Especificação Projeto Final – POO

Disciplina: SCC0103-2019102 – Programação Orientada a Objetos
Professor: Dr. Márcio Delamaro
Aluno PAE: Claudinei Jr e Misael Jr

Grupo 4:
Eduardo Pennone 8937246
Henrique Santos 10819029
Leonardo Luiz 10851691
Witor Oliveira 10692190

1. Aplicação

Área: Apoio pessoal e qualidade de vida
Projeto: Plataforma para apoio a atividade física

2. Descrição e proposta

A prática de atividades físicas é fundamental para melhoria da qualidade de vida. A combinação de dieta balanceada e rotina de exercícios físicos resultam em um organismo saudável e na prevenção de doenças. Portanto, uma rotina de exercícios deve ser adotada por pessoas de todas as idades. Esse projeto tem como objetivo desenvolver uma aplicação web que:

- permita que o usuário tenha acesso a um conjunto de atividades para realizar diariamente;
- ofereça uma configuração de perfil para o usuário realizar atividades físicas específicas (aeróbica etc.);
- envie notificações ao usuário sobre desempenho na atividade realizada e reporte melhorias.

3. Objetivo principal

Em um mundo em que as pessoas em geral possuem pouco tempo disponível, por muitas das vezes, a saúde acaba ficando em segundo plano. Isso pode acarretar problemas sérios quando a pessoa atingir uma certa idade.

Com o intuito de ajudar essas pessoas, esse trabalho tem como seu maior objetivo a implementação de algoritmos, estruturas e heurísticas de programação orientada a objetos vistos na disciplina SSC0103 - Programação Orientada a Objetos, aplicados a problemas reais.

Tendo o conhecimento das causas das dificuldades da prática diária de atividade física, o projeto tem como principal foco estimular o usuário a praticar atividade física sem obrigá-lo a sair de casa e permitindo que ele tenha flexibilidade de tempo e espaço, além de noção de quais atividades ele deve exercer com base em seu objetivo e o seu progresso desde o início do uso do aplicativo.

Ao tratarmos de problemas reais, a realização do projeto nos permite solucionar problemas que iremos enfrentar no mercado de trabalho, fazendo com que a graduação seja mais abrangente, não se restringindo apenas a trabalhos técnicos e acadêmicos.

Com esse projeto, também desejamos ter uma experiência prática de como é produzir um programa que pode servir como um produto final, aproximando o curso da realidade do mercado.

4.Objetivo específico

Aplicar os conhecimentos adquiridos na disciplina SCC0103 de forma a criar um programa bem estruturado, fácil e agradável de se utilizar e que resolva problemas reais de nossa sociedade.

5. Funcionalidades a serem desenvolvidas

Uma interface que permita ao usuário escolher um perfil de treino de acordo com seus objetivos. Cada perfil de treino possuirá exercícios específicos divididos em grupos:

- Treinos de força:
    - Corpo inteiro
        - Treino curto (20 min)
            - Aquecimento
            - Flexão de Braço Fechada
            - Barra Fechada
            - Agachamento
        - Treino médio (40 min)
            - Aquecimento
            - Flexão de Braço Aberta
            - Flexão de Braço Fechada
            - Barra Aberta
            - Barra Fechada
            - Agachamento
            - Flexão de Coxa Unilateral
        - Treino longo (1h)
            - Aquecimento
            - Flexão de Braço Aberta
            - Flexão de Braço Fechada
            - Tríceps Testa
            - Barra Aberta
            - Barra Fechada
            - Rosca Alternada
            - Agachamento
            - Flexão de Coxa Unilateral
            - Gêmeos
    - Peito, ombro e tríceps

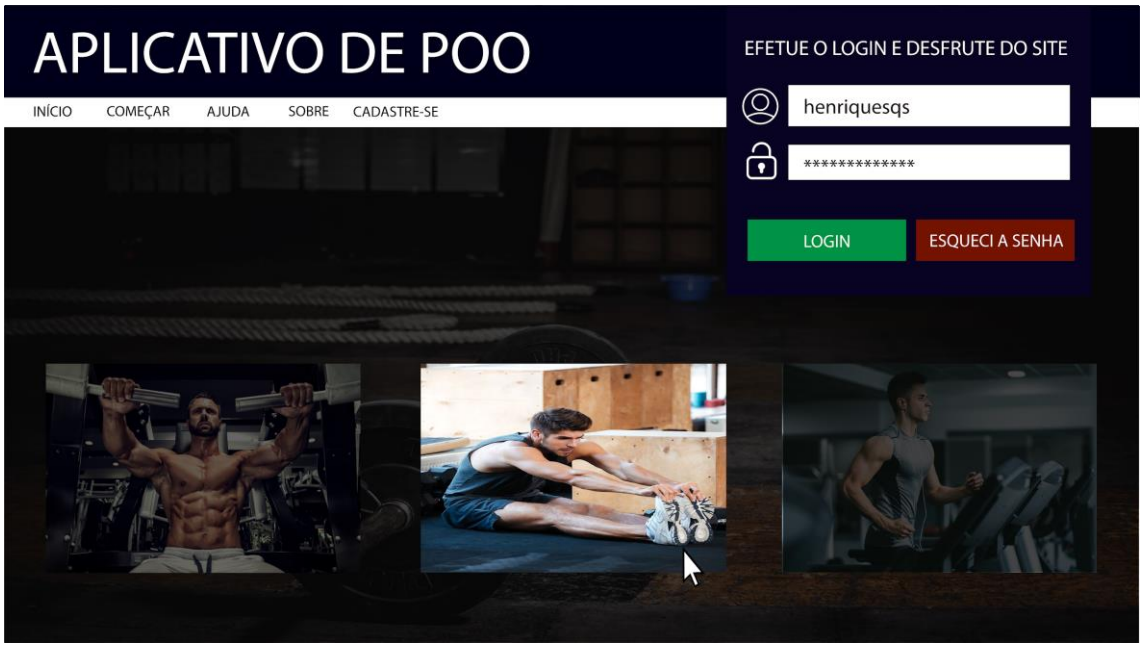
        - Treino curto (20 min)
            - Aquecimento
            - Flexão de Braço Aberta
            - Desenvolvimento Livre
            - Tríceps Testa
        - Treino médio (40 min)
            - Aquecimento
            - Flexão de Braço Aberta
            - Flexão de Braço Fechada
            - Desenvolvimento Livre
            - Elevação Lateral
            - Tríceps Testa
            - Tríceps Francês
        - Treino longo (1h)
            - Aquecimento
            - Flexão de Braço Aberta
            - Flexão de Braço Fechada
            - Crucifixo
            - Desenvolvimento Livre
            - Elevação Lateral
            - Elevação Frontal
            - Tríceps Testa
            - Tríceps Francês
            - Tríceps Coice
    - Costas, bíceps e abdômen
        - Treino curto (20 min)
            - Aquecimento
            - Barra Aberta
            - Rosca Alternada
            - Abdominal Frontal
        - Treino médio (40 min)
            - Aquecimento
            - Barra Aberta
            - Barra Fechada
            - Rosca Alternada
            - Rosca Martelo
            - Abdominal Frontal
            - Abdominal Lateral
        - Treino longo (1h)
            - Aquecimento
            - Barra Aberta
            - Barra fechada
            - Crucifixo Inverso
            - Rosca Alternada

                - Rosca Martelo
                - Rosca Concentrada
                - Abdominal Frontal
                - Abdominal Lateral
                - Prancha
        - Membros inferiores
            - Treino curto (20 min)
                - Aquecimento
                - Agachamento
                - Flexão de Coxa Unilateral
                - Gêmeos
            - Treino médio (40 min)
                - Aquecimento
                - Agachamento
                - Afundo
                - Flexão de Coxa Unilateral
                - Stiff
                - Glúteos
                - Gêmeos
            - Treino longo (1h)
                - Aquecimento
                - Agachamento
                - Afundo
                - Chute
                - Flexão de Coxa Unilateral
                - Stiff
                - Glúteos
                - Elevação de Perna Lateral
                - Coice
                - Gêmeos
- Treino de cárdio:
    - Cárdio completo:
        - Treino curto (20 min)
            - Polichinelo
            - Pique no Lugar
            - Joelhos Altos
            - Pés nos Glúteos
        - Treino médio(40 min)
            - Polichinelo
            - Pique no Lugar
            - Joelhos Altos
            - Pés nos Glúteos
            - Pulando Corda
            - Pulos Laterais

                - Patinador
            - Treino longo (1h)
                - Polichinelo
                - Pique no Lugar
                - Joelhos Altos
                - Pés nos Glúteos
                - Pulando Corda
                - Pulos Laterais
                - Patinador
                - Chute Alto
                - Escalador
                - Pulos com Agachamento
- Treino de alongamento:
    - Alongamento Completo
        - Treino curto (15 min)
            - Braços a Frente do Peito
            - Braços Atrás da Cabeça
            - Alongamento Lateral
            - Alongamento de Posterior de Coxa e Panturrilha
            - Espacate
            - Alongamento de Quadríceps
        - Treino longo (30 min)
            - Alongamento de Pescoço
            - Alongamento de Pulso
            - Braços a Frente do Peito
            - Braços Atrás da Cabeça
            - Alongamento Peitoral
            - Rotação de Tronco
            - Alongamento Lateral
            - Alongamento de Glúteos
            - Alongamento de Posterior de Coxa e Panturrilha
            - Alongamento Borboleta
            - Espacate
            - Alongamento de Quadríceps
- Histórico de treinos:
    - Última Semana
    - Último Mês
    - Último Ano
    - Histórico Completo

6. Protótipos:

Página inicial: Nesta página, a ideia era apresentar uma caixa de login para o usuário, junto com uma opção de "**esqueci a senha**". Os dados cadastrados pelo usuário seriam armazenados em um banco de dados (a decidir), onde permitiria que o aplicativo oferecesse **histórico de exercícios praticados**, **evolução ao longo do uso do aplicativo** e outros dados estatísticos ainda não decididos pelo grupo. No centro do site, imaginamos que poderíamos oferecer imagens e descrições dos tipos de exercícios que nós estamos oferecendo - **treino de cardio**, **força** e **alongamento**. Por fim, a "barra de ferramentas" do site provavelmente oferecerá as opções de **início** (voltar para essa mesma página inicial), **começar** (um mini conjunto de treinos preparados por nós - não decidido ainda), **ajuda** (uma espécie de *Frequent Asked Questions* - FAQ - do uso do site), **sobre** (uma tela de informações sobre os propósitos da criação do aplicativo, desenvolvedores e outras informações adicionais) e, por fim, **cadastre-se** (tela de cadastramento do usuário).



Menu da opção selecionada (neste exemplo, treino de alongamento - imagem central da página inicial): no menu lateral localizado a direita da página, no topo exibiremos o nome do treino selecionado, ao lado de uma imagem descritiva. Logo abaixo, as opções de treinos seriam oferecidas (neste caso, o treino de alongamento oferece o **treino curto** e o **treino longo**, como explicado anteriormente). Algumas informações iniciais seriam exibidas na tela, como o **tempo de duração do treino**, **objetivo do treino** e a opção de **iniciar o treino**. Ao lado, na parte vazia, o grupo ainda não decidiu o que colocar, mas a ideia era preencher com algum tipo de visualização que permitiria ao usuário ter um contato maior com o treino antes de iniciá-lo efetivamente.

Exercícios do treino selecionado (neste exemplo, “selecionamos” o treino curto): a ideia inicial do grupo é oferecer um conjunto de imagens descritivas dos exercícios oferecidos ao usuário de acordo com o treino selecionado anteriormente. Essa página seria uma espécie de menu, onde o usuário apenas selecionaria o exercício desejado, clicando sobre a imagem.

Apresentação do exercício selecionado (alongamento de posterior de coxa e panturrilha, neste exemplo): a ideia do grupo é oferecer uma **descrição mais detalhada sobre o exercício selecionado** anteriormente pelo usuário, juntamente com a **duração esperada deste**. Ao lado dessas informações, exibiremos a mesma imagem do menu anterior e, abaixo desta, um vídeo explicativo/ demonstrativo do exercício sendo feito por um profissional. As opções de **iniciar exercício** e **voltar para página anterior** seriam oferecidas ao lado.

7. O que foi desenvolvido

Para facilitar o entendimento do leitor, o grupo decidiu separar o projeto em partes e, assim, comentar o que foi incluso e descartado durante o desenvolvimento do projeto.

7.1 **Desenvolvimento backend**

Antes de tudo, é de extrema importância mencionarmos o código da aplicação. Utilizamos o **framework Spring** do Java , onde, a partir dela e suas dependências, fizemos a conexão entre o backend (totalmente feito em Java) e o frontend (HTML/ CSS). As dependências utilizadas foram:

- **Spring MVC**: responsável por fornecer a arquitetura MVC (Model-View-Controller), que controla (e separa) a base do aplicativo (código-fonte) com o que é apresentado ao usuário (*interface*).

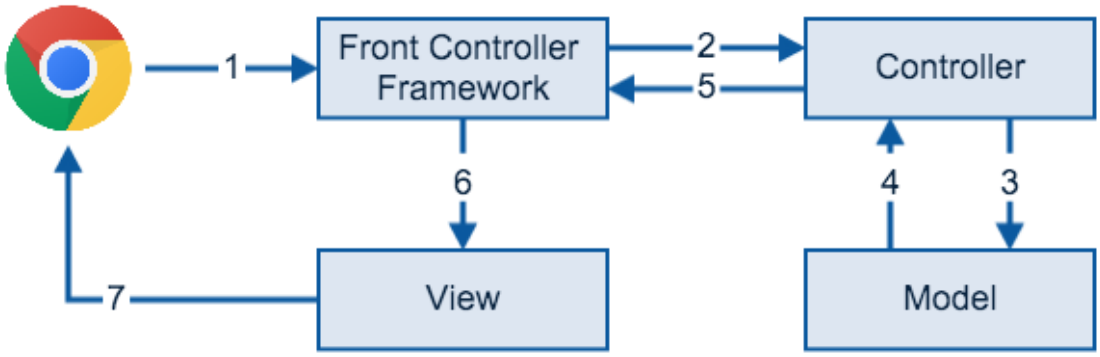


Imagem 1: esquema do funcionamento do padrão MVC, retirada deste **site.**

Detalhadamente:

- O ***Model*** encapsula os dados da aplicação;
- O ***View*** é responsável por fornecer a visualização das páginas ao usuário;
- O ***Controller*** é quem controla as *requests* (requisições) advindas do usuário e define qual dado será passado para ele.

Decidimos ir em busca de mais conhecimento sobre o MVC e acabamos encontrando uma representação em "diálogo" (como definiu o autor) do funcionamento do MVC. Achamos simples e decidimos corrigir erros ortográficos, editar algumas partes e mostrar nesse documento, pensando que facilitaria o entendimento do leitor:

"O diálogo das camadas na Web

View - Fala *Controller*! O usuário acabou de pedir para acessar o Facebook! Pega os dados de login dele aí.

Controller – Beleza. Já te mando a resposta. Aí *Model*, meu parceiro, toma esses dados de login e verifica se ele pode logar.

Model – Os dados são válidos. Mandando a resposta de login.

Controller – Beleza. *View*, o usuário informou os dados corretos. Vou mandar pra vc os dados dele e você carrega a página de perfil.

View – Valeu. Mostrando ao usuário…

OBS: Este texto foi retirado de meu artigo do site Tableless: MVC, Afinal, é o quê ?" [1]

[1] Retirado de https://pt.stackoverflow.com/questions/55486/o-que-é-mvcmodel-view-controller

- **Spring Boot**: fornece facilidades na hora de rodar a aplicação, principalmente por possuir o servidor Tomcat embutido, o que dispensa a necessidade de configurações externas para rodar a aplicação na web.

- **Spring Security:** fornece as ferramentas necessárias para autenticar o *login* e registro do usuário, além de controlar as permissões que cada usuário possui (nesse aplicativo, todos os usuários possuem permissão – *role* (cargo) – ADMIN, mas, em aplicações mais densas, podem existir diferentes cargos para cada usuário. Um exemplo seria um sistema de universidade onde existiriam os cargos *estudante*, *professor*, *administrador*.

- **Spring Data JPA**: essa dependência facilita a comunicação com a base de dados, tornando possível a utilização de queries (linhas de consulta ao banco de dados) durante o desenvolvimento do código-fonte.

- **Spring Boot Thymeleaf**: O Thymeleaf é uma biblioteca que auxilia na visualização (*View* do modelo MVC supracitado) do HTML utilizado no nosso projeto.

Além do Spring, utilizamos uma base dados, já citada anteriormente. Trata-se do **MySQL**, que foi utilizado para salvar as informações dos usuários e outros dados relacionais da aplicação.

Imagem 2: visualização, através do aplicativo web *phpMyAdmin* que permite a administração do banco de dados pela web.

Abaixo, detalhamos o que cada tabela se refere:

1. **Hibernate_sequence**: responsável por salvar e fazer a comunicação entre as classes e objetos feitos no código JAVA com a base de dados;
2. **Persistent_logins**: no começo do desenvolvimento do código-fonte, tivemos a ideia de implementar o "lembre-me", que seria deixar os dados do usuário salvo para que ele não precisasse inserir eles novamente. A ideia não foi adiante pois o grupo julgou desnecessário para o momento;
3. **Role**: essa tabela armazena todos os cargos (tipos de permissões) disponíveis na aplicação. No momento, a aplicação só possui um (que dá acesso total ao site): ADMIN.
4. **User**: salva todas as informações do usuário: primeiro nome, último nome, email, senha, *active* (define o usuário como ativo (1) – caso o usuário

deletasse a conta, apenas definiríamos o ‘active’ para 0), fez_exercicio1, fez_exercicio2, fez_exercicio3 que, respectivamente, definem se um usuário já concluiu os treinos 1, 2 e 3 (mais detalhes sobre isso serão dados a frente).

5. **User_role**: salva qual o *role* (cargo) definido para o usuário. No momento, todos os usuários cadastrados recebem o cargo de ADMIN.

7.2 **Desenvolvimento frontend**

Essa parte do relatório focará mais na interface gráfica e ferramentas desenvolvidas na aplicação, deixando explícito o que, de fato, foi feito seguindo o protótipo antes criado e o que não foi desenvolvido, mencionando os motivos para tal.

7.2.1 **Página Inicial**:

Não há muita diferença entre a página inicial pensada com a concluída. Começando pela *navbar* (barra de navegação), antes o grupo pensou em deixar a mostra a opção ‘COMEÇAR’, o que foi descartado. Anteriormente, pensamos em deixar o usuário desfrutar de um treino e só poder acessar os outros após o cadastro, mas descartamos essa ideia já que fugia do propósito do projeto. Ademais, as imagens utilizadas são outras e a disposição delas também foi alterada (agora estão exibidas em forma de *slide*). Por fim, não implementamos a funcionalidade de “esqueci a senha”, já que necessitava de conhecimentos que ninguém no grupo tinha até o momento e achamos desnecessário gastar tempo com isso.

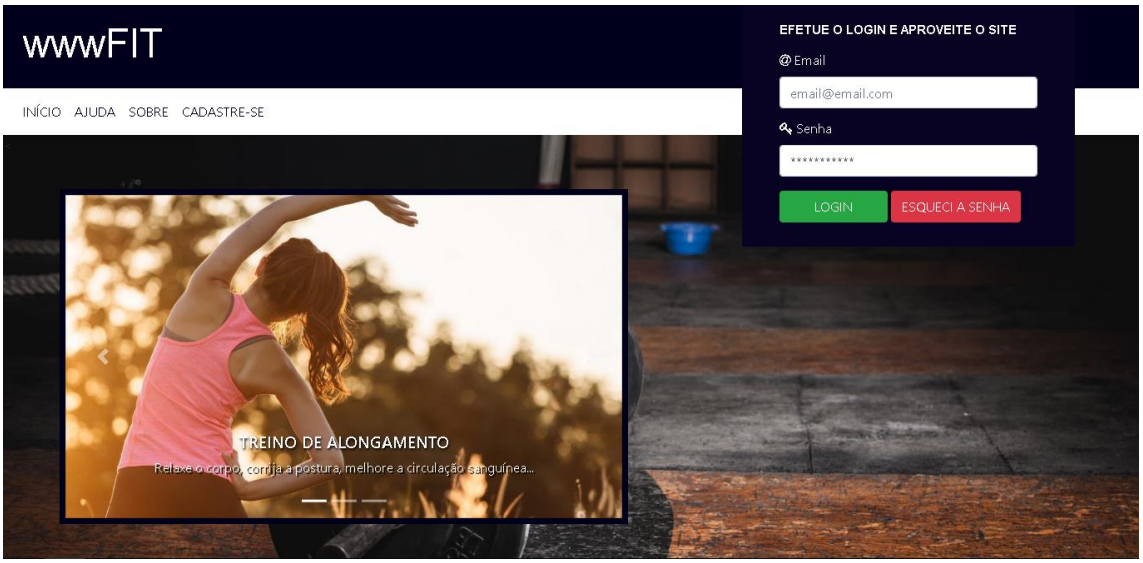


Imagem 3: visualização da página inicial do site

7.2.2 **Página pós login** (não citada no protótipo)

Nessa página, iremos exibir os 3 tipos de treinos disponíveis. Além disso, uma aba será exibida contendo o progresso feito pelo usuário até agora (quantos exercícios que ele fez até o momento). Para isto, foi criado um objeto que recebe da base de dados quais exercícios ele fez (aqui se faz uso das colunas "fez_exercicio1, fez_exercicio2 e fez_exercicio3", onde, caso os valores sejam verdadeiros (1), o usuário será dito ter feito um progresso de 20%, 30% e 50%, respectivamente). Concluindo, o grupo pensou (e até deixou a vista o botão) em implementar uma página de preferências, onde seria possível alterar os dados da conta (primeiro e último nome, senha e email) mas descartou a ideia por falta de tempo.

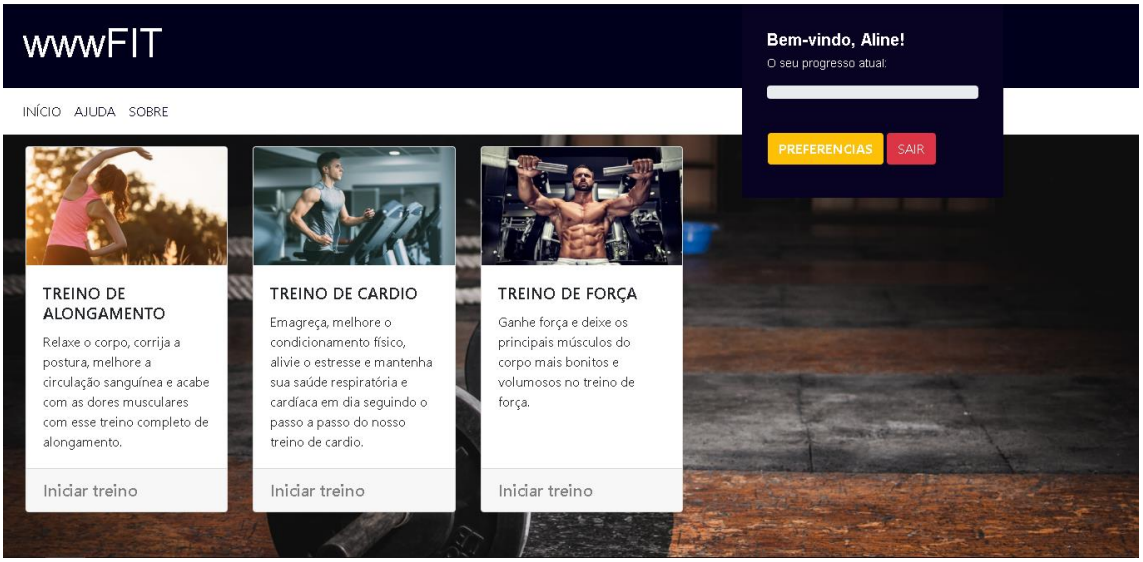


Imagem 4: exibição da página principal após o usuário ter efetuado o login

7.2.3 **Menu da opção selecionada** (neste exemplo, treino de alongamento)

Seguindo o que foi mostrado no protótipo, utilizamos o treino de alongamento para exemplificar o modelo final da página de treinos. Agora, além do menu da opção selecionada, adicionamos uma imagem e uma breve descrição do que o treino se trata. Mantivemos o menu contendo as opções do treino (treino curto e longo, neste caso).

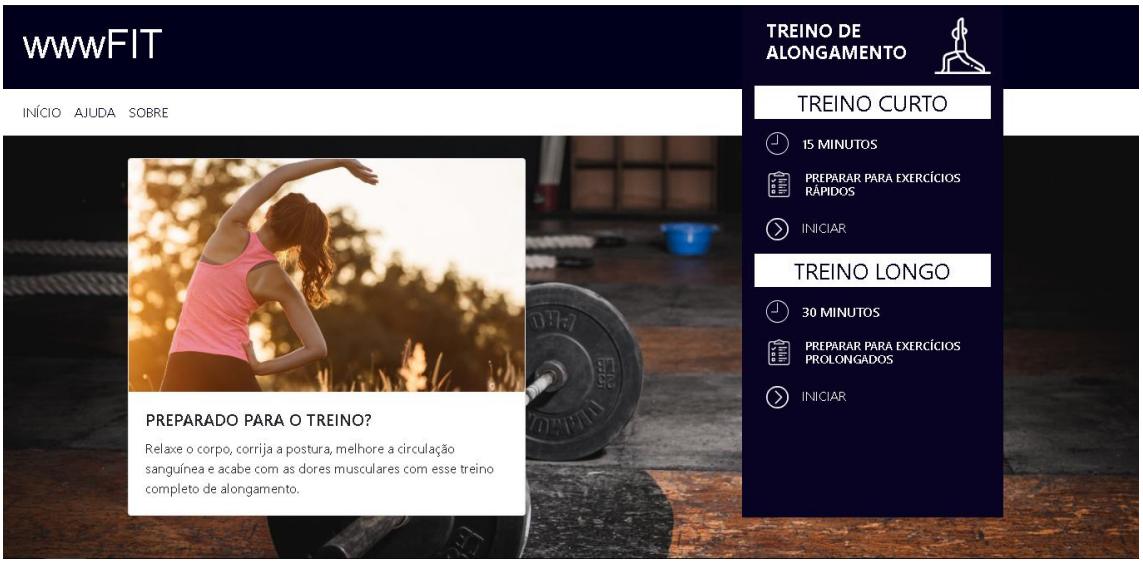


Imagem 5: visualização do menu da opção de treino selecionada.

7.2.4 **Exercícios do treino selecionado** (neste exemplo, selecionamos o treino curto)

Nessa página, a primeira coisa que o usuário terá interação será com um aviso (imagem 6) onde o aplicativo o alerta sobre como funcionará o treino que ele selecionou, bem como os seguintes exercícios. Ao fechar o aviso, o usuário terá acesso a 3 campos: um contém o nome do exercício que ele irá realizar no momento, bem como o tempo de duração ideal (se possuir) e uma imagem demonstrativa. Um vídeo explicativo foi pensado, porém descartado por falta de recursos (não conseguimos achar nada que fosse de uso viável). Na caixa central, estão os exercícios que o usuário deverá realizar, em sequência. Na caixa mais a direita, têm as opções que o usuário tem relacionadas aquele exercício: **próximo exercício** leva o usuário para o exercício seguinte; **desistir do treino curto** leva o usuário de volta ao menu de seleção de treino (7.2.3 neste relatório), onde ele pode optar por escolher outro tipo de treino (o longo, por exemplo); e **voltar ao menu de treinos**, onde o usuário poderá selecionar outro tipo de treino (cardio ou força, por exemplo). Por fim, ao chegar no último exercício da série (imagem 8), uma opção de “finalizar treino” é exibida. Ao clicar nesse botão, o usuário atesta que concluiu todos os exercícios passados e uma comunicação com a base de dados é estabelecida, mudando o valor do campo fez_exercicio1 (nesse caso) para 1, indicando que o usuário concluiu o treino 1 (alongamento). O usuário será redirecionado para a página principal (imagem 9) onde, agora, ele verá a opção de **treino de alongamento** escurecida e, ao passar o mouse sobre a imagem do treino, receberá uma mensagem dizendo que ele já fez aquele treino, mas que não o impede de realiza-lo novamente; e a barra de progresso avançada em 20%.

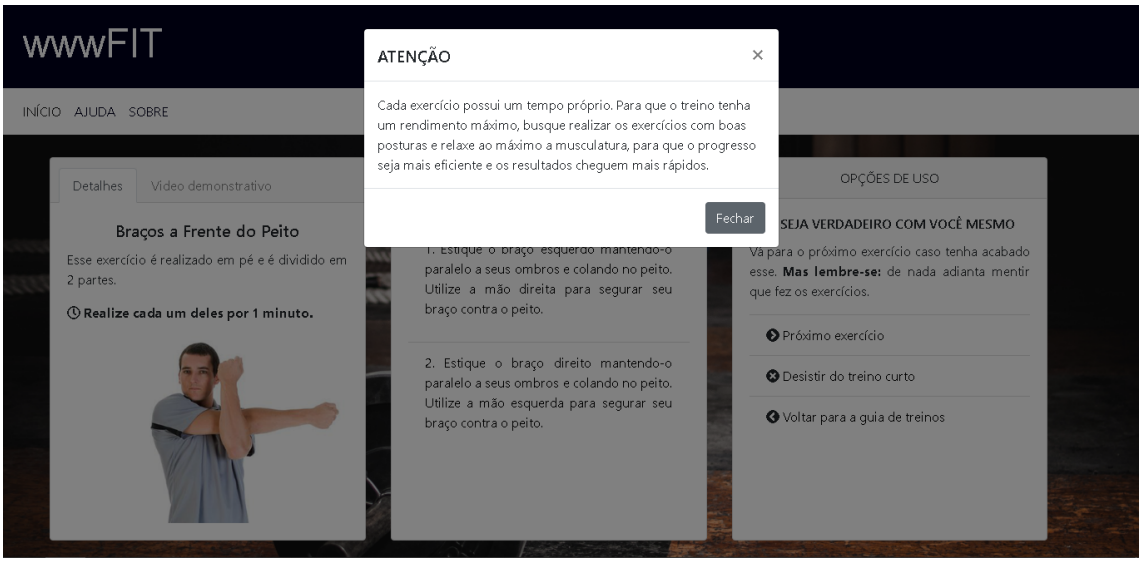


Imagem 6: aviso mostrado ao usuário na primeira interação com a página

Imagem 7: página do primeiro exercício

Imagem 8: último exercício da série o botão de finalizar treino

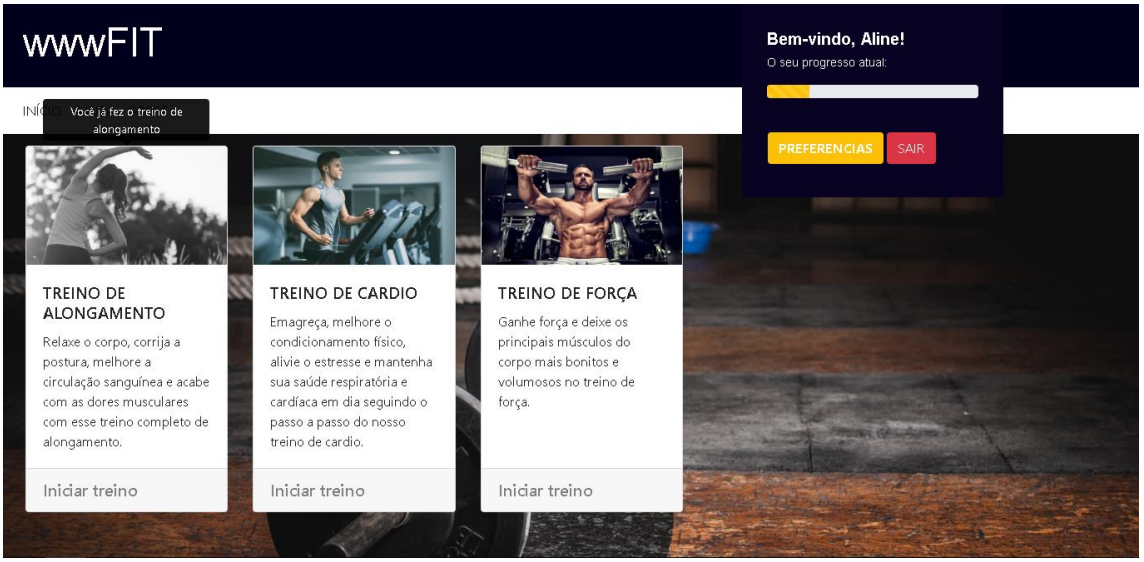


Imagem 9: a página agora mostra o progresso feito pelo usuário, bem como indica que ele já fez o treino de alongamento.

Completando o que foi mostrado no protótipo, porém não foi desenvolvido, o treino de força se mostrou muito grande e o grupo optou por apenas mostrar como ele seria feito e, das 4 opções mostradas no protótipo em **5. Funcionalidades a serem desenvolvidas**, apenas a de Corpo Inteiro foi implementada, a fim de mostrar que a conexão com o banco de dados funciona e é possível obter 100% de progresso no nosso aplicativo.

8. Execução do programa

Para a execução do aplicativo, definimos um roteiro a ser seguido caso o usuário utilize o sistema operacional Windows:

1. Baixe o **xampp**;
2. Faça as configurações necessárias para executar o apache server e o MySQL;
3. Com tudo pronto, inicie o apache e o MySQL e abra o phpMyAdmin (localhost:8080/phpMyAdmin), clique em SQL e insira as seguintes linhas de comandos:

```
CREATE DATABASE `aplicativo` /*!40100 DEFAULT CHARACTER SET utf8 */;

DROP TABLE IF EXISTS `role`;
CREATE TABLE  `role` (
  `role_id` int(11) NOT NULL auto_increment,
  `role` varchar(255) default NULL,
  PRIMARY KEY  (`role_id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `user`;
CREATE TABLE  `user` (
  `id` int(11) NOT NULL auto_increment,
  `firstname` varchar(255) NOT NULL,
  `lastname` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `active` int(11) default NULL,
  `fez_exercicio1` int(11) default 0,
  `fez_exercicio2` int(11) default 0,
  `fez_exercicio3` int(11) default 0,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `user_role`;
CREATE TABLE  `user_role` (
  `user_id` int(11) NOT NULL,
  `role_id` int(11) NOT NULL,
  PRIMARY KEY  (`user_id`,`role_id`),
  KEY `user_role_key` (`role_id`),
  CONSTRAINT `user_userrole` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`),
  CONSTRAINT `role_userrole` FOREIGN KEY (`role_id`) REFERENCES `role` (`role_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `role` VALUES (1,'ADMIN');
```

4. Na sua IDE (de preferência Eclipse ou Visual Studio Code – que foram utilizados para o desenvolvimento desse projeto) baixe as extensões necessárias para utilizar o Spring;
5. Importe o projeto para a sua IDE;
6. Rode a aplicação:
    a. No Eclipse, clique sobre a pasta do projeto e clique Run As > Spring Boot App
    b. No Visual Studio Code, em Spring Boot Dashboard, deve aparecer o nome da aplicação. Clique sobre ela e clique em “Start”

Concluindo este relatório, gostaríamos de enfatizar que outras páginas foram feitas (ajuda, sobre, registro e etc) porém não foram demonstradas nesse relatório, já que, a exemplo das páginas de ajuda e sobre, não surtem efeito no projeto e apenas foram feitas para contemplar a obra; e, a exemplo da página de registro, fica claro o propósito dela, o que dispensa apresentações.