

Rede Social – Fitnets

Manual Técnico do Sistema

V. 1.0

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.

Curso Tecnologia em Análise e Desenvolvimento de Sistemas

A6PGP – Prática de Gerenciamento de Projetos

Professores: Rafael e Ivan Martinez.

Alunos:

Pront.:

Rafael Soares Silva

0970883

Jean Carmona

0965294

Moacir Moda Neto

0961868

Gustavo Alves Gaspar

0965499

Luan Augusto Silveira da Costa

0966134

Sumário

1 - Introdução	4
2 – Contexto	4
3 – Considerações Iniciais	4
4 – Conceitos e Definições	5
4.1 – Django Framework	5
4.2 – Scrum	5
4.3 – Bootstrap	5
5 – UML	6
5.1 – Casos de Uso e Diagramas de Casos de Uso	6
Criar uma conta no Fitnets	7
Permitir entrada (login) do usuário no sistema	8
Recuperar senha do usuário no sistema	8
Encerrar a participação do usuário	9
Trocar e-mail / senha do usuário	9
Ativar uma conta desativada	10
Customizar mural pessoal	10
Selecionar privacidade das informações	11
Aceitar e recusar convites de amizade	11
Remover e bloquear amizades	12
Enviar convites de amizade	12
Comentar mural de amigos	12
Apagar um comentário de seu mural	13
Postar treino da academia.	13
Gerenciar treinos	13
Alterar treino	14
Votar em um treino	14
5.2 – Diagrama de Atividade	15
5.3 – Diagrama de Classe	19
5.4 – Diagrama de Sequência	20
6 – MER	22
Referências Bibliográficas	23

Índice de figuras

Figura 1..... 7

Figura 2..... 7

Figura 3..... 9

Figura 4..... 10

Figura 5..... 13

Figura 6..... 15

Figura 7..... 17

Figura 8..... 17

Figura 9..... 18

Figura 10..... 19

Figura 11..... 20

Figura 12..... 21

Figura 13..... 22

1 - Introdução

Este manual tem por objetivo a definição dos critérios e especificações do sistema fundamentais para o entendimento da solução, bem como o funcionamento do sistema Fitnets.

2 – Contexto

O sistema Fitnets tem como objetivo criar uma rede social para pessoas que praticam atividades físicas em academias, com ou sem ajuda de “personal trainer” e desejam compartilhar seus treinos e alimentação. Criando assim um ambiente especializado para essas pessoas oferecerem e receberem informações.

3 – Considerações Iniciais

O sistema Fitnets foi criado com o intuito primário de atender aos requisitos de software e técnicas da disciplina A6PGP, quaisquer necessidades comerciais ou interesses que possam surgir sobre o software futuramente podem requerer análises mais detalhadas sobre alguns processos para incorporar o ambiente da nuvem internet.

4 – Conceitos e Definições

Esse projeto foi desenvolvido utilizando-se do framework Django que utiliza como linguagem o Python. Como metodologia de controle do projeto, foi usado o Scrum ou uma simulação simples do mesmo, ou seja, optou-se por uma metodologia Ágil de desenvolvimento.

4.1 – Django Framework

O Django é um framework para alta produtividade, entre as principais “features” estão:

- ORM (object-relational mapper) embutido;
- Padrão MVC (model-view-controller);
- Internacionalização;
- Caching automático para alto desempenho;
- Utilização de templates para separação de código;
- Design elegante de URL (Uniform Resource Locator).

Maiores informações e o download do Django podem ser obtidos em:

<https://www.djangoproject.com/>

4.2 – Scrum

Scrum é uma metodologia de controle de projetos de software que integra a família agile de métodos.

No Scrum existem as entidades básicas de controle do projeto:

- Scrum master: o analista / gerente do projeto;
- Proprietário do produto: O “stakeholder”;
- A equipe: os desenvolvedores.

No Scrum existe um backlog, que é aonde se colocam os tickets para a equipe de desenvolvimento. O analista cria ou aprimora esses tickets e envia para os desenvolvedores fazerem o Sprint.

Sprint é a unidade básica de desenvolvimento do projeto, cada Sprint normalmente corresponde a uma funcionalidade do sistema quando se utiliza o FDD (Feature Driven Development).

Todos os ticket estão registrados no backlog do GitHub do projeto, podendo ser vistos no TCC documento original.

Nesse projeto foi utilizado uma versão simplificada do Scrum, que é altamente versátil e adaptável as necessidades de cada equipe.

4.3 – Bootstrap

É um projeto opensource criado pelos engenheiros do Twitter para criar interfaces gráficas amigáveis e rápidas. Utiliza javascript e CSS. É rápido e robusto para criar sistemas WEB.

Download está disponível em: <http://twitter.github.com/bootstrap/index.html>

5 – UML

UML (Unified Modeling Language) é a ferramenta padrão adotada pelos analistas de sistemas para documentar e modelar sistemas orientados a objetos, substituindo e algumas vezes somando a antiga abordagem de análise estruturada.

Nesse manual foi usado apenas a modelagem orientada a objeto, por ser melhor adaptada a sistemas com tecnologia para WEB.

Segundo Russ Miles e Kim Hamilton, autores do livro Learning UML 2.0,

“In systems design, you model for one important reason: to manage complexity. Modeling helps you see the forest for the trees, allowing you to focus on, capture, document, and communicate the important aspects of your system’s design.”

“Nos projetos de sistemas, você modela por uma razão importante: gerenciar a complexidade. Modelar ajuda a enxergar a floresta pelas árvores, permitindo a você focar, capturar, documentar e comunicar a importância dos aspectos do projeto do sistema.”

Partindo desse princípio podemos perceber a importância da documentação e modelagem do sistema, a compreensão do sistema sem a necessidade da leitura e estudo de seu código fonte, permitindo aos analistas descobrir erros e modificar funcionalidades antes mesmo de começarem o desenvolvimento.

A linguagem UML define diversos diagramas, cada um com seu respectivo valor, funcionalidade e objetivo definido. Nesse manual foram desenvolvidos os seguintes diagramas: Casos de Uso e diagramas, Atividades, Classe e Sequência.

Os demais diagramas foram desconsiderados visando a não ostentação da complexidade e a falta de objetivo em desenvolvê-los, pois suas existências não acresceriam em nada para a compreensão do sistema.

5.1 – Casos de Uso e Diagramas de Casos de Uso

Na definição de caso de uso, ainda Russ Miles e Kim Hamilton, autores do livro Learning UML 2.0,

“A use case is a case (or situation) where your system is used to fulfill one or more of your user’s requirements; a use case captures a piece of functionality that the system provides. Use cases are at the heart of your model.”

“Um caso de uso é um caso (ou situação) aonde seu sistema é usado para preencher um ou mais requerimentos do seu usuário; um caso de uso captura a funcionalidade daquilo que o sistema provê. Casos de uso estão no coração do seu modelo.”

Os casos de uso são descritos em formas de ficha com a descrição do requisito e sua análise, abaixo os casos de uso do sistema.

Diagrama geral do Fitnets apresentado na figura 1.



Figura 1

Na figura 2, o diagrama do processo de criar login, fazer login e recuperar senha.

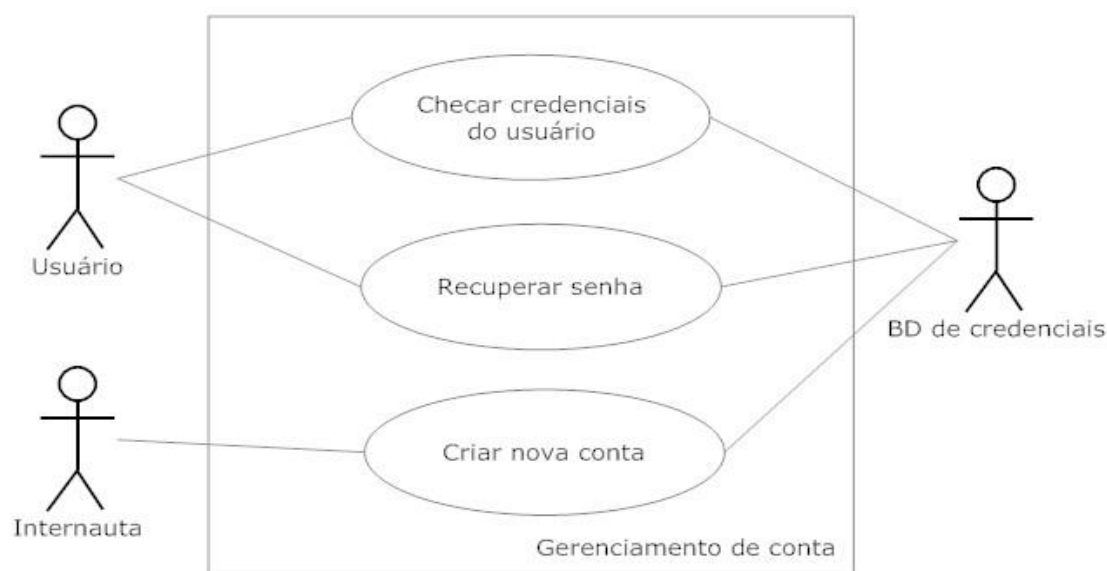


Figura 2

Criar uma conta no Fitnets

Caso de uso relacionado: Requerimento A.1

Objetivos no contexto: Um internauta deseja ingressar no sistema Fitnets e para isso precisa criar uma nova conta, caso não possua uma.

Condições prévias: O usuário precisa possuir seus dados pessoais para preencher o cadastro e um endereço de e-mail válido.

Condição de sucesso: Uma nova conta de usuário é criada.

Condição de fracasso: A conta do novo usuário é rejeitada ou não pode ser finalizada por falta de informações.

Atores primários: Internauta sem conta no Fitnets.

Atores secundários: Banco de dados de credenciais.

Gatilho: O internauta seleciona a opção de cadastro de novo usuário no sistema Fitnets.

Etapas:

1. O internauta seleciona a opção cadastro de novo usuário;
2. O internauta preenche um formulário informando dados pessoais, e-mail e “nickname”;
3. O sistema verifica se o e-mail ou “nickname” já estão em uso;
4. O internauta é cadastrado no sistema e é enviado um e-mail de confirmação para ativar sua conta;
5. O internauta confirma a conta pelo link do e-mail e está apto a ingressar no sistema.
6. O sistema salva a conta do usuário no BD.

Extensões:

- 3.1 O “nickname” ou e-mail inserido já estão em uso;
- 3.2 O formulário é rejeitado.
- 4.1 O e-mail não chega à caixa de e-mail do internauta;
- 4.2 O cadastro não é ativado e expira após um período de tempo.

Permitir entrada (login) do usuário no sistema

Caso de uso relacionado: Requerimento A.2

Objetivos no contexto: Um usuário já cadastrado deseja entrar no sistema.

Condições prévias: O usuário precisa ter uma conta válida no sistema.

Condição de sucesso: O usuário entra no sistema.

Condição de fracasso: O usuário não consegue entrar no sistema.

Atores primários: Usuário do Fitnets.

Atores secundários: Banco de dados de credenciais.

Gatilho: O usuário digita e-mail e senha e seleciona entrar.

Etapas:

1. O usuário seleciona a opção cadastro de novo usuário;
2. O sistema procura no banco de dados de credenciais pelo usuário com e-mail e senha digitados;
3. O sistema encontra o registro do usuário;
4. O sistema permite a entrada do usuário no sistema e inicia a sessão;
5. O sistema encaminha o usuário para a sua página pessoal.

Extensões:

- 3.1 O sistema não encontra o usuário no banco de dados de credenciais;
- 3.2 A entrada no sistema é rejeitada.

Recuperar senha do usuário no sistema

Caso de uso relacionado: Requerimento A.3

Objetivos no contexto: Um usuário já cadastrado esqueceu a senha..

Condições prévias: O usuário precisa ter uma conta válida no sistema.

Condição de sucesso: O usuário troca a senha antiga.

Condição de fracasso: O usuário não consegue trocar a senha.

Atores primários: Usuário do Fitnets.

Atores secundários: Banco de dados de credenciais.

Gatilho: O usuário seleciona resetar senha.

Etapas:

1. O usuário seleciona a opção de resetar sua senha;
2. O usuário informa o e-mail e nickname;
3. O sistema encontra o registro do usuário cria uma senha temporária e envia para o usuário;
4. O usuário entra no sistema com a senha nova e troca a senha.

Extensões:

- 3.1 O sistema não encontra a combinação e-mail/“nickname” no banco de dados de credenciais;

3.2 O reset da senha é rejeitado

Na figura 3, o gerenciamento de conta do usuário é demonstrado.

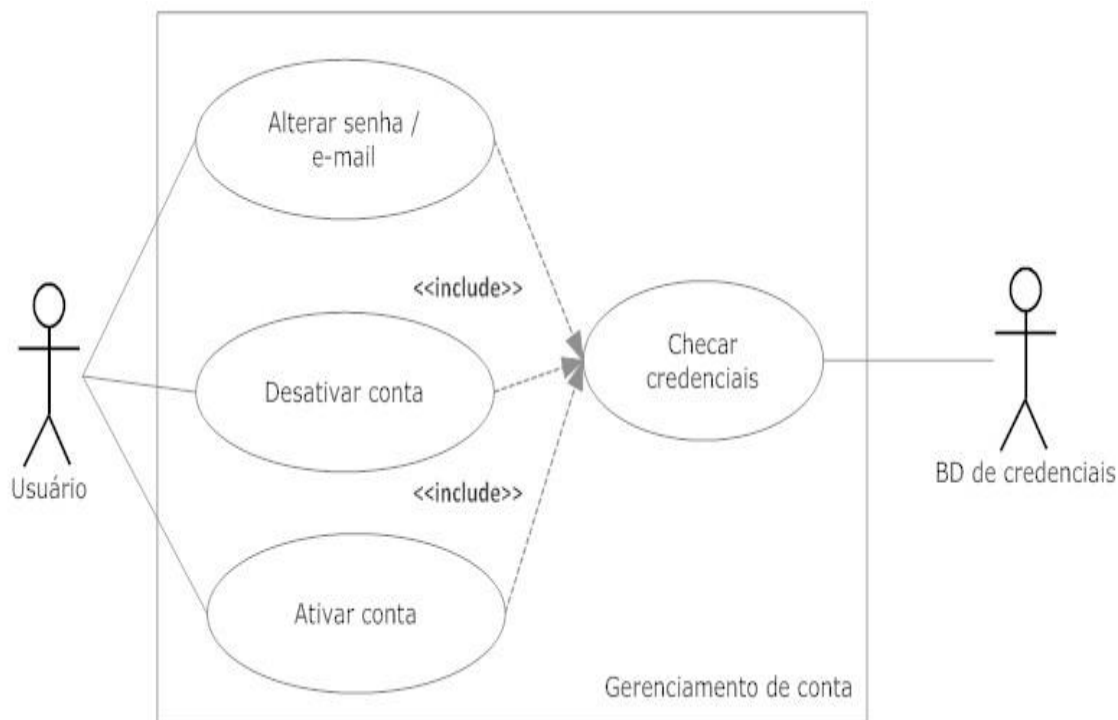


Figura 3

Encerrar a participação do usuário

Caso de uso relacionado: Requerimento A.4

Objetivos no contexto: Um usuário já cadastrado deseja encerrar sua participação no sistema.

Condições prévias: O usuário precisa ter uma conta válida no sistema e estar logado no sistema.

Condição de sucesso: O usuário encerra sua participação.

Atores primários: Usuário do Fitnets.

Atores secundários: Banco de dados de credenciais.

Gatilho: O usuário seleciona encerrar conta.

Etapas:

1. O usuário seleciona a opção de encerrar sua conta;
2. O usuário informa o motivo;
3. O sistema encontra o registro do usuário e modifica o estado da conta para inativo;
4. O usuário desconectado do sistema e encaminhado para a página principal.

Trocar e-mail / senha do usuário

Caso de uso relacionado: Requerimento A.5

Objetivos no contexto: Um usuário já cadastrado deseja trocar sua senha e/ou e-mail.

Condições prévias: O usuário precisa ter uma conta válida no sistema e estar logado no sistema.

Condição de sucesso: O usuário troca sua senha / e-mail.

Atores primários: Usuário do Fitnets.

Atores secundários: Banco de dados de credenciais.

Gatilho: O usuário seleciona modificar dados da conta.

Etapas:

1. O usuário seleciona a opção modificar dados da conta;
2. O usuário informa o novo e-mail e/ou senha;
3. O sistema modifica as informações no banco de dados de credenciais;
4. O sistema envia um e-mail para o usuário notificando as mudanças.

Ativar uma conta desativada

Caso de uso relacionado: Requerimento A.6

Objetivos no contexto: Um usuário já cadastrado deseja ativar sua conta inativa.

Condições prévias: O usuário precisa ter uma conta válida no sistema e esta estar inativa.

Condição de sucesso: O usuário ativa sua conta.

Condição de fracasso: O sistema não permite a ativação da conta.

Atores primários: Ex-usuário do Fitnets.

Atores secundários: Banco de dados de credenciais.

Gatilho: O ex-usuário faz login no sistema com as antigas credenciais.

Etapas:

1. O ex-usuário tenta o login com as antigas credenciais;
2. O sistema procura a conta no banco de dados de credenciais;
3. O sistema encontra a conta e pergunta se o ex-usuário deseja ativar sua conta;
4. O sistema envia um e-mail para o usuário notificando a ativação e o usuário entra no sistema.

Extensões:

- 3.1 O sistema não encontra a conta do ex-usuário;
- 3.2 A ativação é rejeitada.

O controle do mural é descrito na figura 4.

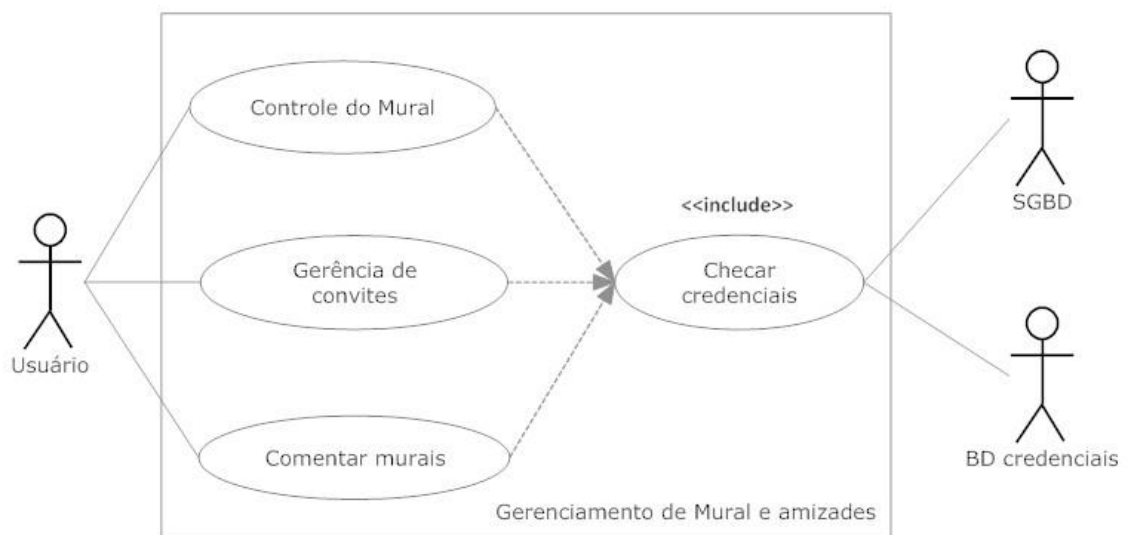


Figura 4

Customizar mural pessoal

Caso de uso relacionado: Requerimento B.1

Objetivos no contexto: Um usuário do sistema quer customizar seu mural.

Condições prévias: O usuário precisa estar logado no sistema.

Condição de sucesso: O usuário customiza o mural.

Condição de fracasso: O usuário não customiza o mural.

Atores primários: Usuário do Fitnets.

Atores secundários: SGBD.

Gatilho: O usuário seleciona a opção customizar.

Etapas:

1. O usuário seleciona a opção customizar;
2. O sistema redireciona o usuário para um formulário com os campos;
3. O usuário preenche os campos que deseja customizar;
4. O usuário confirma e é redirecionado para seu mural com as informações customizadas.
5. O sistema atualiza as informações no SGBD.

Extensões:

- 4.1 O sistema não aceita o(s) valor(es) de um ou mais campos por conterem informações inválidas ou mal formatadas.

Selecionar privacidade das informações

Caso de uso relacionado: Requerimento B.2

Objetivos no contexto: Um usuário do sistema pode selecionar o que é público e o que é apenas para amigos.

Condições prévias: O usuário precisa estar logado no sistema.

Condição de sucesso: O usuário seleciona o que quer que apareça na página.

Atores primários: Usuário do Fitnets.

Atores secundários: SGBD.

Gatilho: O usuário seleciona as informações públicas.

Etapas:

1. O usuário seleciona a opção customizar;
2. O sistema redireciona o usuário para um formulário com os campos;
3. O usuário seleciona os campos que deseja mostrar ao público;
4. O usuário confirma e é redirecionado para seu mural com as informações atualizadas.
5. O sistema atualiza as informações no SGBD.

Aceitar e recusar convites de amizade

Caso de uso relacionado: Requerimento B.3

Objetivos no contexto: Um usuário do sistema pode aceitar ou recusar um convite de amizade de outro usuário do sistema.

Condições prévias: O usuário precisa estar logado no sistema.

Condição de sucesso: O usuário seleciona os convites que quer aceitar e recusar.

Condição de fracasso: Usuário do convite não é encontrado.

Atores primários: Usuário do Fitnets.

Atores secundários: SGBD.

Gatilho: O usuário entra no seu mural e verifica a lista de convites pendentes.

Etapas:

1. O usuário entra no seu mural;
2. O sistema lista todos os convites pendentes do usuário;
3. O usuário seleciona aceitar ou recusar para o convite;
4. O sistema SGBD altera as informações para refletir os novos relacionamentos de amizade e apaga os convites já respondidos.

Extensões:

- 4.1 O sistema não encontra as contas de usuário (contas foram desativadas);
- 4.2 Os pedidos de amizade são apagados.

Remover e bloquear amizades

Caso de uso relacionado: Requerimento B.4

Objetivos no contexto: Um usuário do sistema pode remover uma pessoa e bloqueá-la no sistema.

Condições prévias: O usuário precisa estar logado no sistema.

Condição de sucesso: O usuário seleciona a amizade no painel de amizades que deseja remover e bloquear.

Atores primários: Usuário do Fitnets.

Atores secundários: SGBD.

Gatilho: O usuário entra no seu mural acessa o painel de amizades e remove um amigo.

Etapas:

1. O usuário entra no seu mural;
2. O sistema lista todas as amizades do usuário;
3. O usuário seleciona a amizade a ser removida e confirma se deseja bloquear também;
4. O sistema SGBD apaga a amizade e caso seja selecionado bloquear ele inclui o usuário da amizade em uma tabela de bloqueio.

Enviar convites de amizade

Caso de uso relacionado: Requerimento B.5

Objetivos no contexto: Um usuário do sistema quer enviar um convite de amizade para outro usuário do sistema.

Condições prévias: O usuário precisa estar logado no sistema.

Condição de sucesso: O usuário visita o mural alheio e aperta em solicitar amizade.

Condição de fracasso: O convite não é aceito.

Atores primários: Usuário do Fitnets.

Atores secundários: SGBD.

Gatilho: O usuário acessa mural alheio e seleciona solicitar convite.

Etapas:

1. O usuário visita o mural de outro usuário;
2. O usuário seleciona a opção solicitar amizade;
3. O sistema atualiza o SGBD para incluir o novo pedido de amizade;
4. O sistema informa o novo usuário do pedido de amizade assim que ele for aceito.

Extensões:

- 4.1 O sistema não informa o pedido aceito, pois o mesmo não foi respondido ou foi recusado.

Comentar mural de amigos

Caso de uso relacionado: Requerimento B.6

Objetivos no contexto: Um usuário do sistema quer comentar no mural de um de seus amigos.

Condições prévias: O usuário precisa estar logado no sistema.

Condição de sucesso: O usuário visita o mural do amigo e posta um comentário.

Atores primários: Usuário do Fitnets.

Atores secundários: SGBD.

Gatilho: O usuário acessa mural do amigo, digita um comentário e seleciona comentar.

Etapas:

1. O usuário visita o mural do amigo;
2. O usuário digita um texto na área reservada ao comentário e confirma;
3. O sistema atualiza o SGBD para incluir o novo comentário;
4. O sistema mostra o novo comentário no mural do amigo.

Apagar um comentário de seu mural

Caso de uso relacionado: Requerimento B.7

Objetivos no contexto: Um usuário do sistema quer apagar um comentário em seu mural.

Condições prévias: O usuário precisa estar logado no sistema.

Condição de sucesso: O usuário apaga o comentário do mural.

Atores primários: Usuário do Fitnets.

Atores secundários: SGBD.

Gatilho: O usuário acessa seu mural e seleciona o comentário a ser apagado.

Etapas:

1. O usuário visita seu mural;
2. O usuário escolhe a(s) mensagem(ns) para serem apagadas e confirma;
3. O sistema atualiza o SGBD para excluir o(s) comentário(s);
4. O sistema mostra o mural atualizado.

Figura 5, o gerenciamento de treinos.

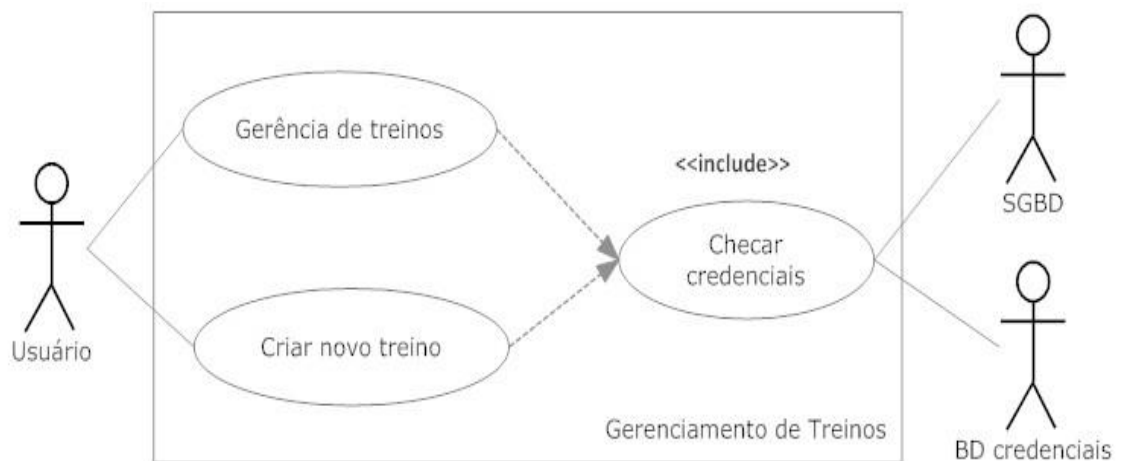


Figura 5

Postar treino da academia.

Caso de uso relacionado: Requerimento C.1

Objetivos no contexto: Um usuário do sistema quer postar um treino da academia.

Condições prévias: O usuário precisa estar logado no sistema.

Condição de sucesso: O usuário posta o treino no mural.

Atores primários: Usuário do Fitnets.

Atores secundários: SGBD.

Gatilho: O usuário acessa seu mural e seleciona postar novo treino.

Etapas:

1. O usuário visita seu mural;
2. O usuário escolhe postar um novo treino;
3. Usuário digita o treino nos respectivos dias e horários e seleciona postar;
4. O sistema atualiza o SGBD e mostra o treino no mural do usuário.

Gerenciar treinos.

Caso de uso relacionado: Requerimento C.2

Objetivos no contexto: O sistema deve gerenciar os treinos do usuário.

Condições prévias: O usuário precisa ter treinos armazenados no SGBD.

Condição de sucesso: O usuário escolhe qual treino é o ativo.

Condição de fracasso: Não existem treinos suficientes no SGBD.

Atores primários: Sistema Fitnets.

Atores secundários: Usuário e SGBD.

Gatilho: O usuário acessa seu mural e seleciona um treino para ser o ativo, tornando os outros treinos, automaticamente, como arquivados.

Etapas:

1. O usuário visita seu mural e seleciona um treino do arquivo;
2. O usuário seleciona o status ativo para o treino;
3. O sistema salva o status no SGBD e atualiza os outros treinos;
4. O sistema atualiza o mural e mostra o treino ativo no mural.

Extensões:

- 2.1 O sistema não encontra nenhum treino arquivado;
- 2.2 O sistema não pode trocar o treino ativo.

Alterar treino.

Caso de uso relacionado: Requerimento C.3

Objetivos no contexto: O sistema deve permitir alterar um treino.

Condições prévias: O usuário precisa ter o treino armazenado no SGBD.

Condição de sucesso: O usuário escolhe qual treino será alterado.

Atores primários: Sistema Fitnets.

Atores secundários: Usuário e SGBD.

Gatilho: O usuário acessa seu mural e seleciona um treino para ser alterado.

Etapas:

1. O usuário visita seu mural e seleciona um treino;
2. O usuário seleciona a opção modificar e altera as informações do treino;
3. O sistema salva o status no SGBD e atualiza os dados do treino;
4. O sistema atualiza o mural e mostra o treino ativo no mural.

Votar em um treino.

Caso de uso relacionado: Requerimento C.4

Objetivos no contexto: O sistema deve permitir aos usuários votarem em um treino.

Condições prévias: O usuário precisa ter o treino armazenado no SGBD.

Condição de sucesso: O usuário vota no treino se gostou ou não.

Atores primários: Sistema Fitnets.

Atores secundários: Usuário e SGBD.

Gatilho: O usuário acessa seu mural e seleciona se gostou ou não.

Etapas:

1. O usuário acessa seu mural e seleciona se gostou ou não;
2. O sistema computa o voto;
3. O sistema salva no SGBD;

5.2 – Diagrama de Atividade

Russ Miles e Kim Hamilton opinam sobre o diagrama de atividades, figura 6:

“Activity diagrams allow you to specify how your system will accomplish its goals. Activity diagrams show high-level actions chained together to represent a process occurring in your system. For example, you can use an activity diagram to model the steps involved with creating a blog account.”

“Diagramas de atividade permitem a você especificar como seu sistema irá alcançar seus objetivos. Diagramas de atividade mostram ações de alto nível juntas para representar um processo ocorrendo no seu sistema.”

Fitnets - Diagrama Atividade

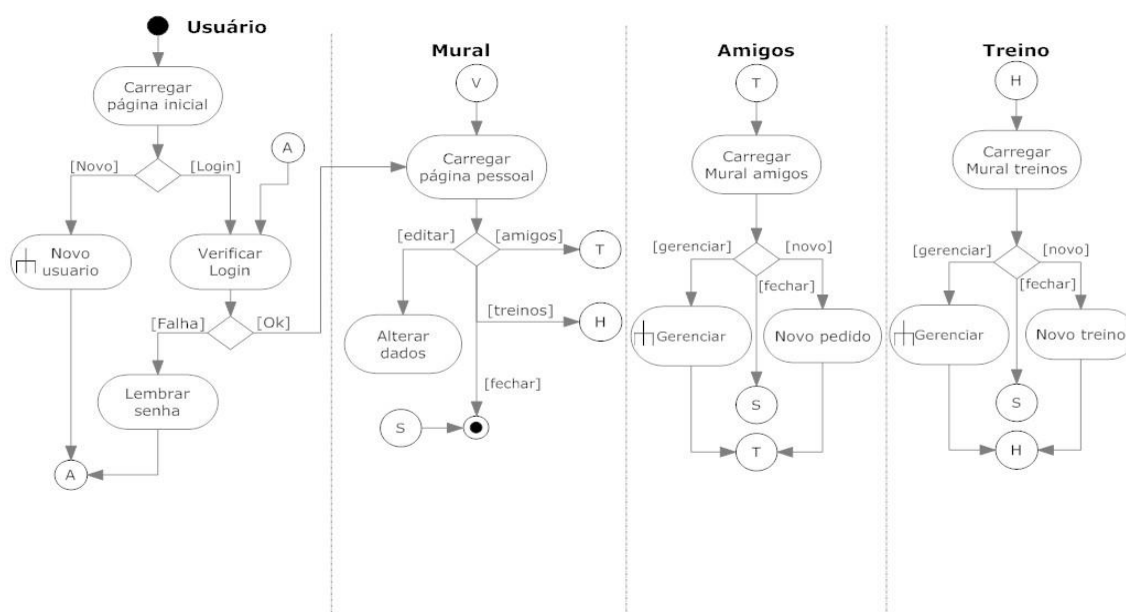


Figura 6

Diagrama expandido.

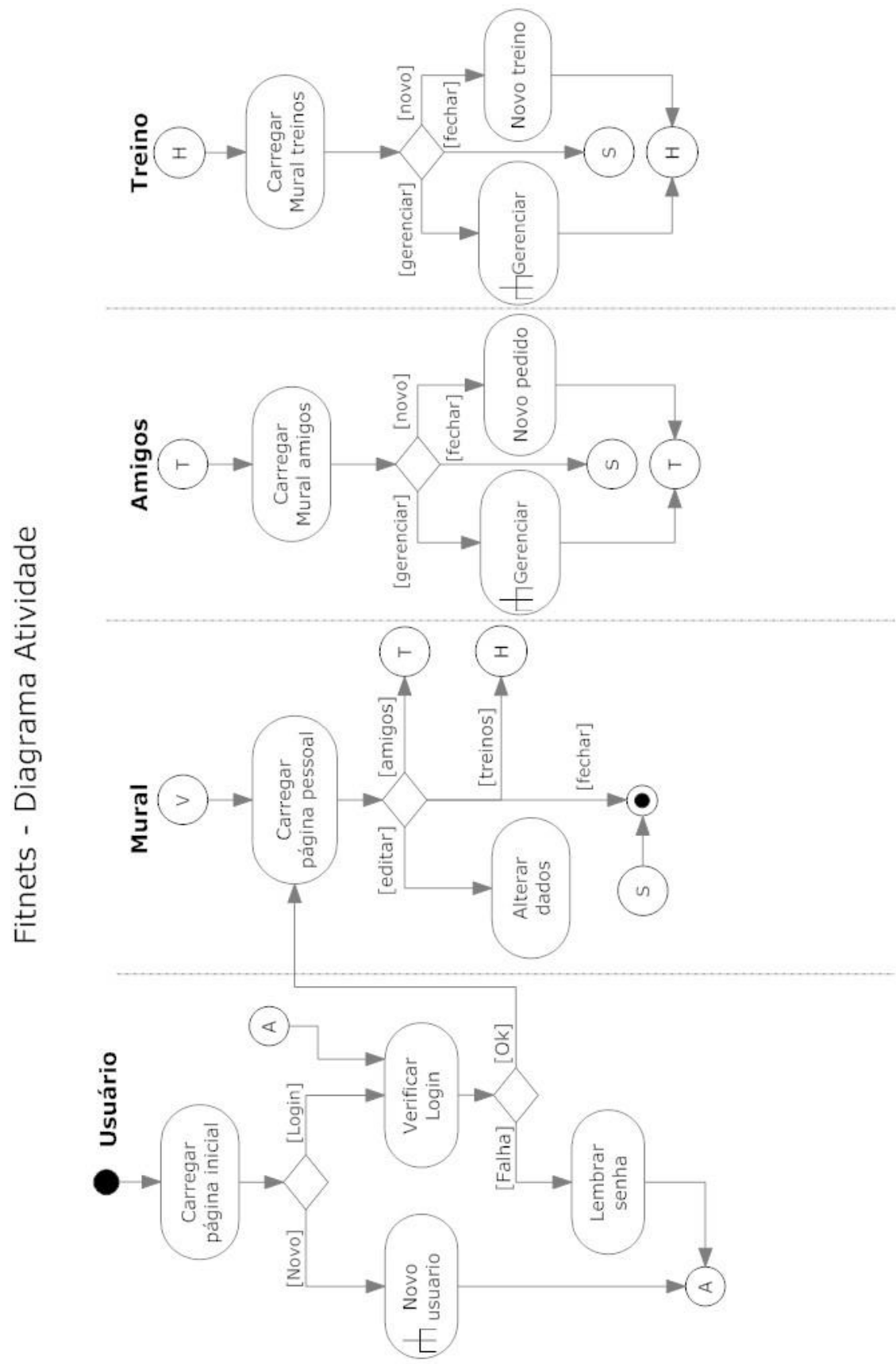


Diagrama criar novo usuário, figura 7.

Diagrama Atividade - Novo usuário

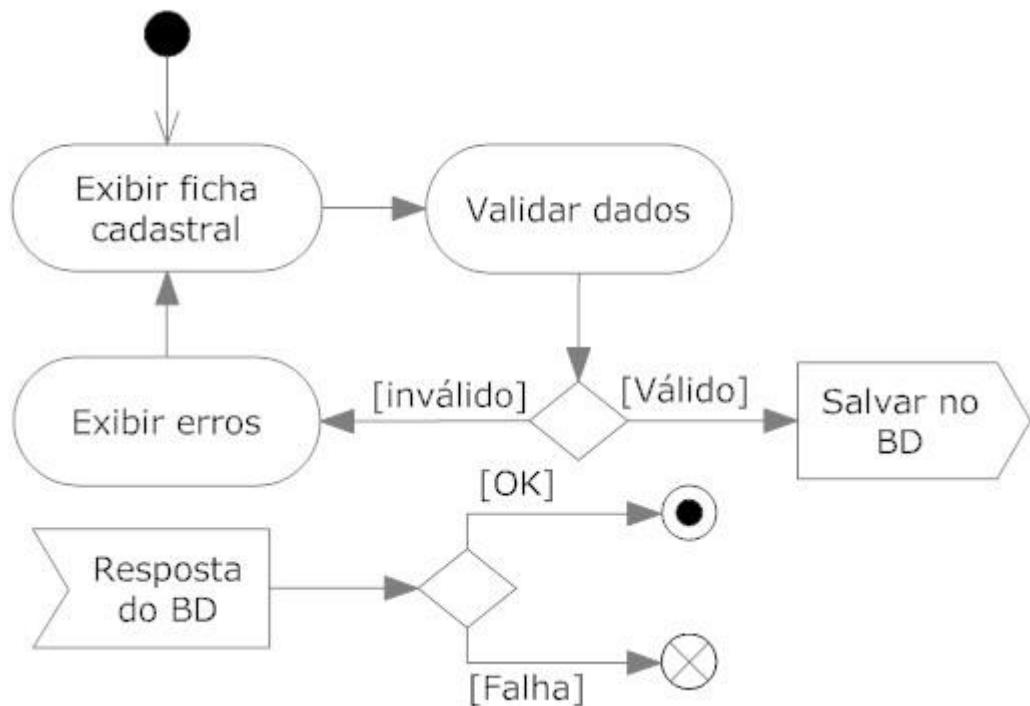


Figura 7

Diagrama gerenciar treinos e amizades, figuras 8 e 9 respectivamente.

Diagrama Atividade: Gerenciar Treinos

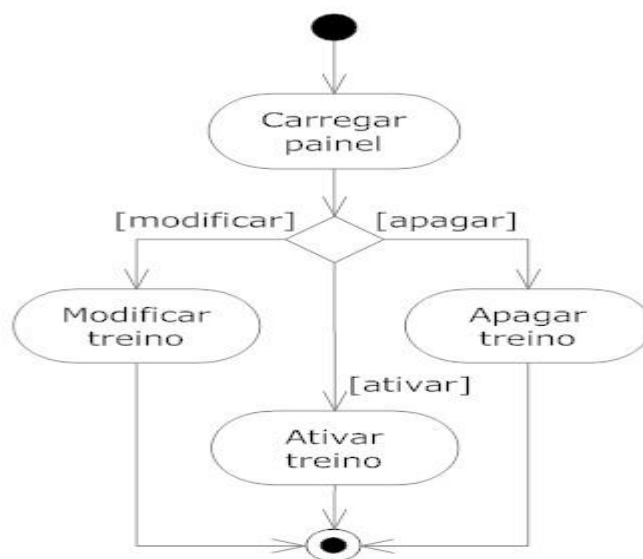


Figura 8

Diagrama Atividade: Gerenciar Amizade

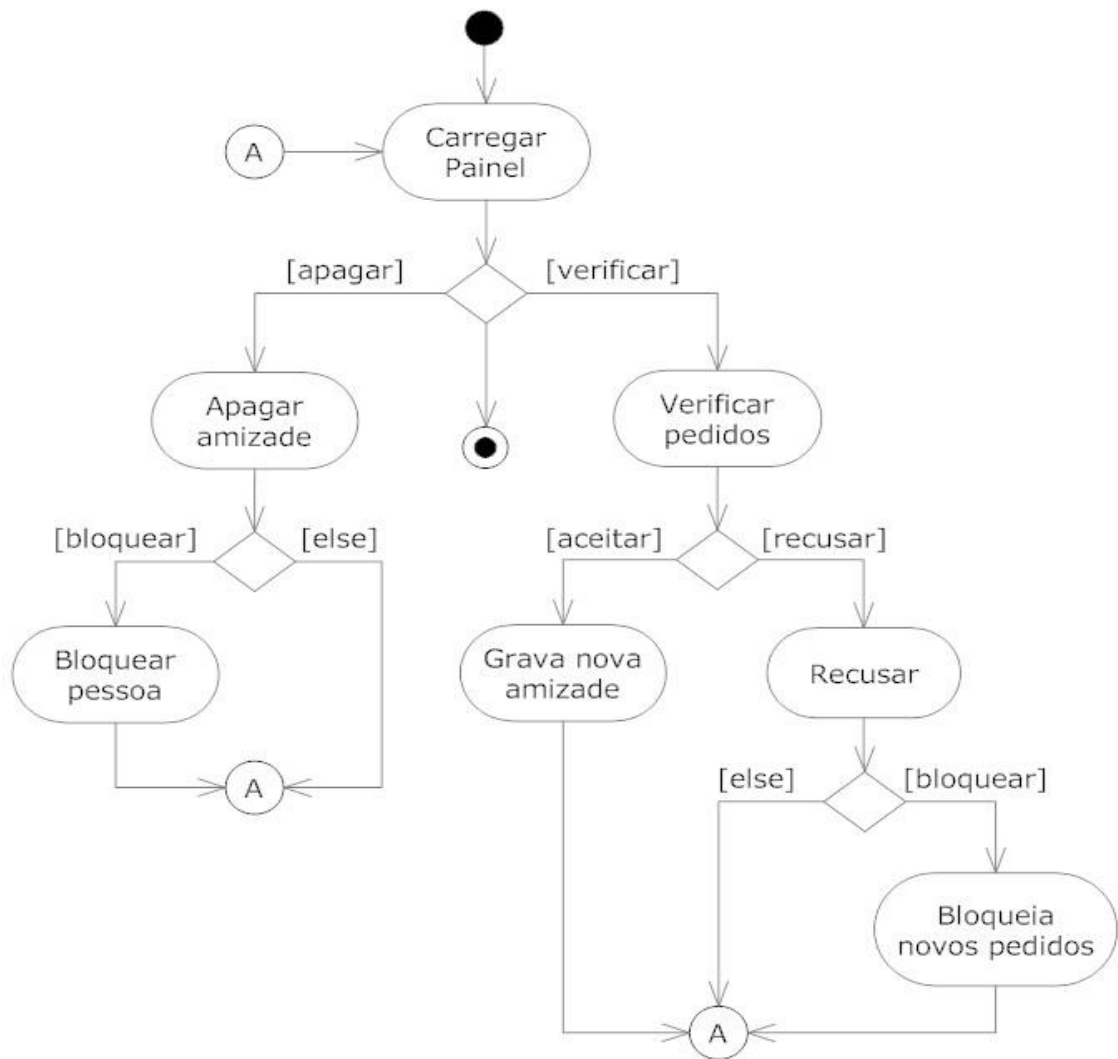


Figura 9

Existe uma observação a ser feita sobre o ponto de fim do diagrama, pois em se tratando de sistemas WEB o usuário nem sempre faz o “logout” do sistema, muitas vezes apenas fechando o browser ou navegando para outro site, por isso não houve o esforço de demonstrar todos os pontos de saída do sistema por serem virtualmente em qualquer lugar.

5.3 – Diagrama de Classe

Martin Fowler UML Distilled 3rd fala sobre diagramas de classe,

“A class diagram describes the types of objects in the System and the various kinds of static relationships that exist among them. Class diagrams also show the properties and operations of a Class and the constraints that apply to the way objects are connected. The UML uses the term feature as a general term that covers properties and operations of a Class.”

“Um diagrama de classe descreve os tipos de objetos no sistema e os vários tipos de relacionamentos estáticos que existem entre eles. Diagramas de classe também mostram as propriedades e operações de uma classe e as restrições que se aplicam ao modo que os objetos estão conectados.”

É uma prática comum em sistemas desenvolvidos para WEB abstrair as classes do sistema para tornarem-se apenas containers, sendo que o sistema utiliza a inteligência do banco de dados com os procedimentos armazenados para processar. Ou através do MVC (model-view-controller) e o uso de ORM (object-relational mapper) fazendo com que as classes do sistema apenas espelhem o desenho das tabelas do banco de dados.

Nesses casos, as classes normalmente não apresentam operações próprias, que é o caso do sistema Fitnets, que utiliza o framework Django baseado em MVC, a camada Model usa as classes apenas como “holders” da informação.

Figura 10 mostra o diagrama de classe simplificado.

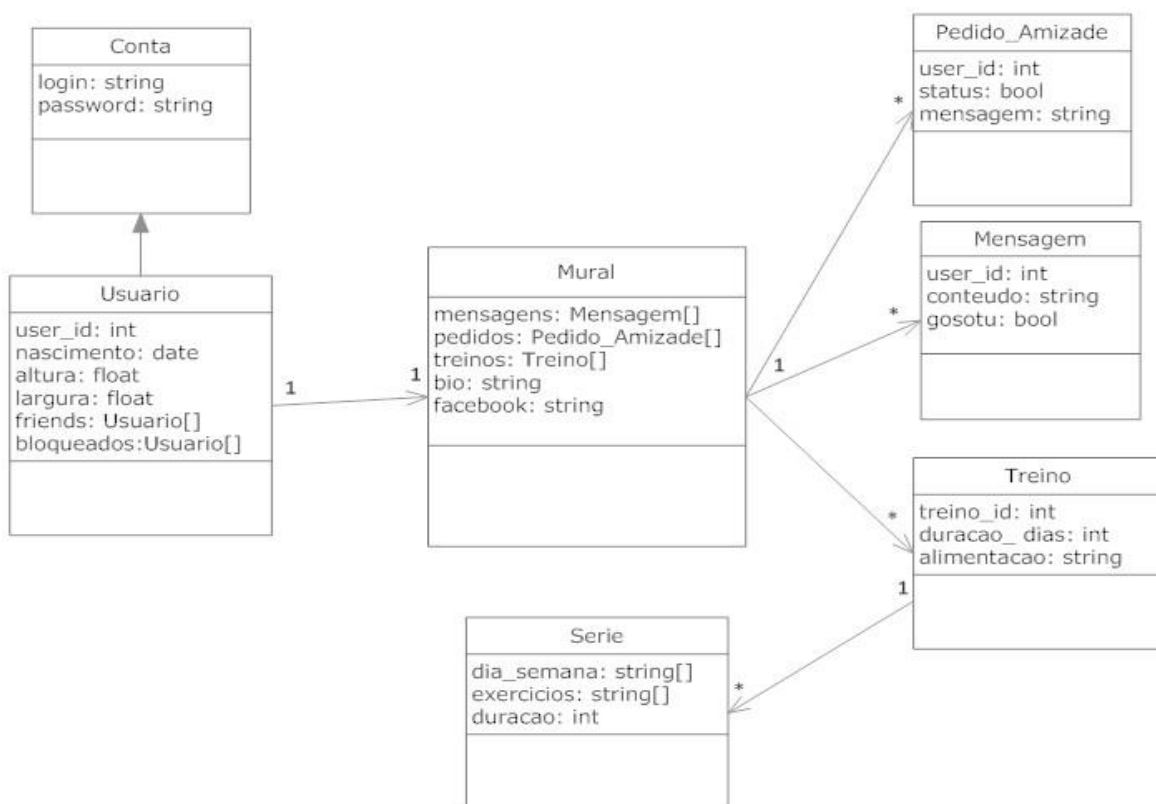


Figura 10

5.4 – Diagrama de Sequência

Consiste em um diagrama que tem o objetivo de mostrar como as mensagens entre os objetos são trocadas no decorrer do *tempo* para a realização de uma operação.

Martin Fowler UML Distilled 3º ed define,

“Interaction diagrams describe how groups of objects collaborate in some behavior. The UML defines several forms of interaction diagram, of which the most common is the sequence diagram .”

“Diagramas interativos descrevem como grupos de objetos colaboram em algum comportamento. O UML define várias formas de diagramas interativos, dos quais o mais comum é o diagrama de sequência.”

A figura 11 mostra o diagrama que inicia o sistema, o “login” do usuário.

Diagrama de Sequência: Login

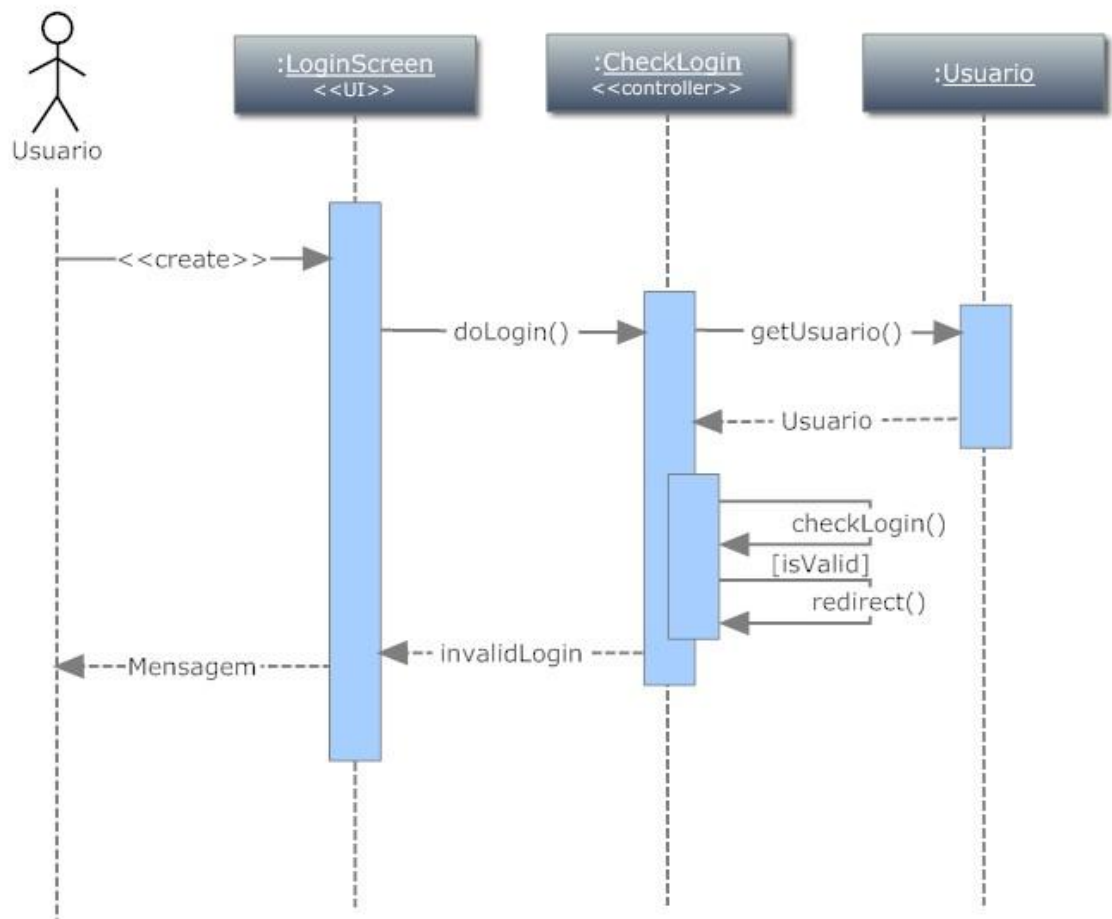


Figura 11

Um usuário do sistema só poderá adentrar o sistema Fitnets se estiver logado no mesmo, para isso será necessário fornecer um usuário e senha.

Por motivos de simplicidade, podem-se esconder os detalhes óbvios do diagrama como a passagem de usuário e senha para checagem, pois isso seria o óbvio a se fazer e poluiria o diagrama.

Ao receber o usuário da camada de modelo, é verificado que o objeto é válido, caso positivo é redirecionado para a próxima etapa, figura 12, carregar o Perfil (Mural) do usuário.

Diagrama de Sequência: Carregar Mural

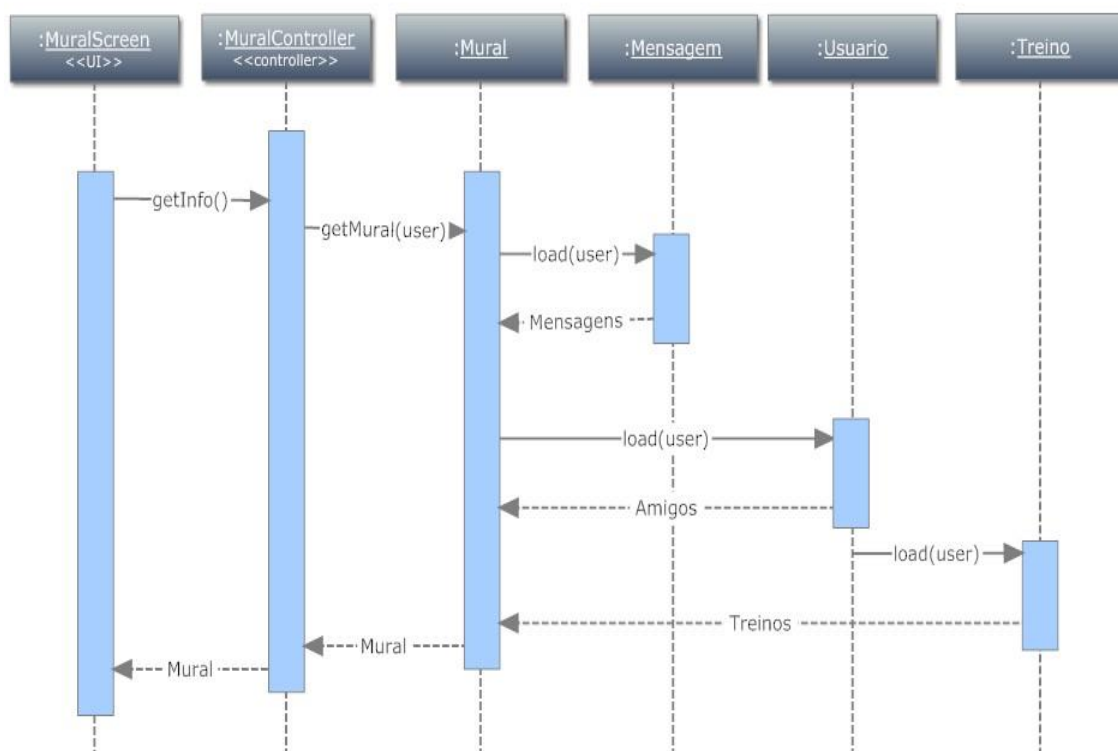


Figura 12

O diagrama demonstra bem o funcionamento do MVC, aonde o controller intermedia as necessidades da camada de interface com o usuário (VIEW) com o modelo (MODEL), bastando apenas o código do usuário para que se obtenham as informações necessárias do banco de dados.

Os demais diagramas foram omitidos, pois apenas são repetições do mesmo padrão para visualização de entidades específicas como: Amigos, Treinos e Mensagens.

6 – MER

O modelo de entidades de relacionamento é um diagrama que mostra como as entidades do sistema se relacionam umas com as outras, muitas vezes esse diagrama é substituído ou anexado ao DER, que seria uma evolução do MER por assim dizer mostrando as relações já com estrutura do banco de dados.

A figura 13 mostra as relações mais importantes do sistema.

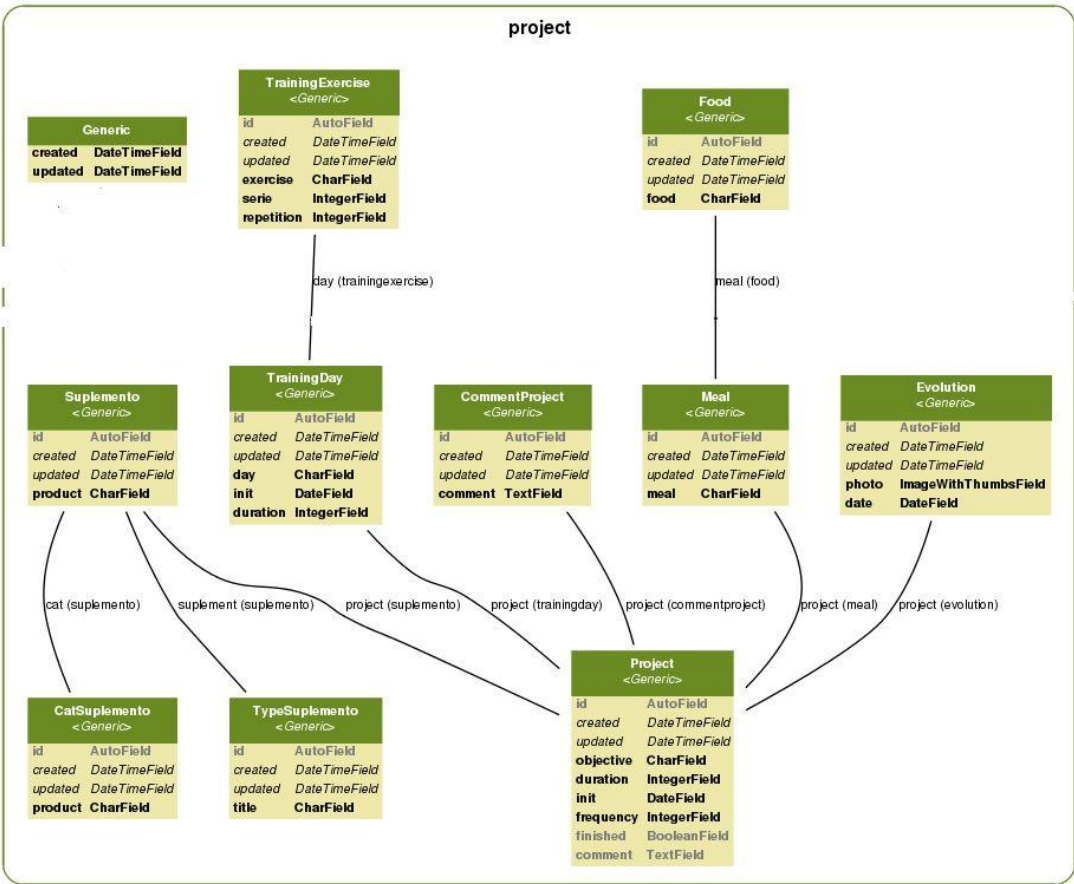


Figura 13

Referências Bibliográficas

Fowler, Martin;
UML Distilled Third Edition; Addison Wesley 2003.

Miles, Russ; Hamilton, Kim;
Learning UML 2.0; O'Reilly 2006.

Pilone, Dan; Pitman, Neil;
UML 2.0 in a Nutshell; O'Reilly 2005.

Rosenberg, D; Scott, K.;
Applying Use Case Driven Object Modeling with UML – An annotated e-Commerce Example;
Addison Wesley 2001.

Rosenberg, D; Scott, K.;
Applying Use Case Driven Object Modeling with UML – An annotated e-Commerce Example;
Addison Wesley 2001.

Per, Kroll; Bruce, MacIsaac;
Agility and Discipline Made Easy: Practices from OpenUP and RUP; Addison Wesley 2006

Guia de Orientação à Normalização de Trabalhos Acadêmicos. – São Paulo: IFSP, 2011.