial, parcerias, serviços, equipamentos, fichas de treino e medidas de alunos. Para a implementação, planejou-se que o sistema, após todo o dado cadastrado ou modificado em uma tabela importante, insira automaticamente os novos dados na tabela auxiliar, contendo a data do acontecimento, e o registro da ação realizada (criação, alteração ou remoção).

Além disso, criou-se colunas auxiliares em algumas tabelas, a fim de controlar acesso, ou ditar o comportamento de um registro do banco de dados. Algumas tabelas possuem a coluna *trash* (lixeira) do tipo *booleano*, a fim de definir qual item está na lixeira, permitindo que usuários descartem e restaurarem registros. Outra coluna auxiliar implementada foi a *order* (ordem), a fim de indicar a posição do registro em determinada lista de informações (como, por exemplo, definir corretamente a ordem em que os exercícios devem ser executados dentro da ficha). E, por último, a coluna auxiliar *hierarchy* (hierarquia), responsável por saber quais registros são mais importantes que os demais —

isso ocorre, por exemplo, na tabela de tipos de usuários, a fim de saber qual tipo de usuário está acima ou abaixo dos demais no nível de hierarquia definido pela empresa.

#### 4.5. Utilização do *framework* Laravel

Os métodos utilizados para o desenvolvimento do sistema serão explicitados neste subtópico, a fim de oferecer um maior detalhamento da implementação da aplicação, tanto no *backend* quanto no *frontend*. Assim, é possível ter uma clara visão geral do funcionamento do *framework* Laravel.

##### 4.5.1. Criando interações com o banco de dados

Através do *PHPMYAdmin* — ferramenta *PHP* destinada a lidar com a administração do *MySQL* através da web (PHPMYADMIN, 2019) —, foi criado o banco de dados a ser utilizado para o projeto. O Laravel trata de criar as tabelas automaticamente. Para isso, tornou-se necessária a configuração do arquivo *.env*, criado pelo *framework*, colocando o nome do esquema, bem como todos os itens necessários para iniciar qualquer comunicação entre aplicação e banco de dados, como *SGBD* utilizado, porta, nome de usuário e senha. Feito isso, tornou-se possível a criação de *migrations*, *factories* e *seeders*.

Com o uso das *migrations*, foram definidas todas as tabelas do banco de dados, assim como foram modeladas inicialmente. Depois disso, foram definidas as *factories*, ou seja, as fábricas de dados para população automática no banco de dados com dados para teste do sistema durante o período de desenvolvimento, permitindo ao desenvolvedor não perder tempo preenchendo tabelas com dados aleatórios em testes. E, por último, foram definidas as *seeders* a fim de, utilizando as *factories*, gerar uma quantidade delimitada de dados, assim como preencher algumas linhas de forma manual — como no caso da tabela de usuários, em que foi necessário criar um usuário manual, a fim de testar o acesso de áreas privadas do sistema durante o desenvolvimento.

##### 4.5.2. Organização de rotas, modelos e controladores

Antes da implementação de fato, organizou-se as rotas do sistema, que são os caminhos dentro do software — no desenvolvimento Web, isso é percebido na *URL*. As rotas feitas com o Laravel, além de organizarem os caminhos do sistema, têm o papel importante de mascarar a *URL* e tornar-la de fácil entendimento para o usuário. Cada rota chama um método de um determinado controlador (LARAVEL, 2018a). A partir da Figura 6, é possível perceber como funcionam as rotas. Dentro de uma rota com o prefixo “servicos” (*linha 1*) encontra-se, por exemplo, a “lixreira” (com requisição do tipo GET), usando o método “trash” do controlador “ServiceController” (*linha 4*). Ou seja, ao acessar a URL “/servicos/lixreira”, o programa chama a função “trash” da classe “ServiceController”. Neste exemplo, mostra-se também uma rota para busca de serviços (*linha 8*), passando o parâmetro da busca

entre chaves ({} ) — que corresponde ao que vai ser procurado. Mostra-se, ainda, rotas para descartar e restaurar informações — com requisição do tipo POST —, e para o histórico da tabela de serviços.

```

1. Route::prefix('servicos')->name('service.')->group(function(){
2.     Route::any('busca/{search}', ['uses' => 'ServiceConroller@search']->name('search');
3.     Route::get('{id}/historico', ['uses' => 'ServiceConroller@historic']->name('historic');
4.     Route::get('lixreira', ['uses' => 'ServiceConroller@trash']->name('trash');
5.     Route::post('{id}/descartar', ['uses' => 'ServiceConroller@discard']->name('discard');
6.     Route::post('{id}/restaurar', ['uses' => 'ServiceConroller@restore']->name('restore');
7. });
8. Route::resource('servicos', 'ServiceConroller', ['name'=> 'service']);

```

Figura 6: Exemplo de definição de rotas dentro do *framework* Laravel.

Para simplificar o uso de rotas, o Laravel disponibiliza o método “*resource*”, capaz de gerar rotas padrões. Cria-se rotas para: listagem de dados (*index*); demonstração das informações de um determinado dado (*show*); formulário para cadastro de dados (*create*); formulário para modificação de dados (*edit*); envio de dados de cadastro (*store*); envio de dados de modificação (*update*); exclusão de dados (*destroy*).

Tabela 2: Rotas padrões do *resource* para funções do *controller*  
(Disponível em: <https://laravel.com/docs/5.7/controllers>)

Ações tratadas pelo <i>resource</i>			
Verb	URI ( <i>Uniform Resource Identifier</i> )	Ação	Nome da rota
GET	/servicos	index	service.index
GET	/servicos/cadastro	create	service.create
POST	/servicos	store	service.store
GET	/servicos/{id}	show	service.show
GET	/servicos/{id}/modificacao	edit	service.edit
PUT/PATCH	/servicos/{id}	update	service.update
DELETE	/servicos/{id}	destroy	service.destroy

Após a criação das rotas, foram gerados os modelos — a partir de um comando no *prompt*<sup>6</sup> —, a fim de fazer a representação da tabela do banco de dados no código PHP. Para isso, foi preenchido o atributo *fillable* com uma matriz, em que consta o nome de todos os campos presentes no banco de dados.

Assim que concluídos os modelos, foram criados os controladores — a partir de outro comando no *prompt*<sup>7</sup> —, com a finalidade de conectar informações do banco de dados, entre outras operações lógicas, com as *views* correspondentes. Nos controladores foram implementados todos os métodos declarados nas rotas, cada um com a funcionalidade que lhe corresponde. Feito isso, foi implementado também o método *\_validate*, utilizado para fazer a validação de formulários preenchidos pelo usuário. Dentro dele, consta uma matriz associativa de regras para os dados recebidos, e outra para atribuir um nome a cada campo, para ser utilizado na mensagem de erro em caso de determinado campo não ter cumprido uma das regras. Estas duas matrizes são enviadas como parâmetros do método *\_validate* da classe *Request* — nativa do Laravel —, a qual verifica se as regras foram cumpridas e, caso não sejam, retornam uma mensagem de erro especificando o campo pelo nome atribuído a ele.

### 4.5.3. Interface gráfica do sistema

Antes de dar início ao desenvolvimento das *views*, foi feito um dicionário de palavras no idioma português do Brasil. O Laravel prevê a utilização de múltiplos idiomas, disponibilizando um diretório especialmente para armazená-las (*resource/lang*). Dentro deste diretório contém uma pasta para cada idioma a ser utilizado (LARAVEL, 2018b). Visto isso, todas as tabelas receberam um arquivo para tradução de dados e, dentro de cada arquivo, foi implementada uma matriz associativa capaz de ligar a chave a uma palavra traduzida, como no exemplo mostrado na Figura 7.

<pre>return [     'street' =&gt; 'Rua',     'number' =&gt; 'Número',     'complement' =&gt; 'Complemento',     'district' =&gt; 'Bairro',     'city' =&gt; 'Cidade',     'state' =&gt; 'Estado',     'country' =&gt; 'País',     'code' =&gt; 'CEP' ];</pre>	<pre>return [     'street' =&gt; 'Street',     'number' =&gt; 'Number',     'complement' =&gt; 'Complement',     'district' =&gt; 'District',     'city' =&gt; 'City',     'state' =&gt; 'State',     'country' =&gt; 'Country',     'code' =&gt; 'Postal code' ];</pre>	<pre>return [     'street' =&gt; 'Calle',     'number' =&gt; 'Número',     'complement' =&gt; 'Complemento',     'district' =&gt; 'Barrio',     'city' =&gt; 'Ciudad',     'state' =&gt; 'Estado',     'country' =&gt; 'País',     'code' =&gt; 'Código postal' ];</pre>
Língua portuguesa	Língua inglesa	Língua espanhola

<sup>6</sup> < php artisan make:model nomeDoModelo >

<sup>7</sup> < php artisan make:controller nomeDoControlador >

Figura 7: Funcionamento do suporte a múltiplos idiomas com o *framework* Laravel.

Nas *views* do Laravel, utilizando o *framework* Bootstrap<sup>8</sup>, fez-se uma interface padrão para a área administrativa a partir de um *template dashboard*, bem como a utilização de temas para a área do cliente. Para esta área, ainda, foi desenvolvido um sistema para modificação de temas, fazendo-se assim um sistema padrão com múltiplas *interfaces* gráficas possíveis. Dentro do diretório das *views* (*resource/view*) está localizado o espaço para a inserção os temas, em que cada pasta corresponde a um tema. Para este trabalho foi desenvolvido um tema para demonstração. A Figura 8 demonstra a personalização do sistema nas configurações do mesmo. No Apêndice A é possível visualizar o efeito da definição deste tema na interface da página inicial do sistema desenvolvido neste trabalho.

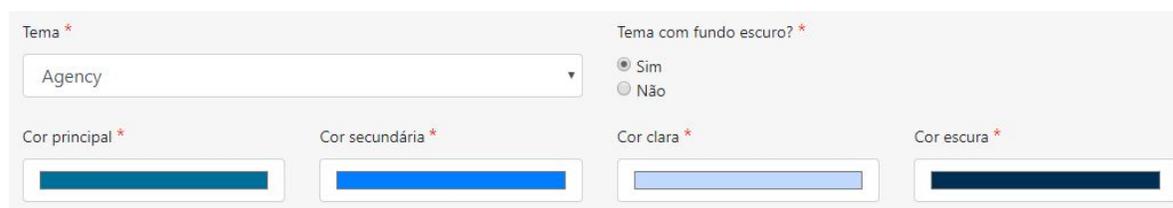
A imagem mostra um formulário de configuração de tema e cores. No topo, há um campo de seleção para o tema, atualmente com o valor "Agency". À direita, há uma opção de rádio para "Tema com fundo escuro?", com "Sim" selecionado. Abaixo, há quatro campos de cor: "Cor principal" (verde escuro), "Cor secundária" (azul), "Cor clara" (azul claro) e "Cor escura" (preto escuro).

Figura 8: Escolha do tema e das cores do sistema dentro do sistema desenvolvido.

## 5. Desenvolvimento do sistema

Nesta seção será abordada a resolução de todos os objetivos específicos propostos, de forma a expor o desenvolvimento das funcionalidades principais do sistema, bem como seus resultados atingidos. As próximas seções apresentam as *interfaces* criadas para a disponibilização das funcionalidades citadas para os diferentes perfis de usuários do sistema desenvolvido.

### 5.1 Gerenciamento de informações e de autorizações

Uma página padrão foi utilizada para fazer as operações de *CRUD* (criação, leitura, modificação e exclusão) do sistema, como pode-se observar na Figura 9. Com isso, ela serviu como base para todas as áreas do sistema, mantendo um padrão em todas as páginas, com botões para voltar, para cadastrar, para modificar, para visualizar detalhes, para excluir, para ver a lixeira, e para ver o histórico do item. Em algumas páginas, podem ser encontrados botões desabilitados, por não ter como fazer essa transição de páginas, ou por conta do usuário logado não ter permissão de realizar tal ação.

<sup>8</sup> Disponível em: <https://getbootstrap.com/>



Figura 9: Modelo de *layout* para as páginas que demandam operações de *CRUD*.

Elaborou-se também um sistema de autorizações para cada tipo de usuário, a fim de controlar o que cada um deles pode fazer no sistema. Com colunas do tipo *booleano*, fez-se uma tela para gerenciamento com campos *on/off*, como pode-se observar na Figura 10. Desta forma, o administrador pode escolher se o usuário deste tipo pode visualizar, modificar, apagar ou cadastrar cada uma das informações do banco de dados.

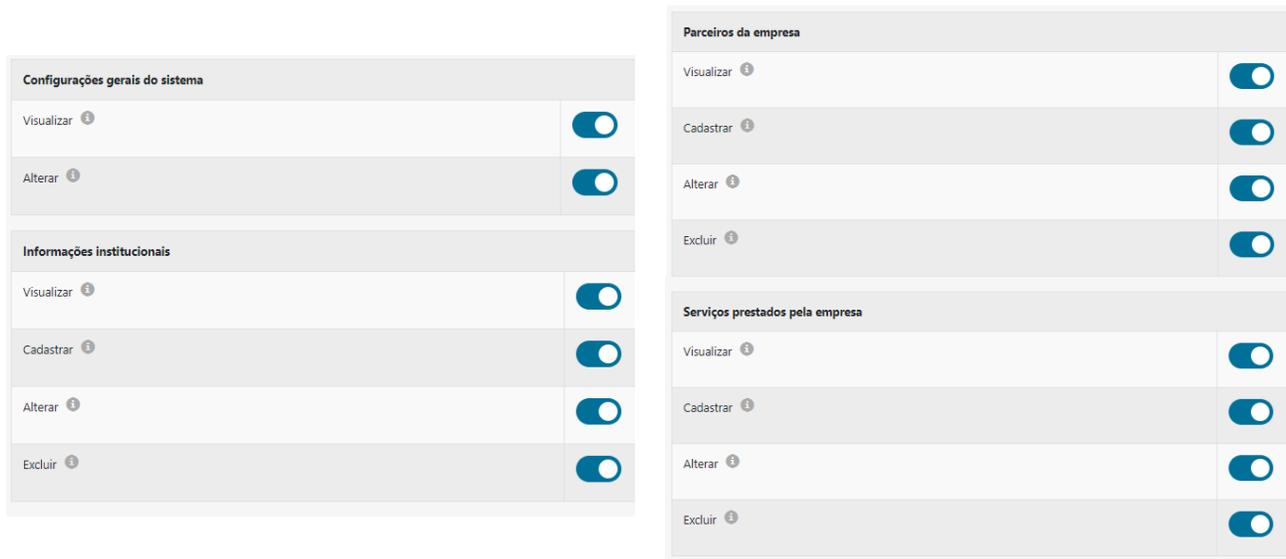


Figura 10: Autorizações de um tipo de usuário

## 5.2. Transparência sobre profissionais e equipamentos da academia

Essa plataforma comporta o gerenciamento de funcionários (Figura 11), assim como de equipamentos que a academia possui (Figura 12). Para cada funcionário, foi atribuído um currículo na forma de texto, assim podendo-se escrever livremente sobre suas experiências profissionais e acadêmicas. O cliente poderá, ainda, visualizar o tempo de empresa do funcionário, seja ele instrutor, recepcionista, atendente, ou que tenha qualquer outro cargo dentro da empresa. Para a transparência sobre informações de equipamentos, foi incluída a possibilidade de cada item possuir uma data de aquisição e uma data para cada manutenção feita. Desta forma, qualquer cliente tem acesso a essas informações, de forma a oferecer a transparência necessária sobre a vida útil do equipamento.

▼ **Dados pessoais**



**Nome completo:** Andrezza Molina  
**Nome de usuário:** andrezza\_molina  
**Data de nascimento:** 28/07/1990  
+ 16/06/2019 14:42

▼ **Profissão**

**Cargo na empresa:** Instrutora e professora  
**Currículo:**  
2010 - Formada como técnico em Aplicação da Educação Física, na UFRS;  
2010 - Estágio na XFit por 2 anos e meio;  
2013 - Recepcionista na XFit por 2 anos;  
2014 - Graduada em Nutrição, na UFRS;  
2014 - Professora na escola XSchool durante 3 anos;  
2016 - Pós graduada em Educação Física, na UFRS.  
2017 - Professora de nutrição na UFRS;  
2019 - Instrutora na academia iMaatz.

► **Biografia**

► **Contato**

► **Localização**

Figura 11: *Interface* desenvolvida para a transparência sobre funcionários.

Equipamento [CrossOver]








---

**Dados do equipamento**

**Código:** 001  
**Título do equipamento:** CrossOver  
**Data de aquisição:** 15/01/2019  
 30/05/2019 02:26:41  
 30/05/2019 02:26:41

**Manutenções do equipamento** 

03/06/2019	Aparelho consertado	 
29/05/2019	Peça X trocada do equipamento	 
25/02/2019	Limpeza do equipamento	 

Figura 12: *Interface* desenvolvida para a transparência sobre equipamentos.

É importante destacar que todas as informações referentes às *interfaces* de transparência de informações podem ser alteradas na área administrativa do sistema.

### 5.3. Guia

Para o desenvolvimento do guia foi utilizada uma relação entre três tabelas do banco de dados, são elas: equipamentos, exercícios e músculos. De acordo com Delavier (2006), um guia necessita de uma relação entre exercícios, músculos e equipamentos. Visto isso, pode-se afirmar as seguintes três correlações entre estas informações: (1) em um equipamento pode ser realizado vários exercícios, trabalhando vários músculos; (2) um exercício pode ser feito em vários equipamentos, e exercitar vários músculos; e, por fim, (3) um músculo pode ser exigido em vários equipamentos e exercícios. A *interface* desenvolvida para o guia está representada na Figura 13.

Peitoral
Supino reto

**Supino reto**



Assistir mais tarde Compartilhar

#ProjetoBomCorpo

**Descrição:** O supino é um exercício físico que é uma forma de levantamento de peso voltado principalmente para o treinamento dos músculos peitorais maiores, mas que também envolve, como sinergistas, os músculos deltóide, serrátil anterior, coracobraquial e tríceps braquial. Enquanto está deitado em decúbito dorsal, de preferência sobre um "banco" específico, o praticante abaixa uma barra com pesos até perto do peito [90º graus], e então a empurra para cima até que seus braços estejam esticados (ou próximos a isso).

**Aeróbico:** Não

Equipamentos para execução	Músculos afetados
<a href="#">Banco reto</a>	<b>Peitoral</b>
<a href="#">Barra reta</a>	Peitoral maior
<a href="#">Halteres</a>	Peitoral menor
	<b>Tríceps</b>
	Cabeça lateral do tríceps
	Cabeça medial do tríceps
	Cabeça longa do tríceps
	<b>Ombros</b>
	Deltóide

Figura 13: Interface desenvolvida para o guia.

A dinâmica de consulta do guia pelo usuário funciona com o seguinte procedimento. Inicialmente, o cliente seleciona um grupamento muscular; ao lado são listados todos os exercícios possíveis para o grupamento escolhido, dentro de um campo de seleção. Em seguida, o cliente seleciona um exercício; logo abaixo aparecem os detalhes do exercício escolhido, bem como todos os aparelhos que há a possibilidade de executá-lo, detalhes sobre os músculos que estão sendo trabalhados, além de mostrar um vídeo com a demonstração do exercícios. Todas as informações do guia podem ser gerenciadas na área administrativa do sistema.

#### 5.4. Fichas de treino

Cada aluno cadastrado no sistema pode ter uma ficha de treino digital. Para isso, será necessário o auxílio de um profissional da academia. Antes de gerar uma ficha de treino, é feita uma série de perguntas ao aluno (questionário da Figura 14), a fim de obter as informações que a instituição julgue importante antes gerá-la. Essas perguntas podem ser relacionadas à saúde do aluno, objetivos do aluno com a academia, ou até mesmo à frequência que o aluno se exercita. Respondendo o questionário, o cliente deve aguardar pela análise do instrutor, que criará uma ficha ao aluno a partir das respostas. Os

instrutores conseguem gerenciar todas as perguntas que devem ser feitas ao aluno na área administrativa do sistema.

### Questionário

Responda as perguntas abaixo, para iniciar o processo de criação de sua ficha.

<p>por Elias Appio Mezalira</p> <p><b>1</b> - Você possui algum problema físico, cardíaco, respiratório, ou qualquer outro problema médico, que dificulte a execução de algum exercício?</p> <p>Responda a pergunta...</p>	<p>por Elias Appio Mezalira</p> <p><b>2</b> - Há quanto tempo você pratica ou praticou exercícios/esportes?</p> <p>Responda a pergunta...</p>
<p>por Elias Appio Mezalira</p> <p><b>3</b> - Quais exercícios/esportes pratica ou já praticou? Com que frequência?</p> <p>Responda a pergunta...</p>	<p>por Elias Appio Mezalira</p> <p><b>4</b> - Qual é seu maior objetivo na academia? (Exemplo: emagrecer/ganhar massa/qualidade de vida/praticar bodybuilding/etc)</p> <p>Responda a pergunta...</p>

Figura 14: *Interface* desenvolvida para o questionário realizado para a definição das fichas de treino.

Cada item da ficha do aluno tem um dia da semana, um exercício — em ordem a ser escolhida pelo aluno e pelo instrutor —, e a quantidade de repetições e séries. Para o caso de exercícios aeróbicos, há a inclusão do tempo que deve ser realizado o exercício. Para ilustração da *interface* desenvolvida para a ficha, a Figura 15 apresenta um exemplo de organização de exercícios a serem executados pelo aluno. No banco de dados, cada cliente tem uma resposta para cada pergunta do questionário, e uma lista de itens da ficha.

Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira	Sábado
Supino reto	Agachamento	Puxada pull-down	Supino reto	Agachamento	Puxada pull-down
Supino inclinado	LegPress	Remada baixa	Supino inclinado	LegPress	Remada baixa
Crucifixo reto	Levantamento terra	Remada inclinada	Crucifixo reto	Levantamento terra	Remada inclinada
Crucifixo inclinado	Extensão de pernas	Pullover	Crucifixo declinado	Extensão de pernas	Pullover
Tríceps na polia alta	Flexão de pernas	Rosca direta	Tríceps na polia alta	Flexão de pernas	Rosca direta
Tríceps francês	Elevação de panturrilhas	Rosca inclinada	Tríceps francês	Elevação de panturrilhas	Rosca inclinada
Tríceps coice		Crucifixo invertido	Tríceps coice		Crucifixo invertido
Desenvolvimento de ombros	Abdominal	Encolhimento de ombros	Desenvolvimento de ombros	Abdominal	Encolhimento de ombros
	Rotação abdominal			Rotação abdominal	
Elevação lateral	<a href="#">+ Novo exercício</a>	<a href="#">+ Novo exercício</a>	Elevação lateral	<a href="#">+ Novo exercício</a>	<a href="#">+ Novo exercício</a>
<a href="#">+ Novo exercício</a>			<a href="#">+ Novo exercício</a>		
<b>Domingo</b>					
Descanso					
<a href="#">+ Novo exercício</a>					

Figura 15: *Interface* desenvolvida para as fichas de treino.

## 5.5. Evolução do aluno

Cada cliente cadastrado no sistema pode acompanhar sua evolução, como peso, medidas corporais, e a carga que ele suporta em cada exercício que deseja monitorar. Para isso, o aluno deve atualizar seus próprios dados através do sistema, formando assim um histórico de seu peso, e demais medidas.

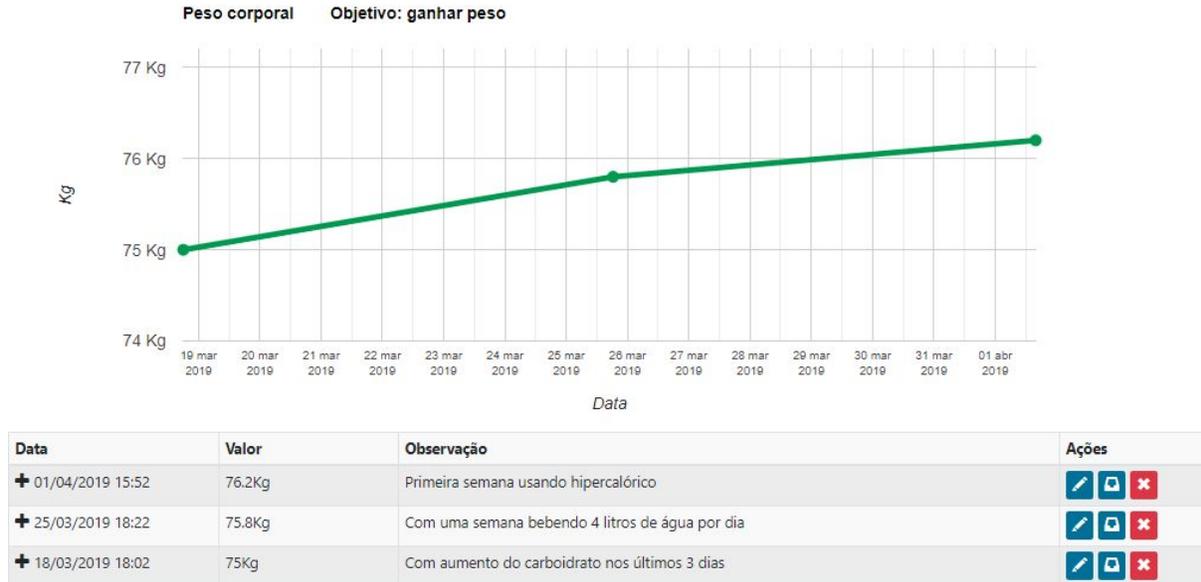


Figura 16: *Interface* desenvolvida para o acompanhamento da evolução de medidas corporais do aluno.

O aluno escolhe quais itens para acompanhamento da evolução ele quer monitorar. Como na captura de tela da Figura 16, ele pode clicar em uma de suas medidas e, assim, visualizar um gráfico com toda a trajetória daquela medida, desde seu valor inicial até o atual.

## 6. Conclusão

Os alunos e as academias demandam por meios tecnológicos para acompanhamento de desempenho, e organização de treinos. Dessa forma, este trabalho poderá auxiliar na saúde e estética corporal das pessoas que usufruem dos serviços da academia, visto que essas empresas carecem desse tipo de serviço, e o público-alvo delas aguarda por isso.

Após o desenvolvimento deste trabalho, serão feitas melhorias em questões visuais do *software*, estudando melhores maneiras do cliente interagir com os dados da ficha de treino e das evoluções. Além disso, o sistema foi construído pensando na implementação de gerenciamento de temas (criado apenas um para mostra), fazendo com que o sistema de cada empresa tenha uma identidade visual mais próxima à sua. Serão feitos, também, testes mais profundos no sistema, corrigindo eventuais *bugs* e deixando o sistema com a menor chance de falha possível, além da possibilidade de desenvolvimento de versões *desktop* e *mobile*, com mais funcionalidades a serem pensadas.

Visto a importância da tecnologia em academias, o produto final deste trabalho será oferecido a empresas do ramo, após um estudo sobre a melhor maneira de implantar o sistema na sociedade. Depois disso, será feito um levantamento com usuários sobre o que o sistema acrescentou na sua rotina

e no seu relacionamento com a academia, e o que pode ser melhorado, sempre visando atender a todo o público da forma mais satisfatória possível.

O *software* implementado neste trabalho faz parte de um projeto maior e mais ambicioso. Denominado Projeto Vex e, pensado de forma modular, o sistema final abrangerá além da academia, outros empreendimentos como lojas do comércio, empresas de mídias em geral, e restaurantes com tele-entrega. Dividido em seis módulos, sendo um deles o núcleo e outro a academia, o projeto tem como papel principal a busca por algo mais próximo da padronização da regra de negócio de cada estabelecimento, a fim de alcançar pequenas e médias empresas com um custo reduzido — já que o mesmo código contemplaria várias empresas, e cada empresa o customizaria da sua maneira. Desta forma, cada empresa poderá disponibilizar o sistema na internet para seus clientes de forma mais simples. Os módulos do Projeto Vex são: núcleo, academia, vendas, mídias, blog e chat.

## 7. Referências

ALBÉ, J. F. **O site organizacional como estratégia de comunicação**. Dissertação de mestrado (Ciências da Comunicação) - Universidade do Vale do Rio dos Sinos, São Leopoldo, 2007.

BARROS, T; SILVA, M; ESPÍNOLA, E. **State MVC: Estendendo o padrão MVC para uso no desenvolvimento de aplicações para dispositivos móveis**. Centro de Estudos e Sistemas Avançados do Recife, 2007.

BENTO, A. R; BALAN, D; NEGRELLO, D. G; HORI, H. H. **Importância da tecnologia ao acompanhamento das atividades dos alunos de academia**. Artigo Convibra - Faculdade Santa Cruz, Curitiba, 2013.

BRITO, W. G. F; MOTA, F. A. O; FILHO, E. F. S. **Análise do perfil do usuário de academia na cidade de Januária: acompanhamento profissional e utilização de aplicativos**. VI Seminário de Iniciação Científica - Instituto Federal do Norte de Minas Gerais Campus Almenara, Januária, 2017.

COBRA, Nuno. **Atividade física é qualidade de vida**. Isto É Gente, Rio de Janeiro, edição 189, mar. 2003. Disponível em: <<http://www.terra.com.br/istoegente/189/saude>>.

ESPÍNDOLA, Evandro (2017). **A importância da Modelagem de Objetos no Desenvolvimento de Sistemas**. [online] Linhadecodigo.com.br. Disponível em: <<http://www.linhadecodigo.com.br/artigo/1293/a-importancia-do-modelagem-de-objetos-no-desenvolvimento-de-sistemas.aspx>>. Acesso em: 23 Nov. 2018.

JQUERY; 2018a. **What is JQuery?**. Disponível em: <<https://jquery.com/>>. Acesso em: 24 nov. 2018.

\_\_\_\_\_ ; 2018b. **jQuery.ajax()**. Disponível em: <<http://api.jquery.com/jquery.ajax/>>. Acesso em: 24 nov. 2018.

LARAVEL; 2018a. **Routing**. Disponível em: <<https://laravel.com/docs/5.7/routing>>. Acesso em: 24 nov. 2018.

LARAVEL; 2018b. **Localization**. Disponível em: <<https://laravel.com/docs/5.7/localization#overriding-package-language-files>>. Acesso em: 24 nov. 2018.

MDN, Mozilla Web Docs; 2019a. **JavaScript**. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 22 nov. 2018.

MDN, Mozilla Web Docs; 2019b. **HTML**. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 22 nov. 2018.

MDN, Mozilla Web Docs; 2019c. **CSS**. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 22 nov. 2018.

MYSQL; 2019a. **What is MySQL?**. Disponível em: <<https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>>. Acesso em: 30 nov. 2018.

MYSQL; 2019b. **MySQL Workbench**. Disponível em: <<https://dev.mysql.com/doc/workbench/en/>>. Acesso em: 23 nov. 2018.

MYSQL; 2019c. **Design e Modelagem de Banco de Dados**. Disponível em: <<https://dev.mysql.com/doc/workbench/en/wb-data-modeling.html>>. Acesso em: 23 nov. 2018.

NGUYEN, Q. H. **Building a Web Application with Laravel 5**. Universidade de Ciências Aplicadas de Oulu, Finlândia, 2015.

PHP; 2019. **What is PHP?**. Disponível em: <[https://secure.php.net/manual/pt\\_BR/intro-what-is.php](https://secure.php.net/manual/pt_BR/intro-what-is.php)>. Acesso em: 4 jun. 2019.

PHPMYADMIN; 2019. **Bringing MySQL to the web**. Disponível em: <<https://www.phpmyadmin.net/>>. Acesso em: 4 jun. 2019.

PRAZERES, M. V. **A prática da musculação e seus benefícios para a qualidade de vida**. Centro de Educação Física, Fisioterapia e Desportos - Universidade do Estado de Santa Catarina, Florianópolis, 2007.

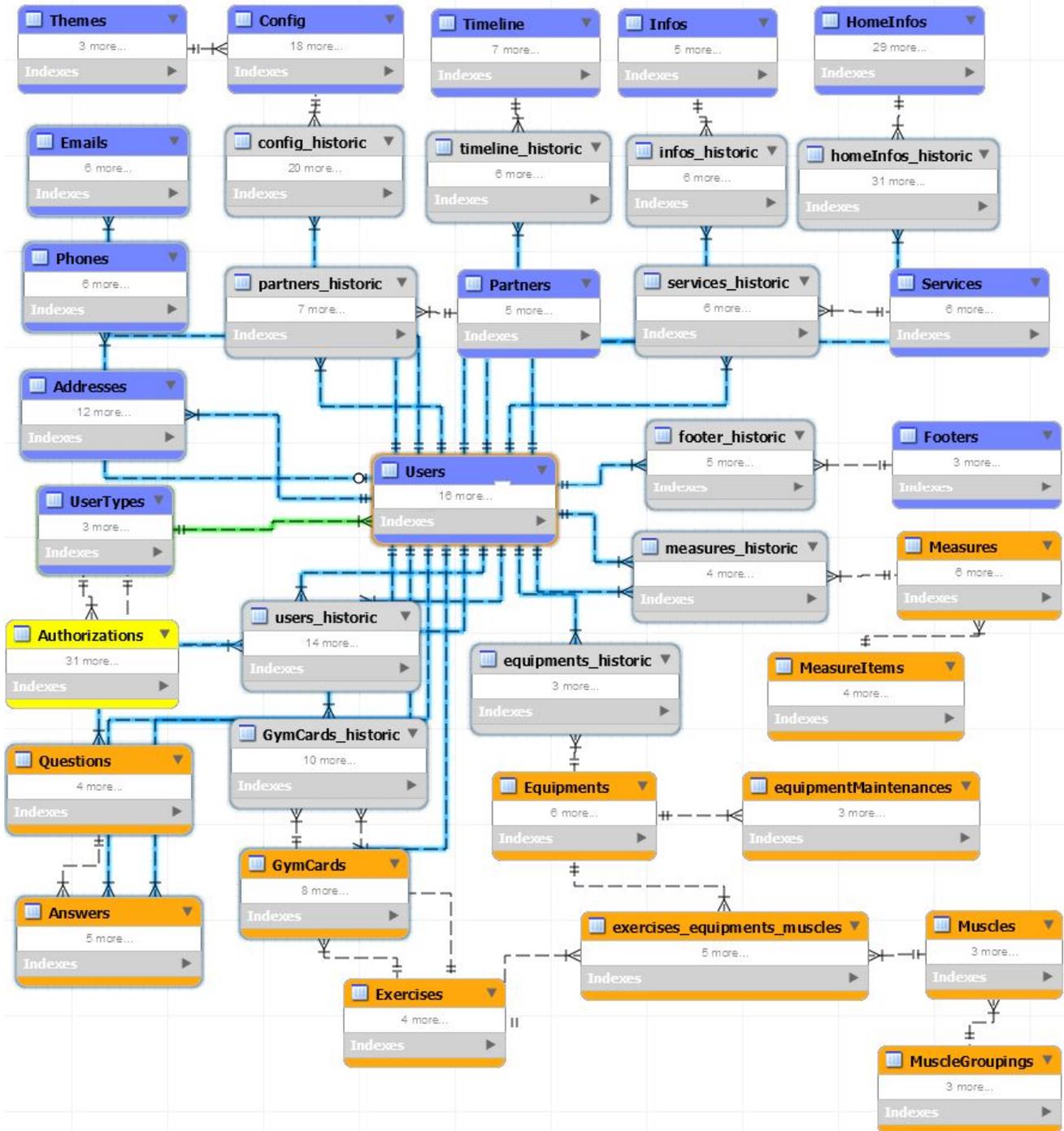
PINOCHET, L. H. C. **Tendências de Tecnologia de Informação na Gestão da Saúde**. Revista O Mundo da Saúde. Centro Universitário São Camilo. v. 35, p. 382-394, São Paulo, 2011.

RUÃO, Tereza. **A organização comunicativa: teoria e prática em Comunicação Organizacional**. Braga, Portugal: CECS - Centro de Estudos de Comunicação e Sociedade Universidade do Minho, 2016. E-book.

SILVA, W. A . L. **Laravel Tutorial**. DevMedia, 2015. Disponível em:  
<<https://www.devmedia.com.br/laravel-tutorial/33173>>. Acesso em: 23 nov. 2018.

WAMP, 2019. **WampServer: plate-forme de développement Web sous Windows**.  
<<http://www.wampserver.com/en/>>. Acesso em: 4 jun. 2019.

## Apêndice A - Diagrama lógico de banco de dados



## Apêndice B - Página inicial do sistema com o tema visual aplicado

