**UNIVERSIDADE METODISTA DE SÃO PAULO**

**FACULDADE DE EXATAS E TECNOLOGIA**

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**MATHIAS NEVES TAVARES**

**PEDRO CERIONI TOGNATO**

**RAFAEL SOUZA DA SILVA**

**APLICATIVO ANDROID FIT CONTROL**

**DOCUMENTAÇÃO E DESENVOLVIMENTO DO APLICATIVO**

**SÃO BERNARDO DO CAMPO**

**2016**

**MATHIAS NEVES TAVARES**

**PEDRO CERIONI TOGNATO**

**RAFAEL SOUZA DA SILVA**

**APLICATIVO ANDROID FIT CONTROL**

**DOCUEMNTAÇÃO E DESENVOLVIMENTO DO APLICATIVO**

Atividade realizada pelos alunos do 5º semestre do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Metodista de São Paulo para atendimento ao módulo Desenvolvimento Embarcado através do Projeto de Ação Profissional - PAP.

**SÃO BERNARDO DO CAMPO**

**2016**

**MATHIAS NEVES TAVARES**

**PEDRO CERIONI TOGNATO**

**RAFAEL SOUZA DA SILVA**

**APLICATIVO ANDROID FIT CONTROL**

**DOCUEMNTAÇÃO E DESENVOLVIMENTO DO APLICATIVO**

Atividade realizada pelos alunos do 5º semestre do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Metodista de São Paulo para atendimento ao módulo Desenvolvimento Embarcado através do Projeto de Ação Profissional - PAP.

**Área de Concentração: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Data da Apresentação: \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Resultado: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**BANCA EXAMINADORA**

|  |  |  |
| --- | --- | --- |
| **Rafael Sakurai** | **Prof.** |  |
| **Universidade Metodista de São Paulo** | | |

|  |  |  |
| --- | --- | --- |
| **Tomaz Sasaki** | **Prof.** |  |
| **Universidade Metodista de São Paulo** | | |

|  |  |  |
| --- | --- | --- |
| **Fábio Versolato** | **Prof.** |  |
| **Universidade Metodista de São Paulo** | | |

Resumo

Este trabalho apresenta um projeto de desenvolvimento de um aplicativo Android Fit Control, desenvolvido pela empresa Dev Solutions. O aplicativo Fit Control gerencia o treino de acordo com seu desempenho, performance e saúde, evitando o overtraining e a lesão muscular. A escolha para desenvolver um aplicativo dessa categoria faz com que o usuário tenha um melhor gerenciamento de seu treino. Por isso, para esse projeto foi desenvolvido uma interface totalmente limpa e interativa para que o usuário possa ter a melhor experiência ao utilizar o aplicativo. Complementando, foi desenvolvido um banco de dados que supre as necessidades básicas do software.

Palavras-chave: treino, aplicativo, desenvolvimento, fit, solutions.

Abstract

This academic work presents a Android application development project, the Fit Control, developed by Dev Solutions. The app Fit Control manages the workout according your performance, health, avoiding the overtraining and the muscle injury. The choice to develop an application it causes the user have a greater management of your workout. For this project a clean and interactive interface was developed so that user may have the best experience using this application. Complementing, it developed a database that meets the basic needs of the software.

Key-words: workout, application, development, fit, solutions.

Lista de Tabelas

[Tabela 1 - Cronograma de Trabalho 9](#_Toc447800428)

Tabela de Figuras

[Figura 1 - Diagrama de caso de uso 15](#_Toc447800440)

[Figura 2 - Modelagem física 17](#_Toc447800441)

Sumário

1. Introdução

Esse trabalho tem como propósito o desenvolvimento de um aplicativo na plataforma Android, nomeado de Fit Control. O aplicativo Fit Control tem como principal objetivo gerenciar seu treino de forma adequada, evitando o *overtraining* e a lesão muscular, isso que também conta com o auxílio de um instrutor para o treino adequado. A Dev Solutions é uma empresa que está há 15 anos no mercado, atuando em soluções desktop, web e mobile. A escolha para desenvolver esse tipo de aplicativo é devido a grande popularização das academias e da musculação, e consequentemente o surgimento de diversos treinos, e por isso a necessidade de um aplicativo que administrasse adequadamente o treino de acordo com a saúde e resistência do usuário.

* 1. Cronograma de Trabalho

Para melhor prosseguimento das tarefas e do projeto como um todo, a Dev Solutions elaborou um cronograma de trabalho onde todas as tarefas/fases dos projetos foram dividas igualmente entre os membros da equipe, levando em consideração.

|  |  |  |  |
| --- | --- | --- | --- |
| **INICIATIVA** | **AÇÃO - DESCRIÇÃO DE CADA AÇÃO** | **PERÍODO OU DATA PARA ENREGA** | **RESPONSÁVEIS** |
|
| **Desenvolvimento e Modelagem** | Definição da equipe de trabalho | 19/fev-22/fev | Todos |
| Elaboração do modelo lógico do banco de dados | 22/fev-11/abr | Pedro |
| Definição e descrição do design pattern utilizado | 22/fev-11/abr | Rafael e Pedro |
| Medição da maturidade dos processos de desenvolvimento | 22/fev-11/abr | Mathias |
| Desenvolvimento do Aplicativo | 22/fev-23/mai | Todos |
| Modelagem UML | 22/fev-11/abr | Rafael |
| Entrega da Prévia | 22/fev-11/abr | Todos |
| Entrega parcial da Documentação | 22/fev-11/abr | Todos |
| Requisitos | 22/fev-11/abr | Rafael e Mathias |
| Elaboração dos scripts dos bancos de dados | 22/fev-23/mai | Rafael |
| Versionar projeto no GitHub | 22/fev-23/mai | Mathias |
| Modelagem dos processos de qualidade para nível F do MPS.BR | 22/fev-23/mai | Pedro e Mathias |
| Entrega da documentação | 22/fev-23/mai | Todos |
| Entrega do projeto | 22/fev-23/mai | Todos |

Tabela - Cronograma de Trabalho

1. Análise dos Requisitos Funcionais e Não Funcionais
   1. Requisitos Funcionais

Esse item tem como propósito a descrição dos requisitos funcionais do sistema.

* + 1. Requisitos de Sistema

[RF009] Disponibilizar Dieta

O sistema deve disponibilizar as dietas para o usuário assim que as mesmas forem cadastradas.

Prioridade: [x] Essencial [ ] Importante [ ] Desejável

Entradas e pré-condições: Dieta e aluno cadastrados no aplicativo.

Saídas e pós-condições: Dieta disponibilizada para o aluno.

[RF009] Disponibilizar treino

O sistema deve disponibilizar os treinos para o usuário assim que os mesmos forem cadastrados.

Prioridade: [x] Essencial [ ] Importante [ ] Desejável

Entradas e pré-condições: Treino e alunos cadastrados.

Saídas e pós-condições: Treinos disponibilizados para o aluno.

* + 1. Requisitos de Usuário

**[RF001] Cadastrar usuário**

O sistema deve permitir que o consultor cadastre um novo usuário

**Prioridade**: [x] Essencial [ ] Importante [ ] Desejável

**Entradas e pré-condições:**

**Saídas e pós-condições:** Novo usuário cadastrado

**[RF002] Cadastrar treino**

O sistema deve permitir que o consultor cadastre um treino para o usuário.

**Prioridade**: [x] Essencial [ ] Importante [ ] Desejável

**Entradas e pré-condições**: Usuário deve estar cadastrado.

**Saídas e pós-condições:** Treino cadastrado

**[RF003] Cadastrar dieta**

O sistema deve permitir que o consultor cadastre uma dieta para o usuário.

**Prioridade**: [x] Essencial [ ] Importante [ ] Desejável

**Entradas e pré-condições**: Usuário e treino devem estar cadastrados.

**Saídas e pós-condições:** Dieta cadastrada.

**[RF004] Disponibilizar treino**

O sistema deve permitir que o consultor disponibilize o treino para o aluno, após cadastrado.

**Prioridade**: [x] Essencial [ ] Importante [ ] Desejável

**Entradas e pré-condições**: Usuário e treino devem estar cadastrados.

**Saídas e pós-condições:** Treino disponibilizado.

**[RF005] Disponibilizar dieta**

O sistema deve permitir que o consultor disponibilize a dieta para o aluno.

**Prioridade**: [x] Essencial [ ] Importante [ ] Desejável

**Entradas e pré-condições**: Usuário, dieta e treino devem estar cadastrados.

**Saídas e pós-condições:** Dieta disponibilizada.

**[RF006] Realizar feedback ao aluno**

O sistema deve permitir que o consultor dê um feedback ao aluno.

**Prioridade**: [x] Essencial [ ] Importante [ ] Desejável

**Entradas e pré-condições**: Usuário e treino devem estar cadastrados.

**Saídas e pós-condições:** feedback realizado.

**[RF007] Cadastrar treino**

O sistema deve permitir que o consultor cadastre um treino para o usuário.

**Prioridade**: [x] Essencial [ ] Importante [ ] Desejável

**Entradas e pré-condições**: Usuário deve estar cadastrado.

**Saídas e pós-condições:** Treino cadastrado

**[RF008] Enviar mensagem ao consultor**

O sistema deve permitir que o usuário envie mensagens/feedback ao consultor.

**Prioridade**: [x] Essencial [ ] Importante [ ] Desejável

**Entradas e pré-condições:** Usuário deve estar cadastrado.

**Saídas e pós-condições:** Mensagem/feedback enviada ao consultor.

**[RF008] Acessar feedback**

O sistema deve permitir que o usuário acesse os feedbacks.

**Prioridade**: [x] Essencial [ ] Importante [ ] Desejável

**Entradas e pré-condições:** Usuário deve estar cadastrado.

**Saídas e pós-condições:** Acesso ao feedback liberado.

* + 1. Requisitos de Administrador
  1. Requisitos Não Funcionais

Esse item tem como propósito a descrição dos requisitos não funcionais do sistema.

* + 1. Requisitos de Aparência e Estilo

[RNF001] Aparência e estilo

O sistema deve possuir predominante as cores azul, vermelho e cinza.

Prioridade: [x] Essencial [ ] Importante [ ] Desejável

Entradas e pré-condições:

Saídas e pós-condições:

[RNF002] Aparência e estilo

O sistema deve ter como fonte padrão a fonte “Arial” em todo sistema.

Prioridade: [x] Essencial [ ] Importante [ ] Desejável

Entradas e pré-condições:

Saídas e pós-condições:

* + 1. Requisitos de Usabilidade e Humanidade

[RNF003] Idioma

O sistema deve ter como padrão a língua portuguesa.

Prioridade: [x] Essencial [ ] Importante [ ] Desejável

Entradas e pré-condições:

Saídas e pós-condições:

[RNF004] Interatividade

O sistema deve ser de fácil entendimento para todo tipo de usuário.

Prioridade: [x] Essencial [ ] Importante [ ] Desejável

Entradas e pré-condições:

Saídas e pós-condições:

[RNF005] Interatividade

O sistema deve ter fácil manuseio para uma melhor interação com todos os tipos de usuários.

Prioridade: [x] Essencial [ ] Importante [ ] Desejável

Entradas e pré-condições:

Saídas e pós-condições:

[RNF006] Interatividade

O sistema deve ter simbologias padronizadas para facilitar na utilização do sistema para qualquer tipo de usuário.

Prioridade: [x] Essencial [ ] Importante [ ] Desejável

Entradas e pré-condições:

Saídas e pós-condições:

1. Diagramas
   1. Diagrama de Classes
   2. Diagrama de Caso de Uso

Abaixo está o diagrama de Caso de Uso utilizado para modelagem do aplicativo Fit Control.

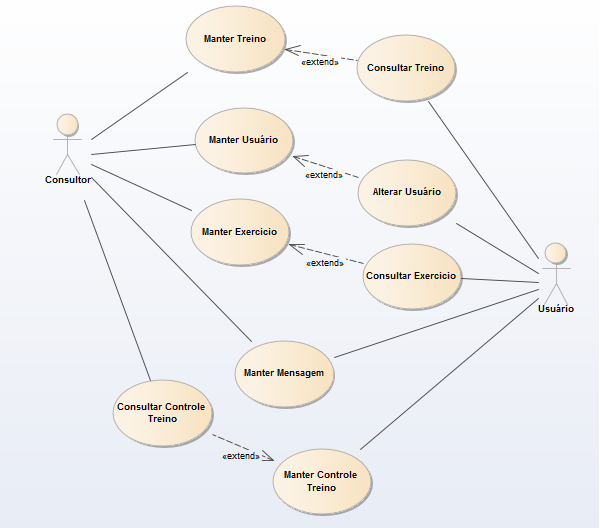


Figura 1 - Diagrama de caso de uso

* 1. Diagrama de Atividade

1. Design Pattern

No desenvolvimento do projeto foram utilizados os design patterns que estão descritos nos próximos itens.

* 1. Singleton
* Será utilizado como padrão um objeto único e instanciado uma única vez para a conexão com o banco de dados.
* Será utilizado como padrão um objeto único e instanciado uma única vez para a utilização das chamadas de serviços.
  1. Composite

Será utilizado como padrão uma classe que em momento de execução definirá qual a instância apropriada para execução, fazendo assim com que tenhamos centralizado um objeto todos as ações.

* 1. Factory Method

Será utilizado para métodos genéricos que conforme uma ação desejada, terão comportamentos diferentes.

1. Desenvolvimento

Esse capítulo aborda todo processo de desenvolvimento do aplicativo, desde a concepção inicial, documentação, diagramação, até o desenvolvimento do código e resultados finais.

* 1. Processos da qualidade

A Dev Solutions está na busca contínua para melhorar a qualidade dos seus processos de desenvolvimento de software, pois entende que qualificar seus processos, trará uma melhor competitividade no mercado, que por sua vez encontra-se mais exigente. Para isso a empresa deve medir a maturidade de seus processos de desenvolvimento de software e implementar um nível superior ao atual. Foi utilizado como base o modelo de referência MPS.Br.

* + 1. Nível F – Parcialmente Gerenciado
    2. Nível G - Gerenciado

1. Resultados

Nesse capítulo são abordados os resultados da aplicação, sendo eles todas as telas da aplicação e suas respectivas funções. A tela apresentada abaixo ( Figura 2 ) é a tela inicial do aplicativo. A mesma contém um plano de fundo, e dois botões: *Login* e *Cadastro*.



Figura - Tela inicial

Ao clicar no botão  *Login,* é apresentado um pop up ( Figura 3 ) para inserir as credenciais de acesso (usuário e senha), e o botão Confirma, para validação de dados.

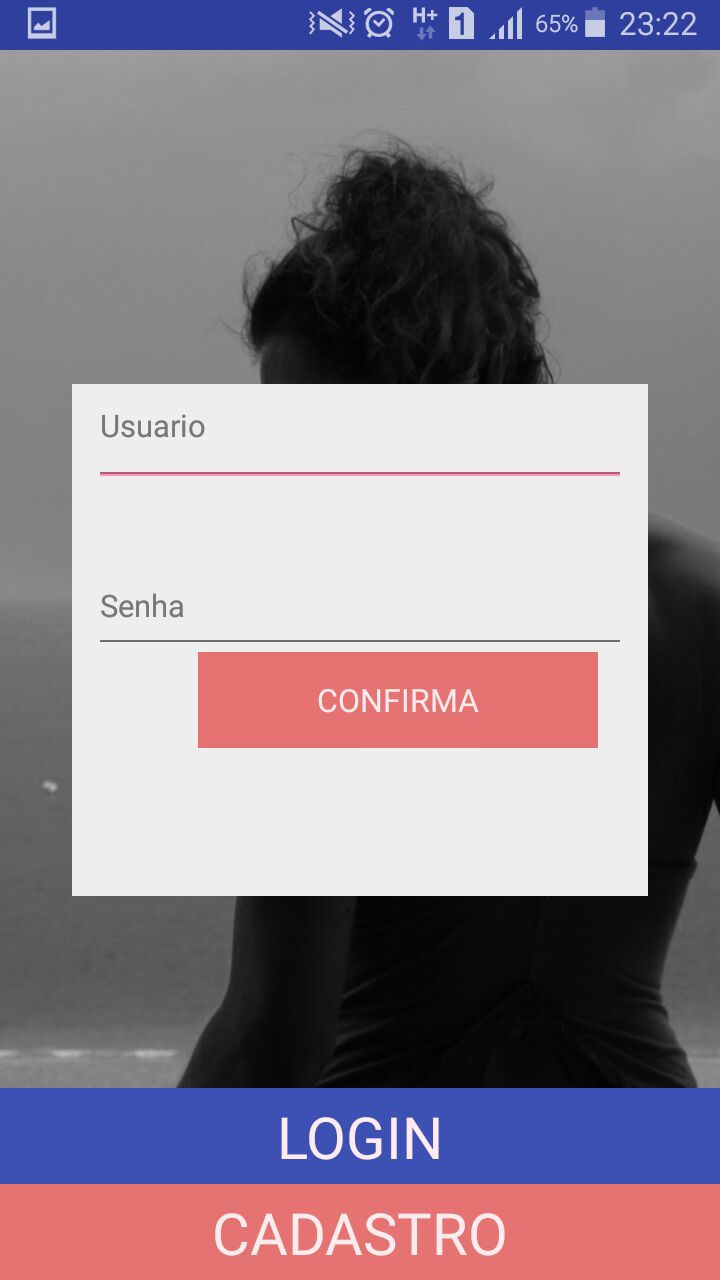


Figura - Campos usuário e senha

Ao clicar no botão *Cadastro,* o usuário é direcionado para uma tela onde o mesmo deve inserir os dados requisitados para criar uma conta no aplicativo Fit Control ( Figura 4 ) .

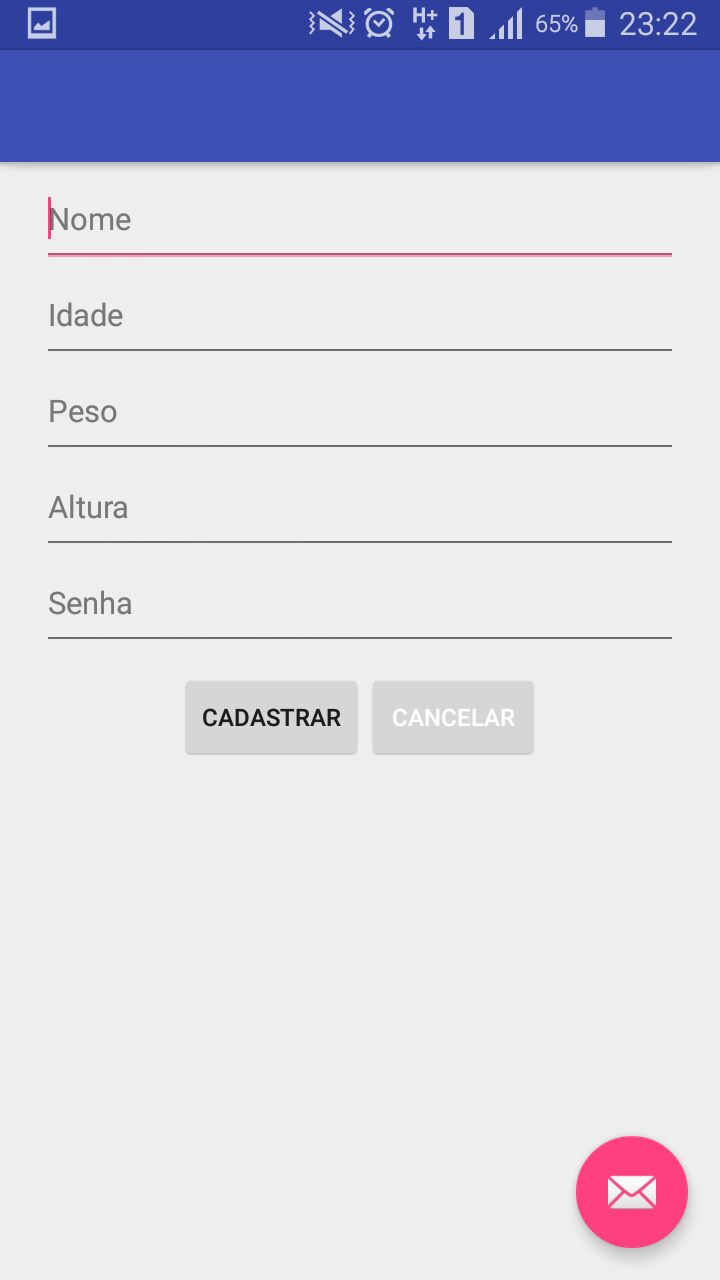


Figura - Tela cadastro de usuário

Ao entrar com seu usuário e senha, o usuário é direcionado para uma tela onde o mesmo deve atribuir uma nota referente à dois requisitos antes de iniciar seu treino: Fadiga e Dor muscular ( Figura 5 ).

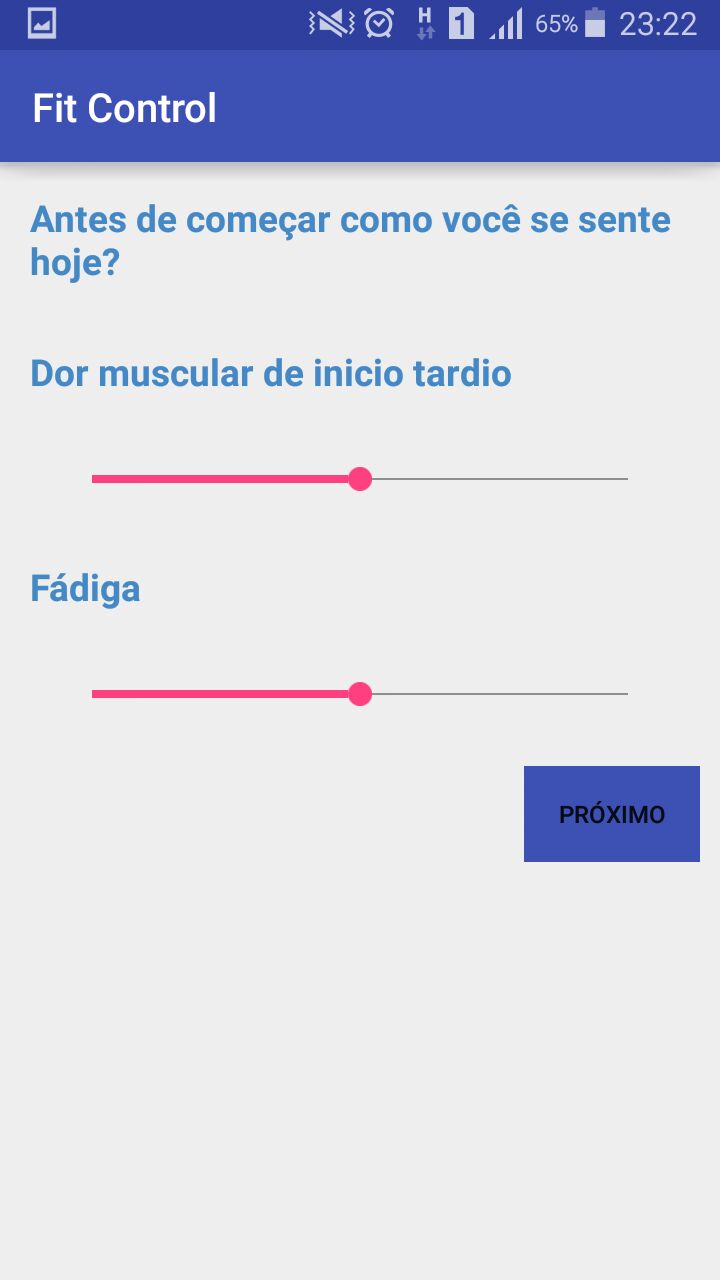


Figura - Tela inicial app

Através do menu principal, o usuário pode escolher a opção “Histórico de Treinos”, onde o mesmo pode verificar todos treinos e dados de treinos já realizados anteriormente ( Figura 6 ).

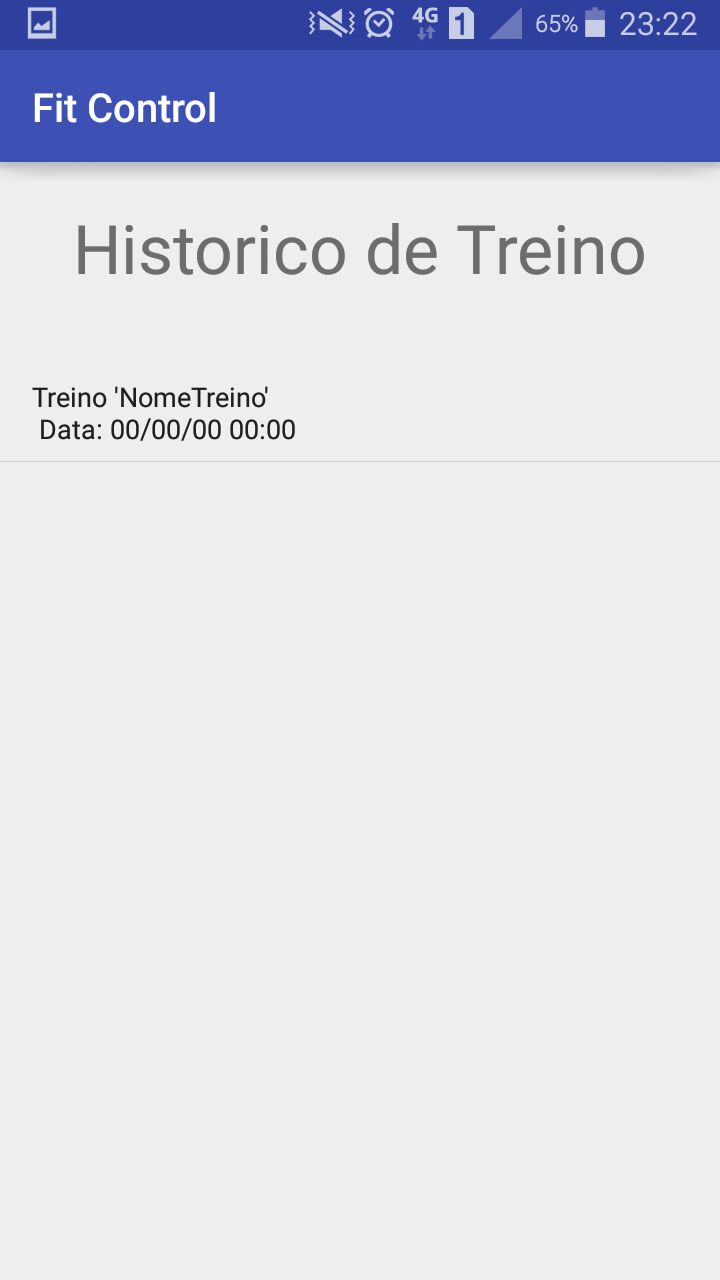


Figura - Histórico treinos

1. Discussão
2. Melhorias Futuras
3. Conclusão
4. Referências

Escola de Android.

Disponível em: <http://escoladeandroid.blogspot.com.br/2012/02/android-gerando-graficos-com-google.html> Acesso em 05 de Abril de 2016

StackOverflow em Português <http://pt.stackoverflow.com/questions/29718/gr%C3%A1ficos-no-android> Acesso em 03 de Abril de 2016

Apêndice A – Banco de Dados: Modelagem Física e Lógica

**Modelagem Física**

Abaixo segue modelagem física do banco de dados utilizado no aplicativo Fit Control.

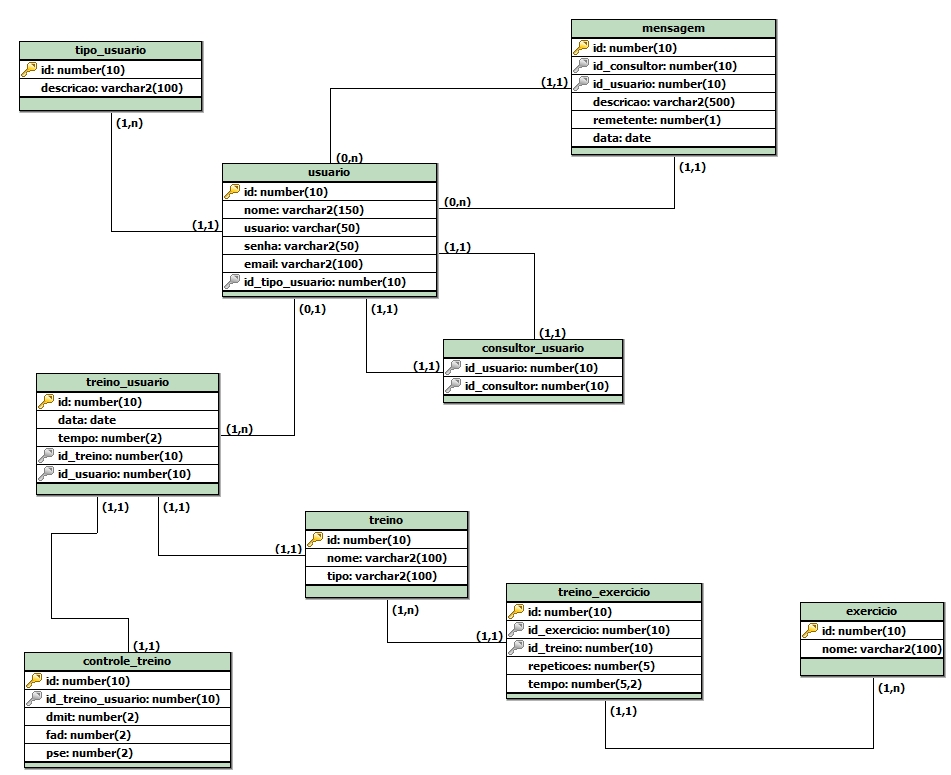


Figura 7 - Modelagem física

**Modelagem Lógica**

Abaixo segue script PL/SQL do banco de dados utilizado na aplicação FIT Control.

CREATE TABLE tipo\_usuario (

id NUMBER(10) PRIMARY KEY,

descricao VARCHAR2(100)

);

CREATE TABLE usuario (

id NUMBER(10) PRIMARY KEY,

nome VARCHAR2(150),

usuario VARCHAR2(50),

senha VARCHAR2(50),

email VARCHAR2(100),

id\_tipo\_usuario NUMBER(10),

CONSTRAINT FK\_TIPO\_USUARIO

FOREIGN KEY (id\_tipo\_usuario)

REFERENCES tipo\_usuario(id)

);

CREATE TABLE mensagem (

id NUMBER(10) PRIMARY KEY,

id\_consultor NUMBER(10),

id\_usuario NUMBER(10),

descricao VARCHAR2(500),

remetente NUMBER(1),

data DATE,

CONSTRAINT FK\_CONSULTOR

FOREIGN KEY (id\_consultor)

REFERENCES usuario(id),

CONSTRAINT FK\_USUARIO

FOREIGN KEY (id\_usuario)

REFERENCES usuario(id)

);

CREATE TABLE consultor\_usuario (

id\_usuario NUMBER(10),

id\_consultor NUMBER(10),

CONSTRAINT FK\_CONSULTOR

FOREIGN KEY (id\_consultor)

REFERENCES usuario(id),

CONSTRAINT FK\_USUARIO

FOREIGN KEY (id\_usuario)

REFERENCES usuario(id)

);

CREATE TABLE treino (

id NUMBER(10) PRIMARY KEY,

nome VARCHAR2(100),

tipo VARCHAR2(100)

);

CREATE TABLE treino\_usuario (

id NUMBER(10) PRIMARY KEY,

data DATE,

tempo

);

CREATE TABLE controle\_treino (

id NUMBER(10) PRIMARY KEY,

id\_treino\_usuario NUMBER(10),

dmit NUMBER(2),

fad NUMBER(2),

pse NUMBER(2),

CONSTRAINT FK\_ID\_TREINO\_USUARIO

FOREIGN KEY (id\_treino\_usuario)

REFERENCES treino\_usuario(id)

);

CREATE TABLE exercicio (

id NUMBER(10) PRIMARY KEY,

nome VARCHAR2(100),

);

CREATE TABLE treino\_exercicio (

id NUMBER(10) PRIMARY KEY,

id\_exercicio NUMBER(10),

id\_treino NUMBER(10),

repeticoes NUMBER(5),

tempo NUMBER(5,2),

CONSTRAINT FK\_ID\_EXERCICIO

FOREIGN KEY (id\_exercicio)

REFERENCES exercicio(id),

CONSTRAINT FK\_ID\_TREINO

FOREIGN KEY (id\_treino)

REFERENCES treino(id),

);