

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Visconde da Graça



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE – CAMPUS PELOTAS - VISCONDE DA GRAÇA
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

SISTEMA DE GERENCIAMENTO PARA ESTÚDIO DE PILATES

Bruno Falcão Tessmer

Pelotas, 2019

Bruno Falcão Tessmer

SISTEMA DE GERENCIAMENTO PARA ESTÚDIO DE PILATES

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Técnico, do curso Técnico em Desenvolvimento de Sistemas, do Instituto Federal Sul-Rio-Grandense – Câmpus Pelotas - Visconde da Graça.

Orientadora: Prof. Dra. Verlani Timm Hinz

Pelotas, dezembro de 2019.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1: Modelo de caso de uso..... | 9 |
| Figura 2: Diagrama de entidade e relacionamento..... | 10 |
| Figura 3: Modelo lógico | 11 |
| Figura 4: Tela de login | 14 |
| Figura 5: Menu do sistema | 14 |
| Figura 6: Cadastro de pacientes..... | 15 |
| Figura 7: Cadastro de usuários | 15 |
| Figura 8: Lista de pacientes..... | 16 |
| Figura 9: Ficha inicial do paciente | 16 |
| Figura 10: Ficha de evolução | 17 |
| Figura 11: Cadastro de exercícios | 17 |
| Figura 12: Evoluções do paciente | 18 |

SUMÁRIO

| | | |
|-----|--|----|
| 1 | INTRODUÇÃO..... | 5 |
| 1.1 | AMBIENTE DE DESENVOLVIMENTO DO SISTEMA | 5 |
| 1.2 | TEMA..... | 6 |
| 1.3 | MOTIVAÇÕES | 6 |
| 1.4 | OBJETIVOS | 6 |
| 2 | ESPECIFICAÇÃO DE REQUISITOS..... | 7 |
| 2.1 | MÉTODOS DE ESPECIFICAÇÃO DE REQUISITOS | 7 |
| 2.2 | REQUISITOS FUNCIONAIS | 7 |
| 2.3 | REQUISITOS NÃO FUNCIONAIS | 8 |
| 3 | MODELAGEM | 9 |
| 3.1 | MODELO DE CASOS DE USO | 9 |
| 3.2 | MODELAGEM CONCEITUAL DO BANCO DE DADOS | 10 |
| 3.3 | MODELAGEM LÓGICA DO BANCO DE DADOS..... | 11 |
| 4 | TECNOLOGIAS UTILIZADAS | 12 |
| 5 | DESCRIÇÃO DO SISTEMA | 14 |
| 6 | CONSIDERAÇÕES FINAIS..... | 19 |
| 7 | REFERÊNCIAS | 20 |
| | APÊNDICE I – Instruções SQL para a criação da base de dados do sistema.. | 21 |

1 INTRODUÇÃO

Com o passar dos anos, o número de locais que nos possibilitam realizar atividades físicas têm aumentado significativamente, dessa forma, academias e centros esportivos têm se tornado cada vez mais comuns na sociedade. Dentre estes locais estão, também, os estúdios de pilates, que possibilitam realizar atividades físicas, para várias faixas etárias, com exercícios específicos, contando com um constante acompanhamento profissional de um fisioterapeuta.

No entanto, as demandas nos estúdios de Pilates tem crescido atualmente, o que gerou uma necessidade de informatização de rotinas que antes eram feitas manualmente. Diante disso, o presente trabalho tem por objetivo o desenvolvimento de um sistema para o gerenciamento das atividades dos pacientes do estúdio de pilates Cliff Pilates. O sistema tem a função de gerenciar os pacientes do estúdio, mantendo um controle das atividades semanais realizadas pelos pacientes, assim como, gerenciar os exercícios que são propostos de acordo com as necessidades dos pacientes.

Este controle, atualmente, é feito manualmente, por meio de fichas, o que pode se tornar algo trabalhoso e demorado, devido ao aumento do número de pacientes. Diante disso, esse software tem a função agilizar esse processo, assim como, evitar a perda de dados, obtendo uma melhor eficácia na gestão do estúdio.

1.1 AMBIENTE DE DESENVOLVIMENTO DO SISTEMA

O estúdio de pilates denominado de Cliff Pilates localiza-se na Rua Nilo Peçanha, em Pelotas, RS – Brasil, tendo como responsável a fisioterapeuta e proprietária Paula de Oliveira Nobre.

É um estúdio, de porte médio, que atende diversos tipos de pacientes, desde os que possuem recomendação médica, tanto os que frequentam o estúdio por motivos pessoais.

Com o surgimento da necessidade de um melhor controle dos dados dos pacientes matriculados no estúdio e exercícios por eles realizados, surgiu a oportunidade do desenvolvimento deste sistema para o local, a fim de automatizar alguns processos que, atualmente, são feitos manualmente.

1.2 TEMA

O tema deste trabalho consiste em um sistema que auxilia na gestão dos dados sobre os pacientes de um estúdio de pilates, além de possibilitar um maior controle sobre as atividades realizadas por seus alunos. O software busca melhorar e registrar as atividades realizadas no estúdio de pilates.

1.3 MOTIVAÇÕES

A principal motivação para o desenvolvimento desse sistema é informatizar as atividades de acompanhamento das fichas e evoluções dos pacientes, provendo uma gestão do estúdio mais automatizada.

Diante disso, será desenvolvido um software que facilitará o gerenciamento para instrutores e um melhor controle dos pacientes do estúdio. A fisioterapeuta, e também proprietária do Cliff Pilates, aceitou ser cliente do software a ser desenvolvido, e com sua ajuda, foi desenvolvido o sistema para suprir algumas demandas do estúdio.

Atualmente as informações são salvas em planilhas eletrônicas ou até mesmo em papéis, formas de gestão com alta probabilidade de perda de dados.

1.4 OBJETIVOS

O objetivo geral deste trabalho é o desenvolvimento de um sistema que gerencie os dados dos pacientes para um estúdio de pilates, facilitando o acompanhamento dos exercícios realizados pelos pacientes e permitindo acesso a um conjunto de informações a respeito dos mesmos.

Para atingir o objetivo geral devem ser contemplados os seguintes objetivos específicos:

- Definir os requisitos do sistema.
- Possibilitar o gerenciamento de pacientes e exercícios.
- Realizar o controle das atividades e avaliações físicas realizadas pelos pacientes do estúdio de Pilates.
- Permitir que o software mantenha instrutor e aluno atualizados com relatórios sobre os pacientes, sempre que for requisitado.
- Emitir relatórios detalhados das atividades realizadas pelos pacientes.
- Treinar os usuários para utilização do sistema.

2 ESPECIFICAÇÃO DE REQUISITOS

2.1 MÉTODOS DE ESPECIFICAÇÃO DE REQUISITOS

Os requisitos funcionais e não funcionais que foram utilizados para o desenvolvimento do software foram adquiridos através de entrevistas e análise de documentos existentes.

As entrevistas foram realizadas com a proprietária e fisioterapeuta do estúdio que aceitou ser cliente deste produto que será produzido. Foram realizadas entrevistas, tanto pessoalmente, quanto por ferramentas web, como questionários do *Google* e *WhatsApp*. Além disso, foram consultadas fichas que a proprietária utiliza atualmente. Com base nas informações coletadas foi possível identificar os requisitos do sistema, os quais estão descritos nas próximas seções.

2.2 REQUISITOS FUNCIONAIS

O sistema deve viabilizar as seguintes funcionalidades:

REF01: Cadastro de instrutores que possuem acesso ao sistema. Atualmente não há necessidade de mais de um tipo de usuário, portanto só haverá um usuário administrador.

REF02: Gerenciamento de ficha de dados dos pacientes, que contém os dados pessoais dos pacientes.

REF03: Gerenciamento da ficha inicial do paciente contendo os dados clínicos do atendimento inicial do paciente.

REF04: Gerenciamento das fichas de evolução do paciente onde, semanalmente, são registradas as atividades realizadas pelos pacientes, assim como sua evolução clínica.

REF05: Gerenciamento dos exercícios que serão realizados pelos pacientes.

REF06: Emissão de relatórios, das evoluções dos pacientes através das suas fichas de evolução.

REF07: Realização de *login*, o instrutor realizará *login* no sistema.

2.3 REQUISITOS NÃO FUNCIONAIS

As características abaixo descrevem aspectos não funcionais que devem ser contemplados pelo sistema:

RENF01 - Operacionais: Ser desenvolvido em linguagem visual.

RENF02- Operacionais: Ter as informações armazenadas em um banco de dados através do Sistema Gerenciador de Banco de Dados.

RENF03 - Usabilidade: Ter um layout deve que se adapte a qualquer sistema Windows.

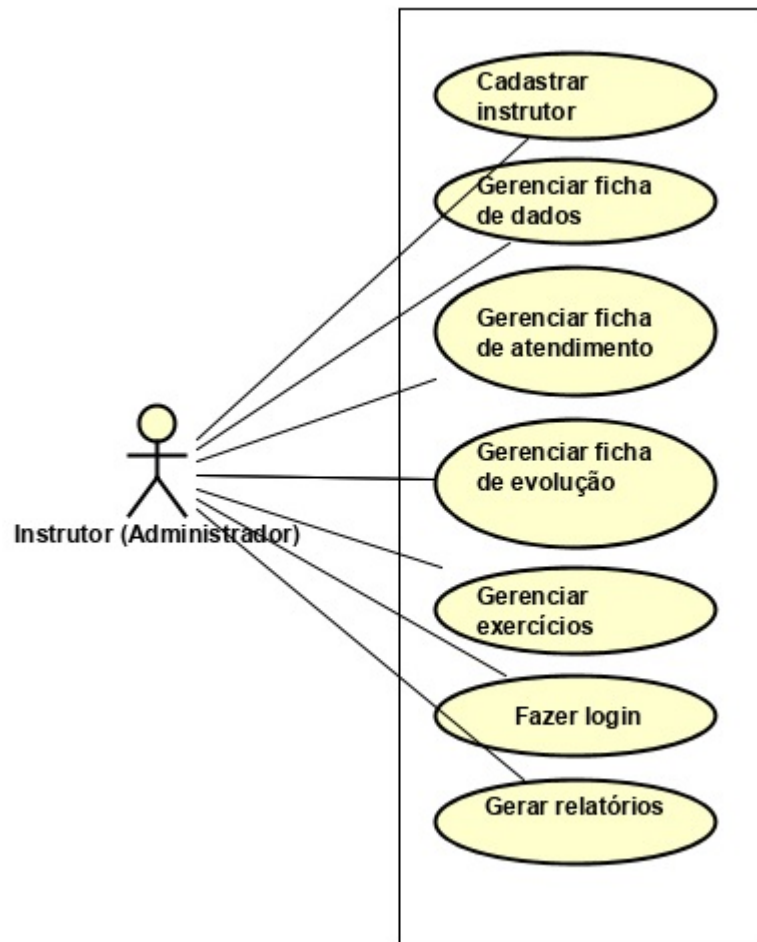
RENF04 - Compatibilidade: Ter compatibilidade com o sistema operacional Windows.

3 MODELAGEM

3.1 MODELO DE CASOS DE USO

O seguinte diagrama da Figura 1 foi criado a partir do software Astah UML, ferramenta de modelagem de diagrama UML.

Figura 1: Modelo de caso de uso.



Fonte: Elaborado pelo autor

O ator “Instrutor” representa de forma geral todos os usuários do sistema, que são também administradores. O caso de uso “Cadastrar instrutor” caracteriza a capacidade de um instrutor poder cadastrar mais instrutores. Estes instrutores poderão utilizar todos os recursos do sistema e terão acesso aos dados (cadastrais e clínicos, quanto os dados dos exercícios realizados pelos pacientes).

O caso de uso “Gerenciar ficha de dados” contém informações pessoais do paciente, como idade, sexo, nome, dentre outros. Os dados clínicos do paciente

estão representados pelo caso de uso “Gerenciar ficha de atendimento”, sendo esta preenchida no primeiro atendimento realizado com o paciente.

A ficha de evolução, por sua vez, serve para gerenciar as evoluções clínicas sofridas pelos pacientes ao longo do seu tratamento e está representada pelo caso de uso “Gerenciar ficha de evolução”.

Além disso, o caso de uso “Gerenciar exercícios”, possui dados dos exercícios que são propostas aos pacientes durante o seu tratamento.

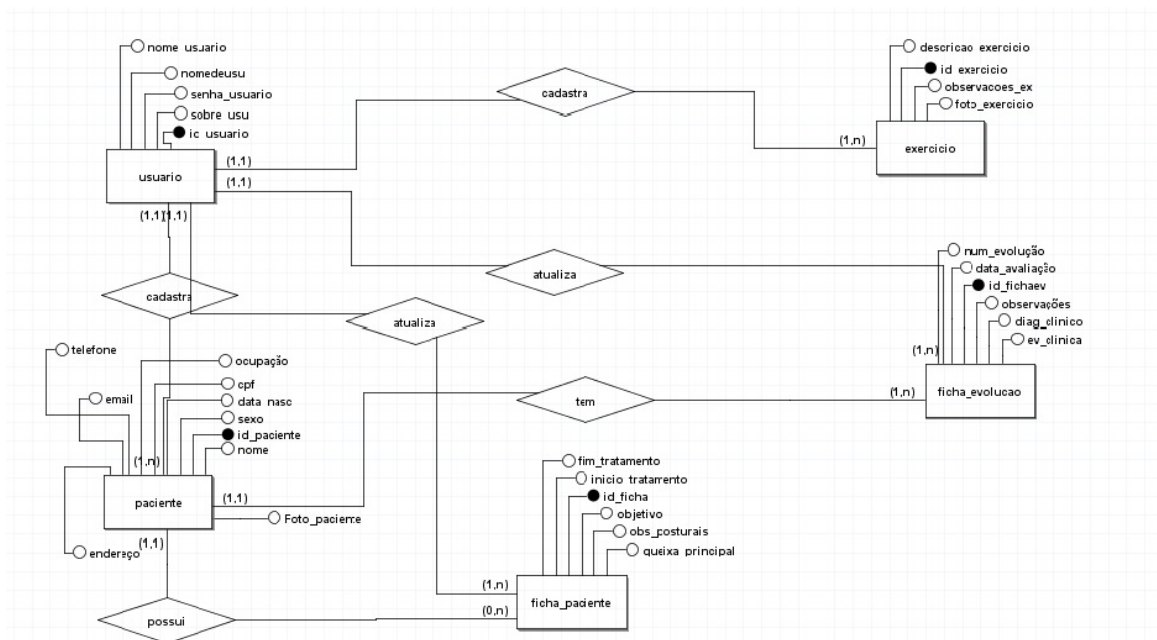
O caso de uso “Fazer login”, serve para que seja possível o acesso ao sistema, onde devem ser informados dados de usuário e senha.

A opção de geração de relatórios é representada pelo caso de uso “Gerar relatórios”, que contém os dados da ficha de evolução dos pacientes.

3.2 MODELAGEM CONCEITUAL DO BANCO DE DADOS

A Figura 2 mostra a representação gráfica do modelo ER, a fim de mostrar os dados armazenados e quais dados se relacionam.

Figura 2: Diagrama de entidade e relacionamento



Fonte: Elaborado pelo autor

O diagrama foi gerado através da ferramenta *brModelo*, ferramenta para criação e geração de modelos lógicos e conceituais.

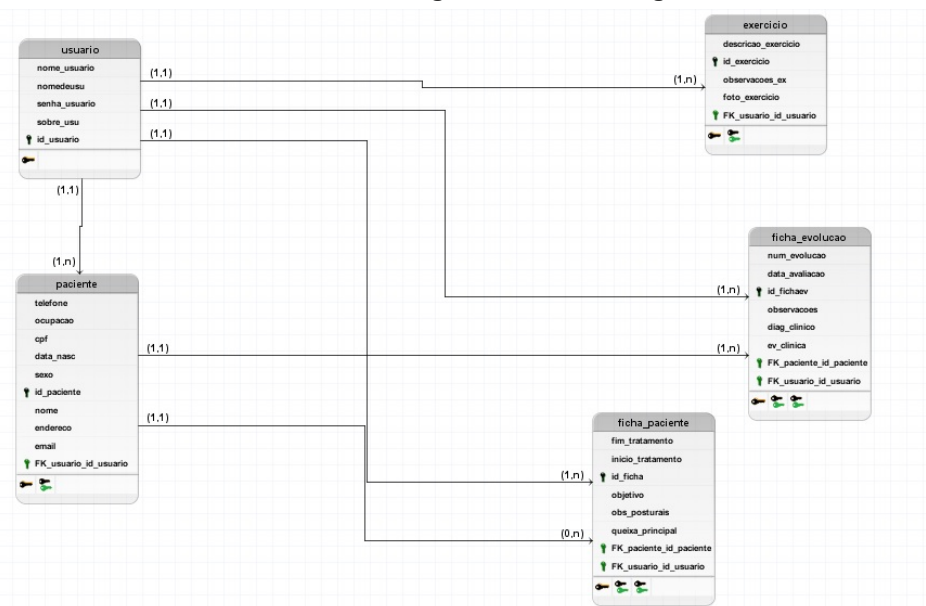
3.3 MODELAGEM LÓGICA DO BANCO DE DADOS

A modelagem lógica do banco de dados é descrita através do modelo relacional, o qual se caracteriza por descrever como os dados serão armazenados no sistema e como serão criados os relacionamentos do modelo conceitual. No modelo relacional os dados são representados por uma coleção de tabelas com seus respectivos atributos, sendo definidos os tipos de dados para cada atributo, bem como detalhadas as chaves primárias e incluídas as chaves estrangeiras. (HEUSER, 2009).

As chaves estrangeiras nas tabelas garantem a integridade referencial do modelo relacional, evitando que se faça referência a valores que não existam na base de dados. Essa integridade é garantida através dos atributos, uma vez construídos os modelos conceitual e lógico do sistema pode-se fazer a implementação do modelo lógico, com a criação da base de dados em um sistema gerenciador de banco de dados (SGBD) relacional.

A Figura 3 mostra o diagrama relacional do sistema. Pode-se observar o conjunto de tabelas que forma a base de dados do sistema, com a identificação de todos os atributos e seus respectivos tipos, bem como as chaves primárias e as chaves estrangeiras que programam os relacionamentos entre as tabelas.

Figura 3: Modelo lógico



Fonte: Elaborado pelo autor

4 TECNOLOGIAS UTILIZADAS

Neste projeto foram utilizadas ferramentas para modelagem do sistema e para modelagem e criação do banco de dados, bem como tecnologias que possibilitam a implementação de sistemas para acesso através local e também *Web*. Assim, a seguir são descritas resumidamente as ferramentas e tecnologias utilizadas no desenvolvimento do sistema.

Para a modelagem do sistema e do banco de dados foram utilizadas, respectivamente, as ferramentas *Astah*¹ e *brModelo*². O *Astah* foi utilizado exclusivamente para criar os modelos de caso de uso e o *brModelo*: foi empregado na criação dos diagramas de entidade e relacionamento e no modelo lógico do sistema.

O *MySQL* foi o sistema gerenciador de banco de dados relacional utilizado na construção da base de dados do sistema, sendo acessado através do aplicativo *XAMPP*³, um servidor independente de plataforma e também sendo um software livre. O *MySQL* é um SGBD de código aberto executado em diversas plataformas. Tem suporte a controle transacional, gatilhos e procedimentos armazenados, implementa integridade referencial através da definição de relações entre as tabelas. Permite a criação de contas de usuários com a definição de permissões de acesso que abrangem desde bases de dados como um todo até especificamente colunas de uma tabela.

O *PHPMYAdmin*⁴ é um aplicativo *Web*, com código aberto, para administração do SGBD. Através deste aplicativo é possível criar e remover bases de dados, criar, remover e alterar tabelas, inserir, remover e editar campos e executar comandos SQL, além de criar e manipular as chaves (primárias, estrangeiras, únicas) e índices, gerenciar procedimentos armazenados e gatilhos; dentre outras funcionalidades.

O *C#* foi a linguagem programação utilizada para o desenvolvimento das funcionalidades do sistema. O *C#* é uma linguagem de programação desenvolvida pela Microsoft, e é basicamente orientada a objetos, possui forte influência da

¹<http://astah.net>

² <http://www.sis4.com>

³ <http://www.apachefriends.org>

⁴ <http://www.phpmyadmin.net>

linguagem C++, na qual foi baseada, mas tem influência de muitas outras linguagens, como *Java*.

Para a manipulação da linguagem foi utilizado a IDE *Microsoft Visual Studio Community 2017*⁵, software que permite a criação de conteúdo dinâmico e visual em C#, através da criação de botões e telas de formas simples, possibilitando inserirmos eventos nestes componentes visuais da linguagem. O *Microsoft Visual Studio Community*, possui a extensão para a biblioteca de classes do *MySQL*, possibilitando a utilização dos comandos SQL e sendo possível utilizar um banco de dados para armazenamento das informações.

Além disso, o Visual Studio conta com extensões para criação de relatórios do sistema também dentro da IDE, o que torna o desenvolvimento dos mesmos muito prático.

A fim de controlar o código e ter uma gestão de versões do sistema, foi utilizada a plataforma *Github*⁶. O *GitHub* é uma plataforma de hospedagem de código-fonte com controle de versão. Ele permite que programadores, usuários cadastrado na plataforma contribuam em projetos privados e/ou *Open Source* de qualquer lugar do mundo. Para este trabalho, foi utilizado para ter segurança sob o código, visto que o mesmo fica armazenado em “nuvem”, assim, ressalto a importância da sua utilização.

⁵ <http://visualstudio.microsoft.com>

⁶ <http://github.com>

5 DESCRIÇÃO DO SISTEMA

Neste capítulo são apresentadas as principais funcionalidades do sistema, sendo caracterizado um fluxo de utilização a partir da exibição das principais interfaces geradas.

A Figura 4 representa a tela de *login* do usuário, no caso o instrutor. O mesmo entrará com o seu *login* e senha, caso ele não possua um usuário, nesta mesma tela ele poderá criar um novo, no botão “Criar usuário”. Em casa de erro de senha ou *login*, será exibida uma mensagem de erro para o usuário.

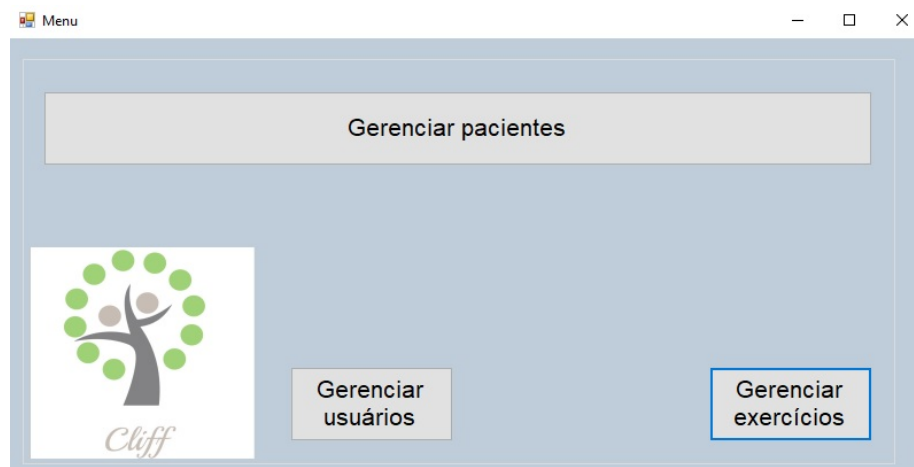
Figura 4: Tela de login



Fonte: Elaborado pelo autor

A Figura 5, representa, basicamente, o menu principal do sistema, contendo os botões que nos permitem navegar entra as funções.

Figura 5: Menu do sistema



Fonte: Elaborado pelo autor

. Na Figura 6 é representada a tela onde o usuário dará a entrada do paciente no sistema, ou seja, entrar com os dados cadastrais e pessoais dos usuários.

Figura 6: Cadastro de pacientes

The screenshot shows a web form titled 'Cadastro de pacientes:'. It contains the following fields and controls:

- Nome: text input field.
- Data de nascimento: date picker showing 21/11/2019.
- Endereço: text input field.
- Ocupação: text input field.
- E-Mail: text input field.
- Telefone: text input field.
- CPF: text input field.
- Sexo: radio buttons for 'Masculino' and 'Feminino'.
- Buttons: 'Cadastrar' and 'Sair'.

Fonte: Elaborado pelo autor.

Na Figura 6 é representada a tela onde é realizado o cadastro de usuários que terão acesso ao sistema.

Figura 7: Cadastro de usuários

The screenshot shows a web form titled 'Cadastro de usuário'. It contains the following fields and controls:

- Nome: text input field.
- Sobrenome: text input field.
- CPF: text input field.
- Nome de usuário: text input field.
- Senha: text input field.
- Repita sua senha: text input field.
- Tipo de usuário: radio buttons for 'Usuário Padrão' and 'Usuário Administrador'.
- Button: 'Cadastrar'.

Fonte: Elaborado pelo autor.

Na figura 7 é visualizada a lista de todos os pacientes cadastrados no sistema. O usuário pode selecionar o paciente que deseja gerenciar a ficha de atendimento inicial e clicar no botão “Nova ficha”, isso ocorrerá quando o paciente estiver realizando o cadastramento inicial no estúdio. O botão “Gerenciar fichas”,

serve para quando já existir uma ficha cadastrada para aquele paciente, o usuário poderá excluir uma ficha.

Figura 8: Lista de pacientes



Fonte: Elaborado pelo autor.

Na Figura 8, temos a ficha inicial do paciente, com os dados clínicos de quando ele deu entrada no estúdio. Após clicar no botão "Salvar", ele poderá começar a cadastrar as fichas de evolução.

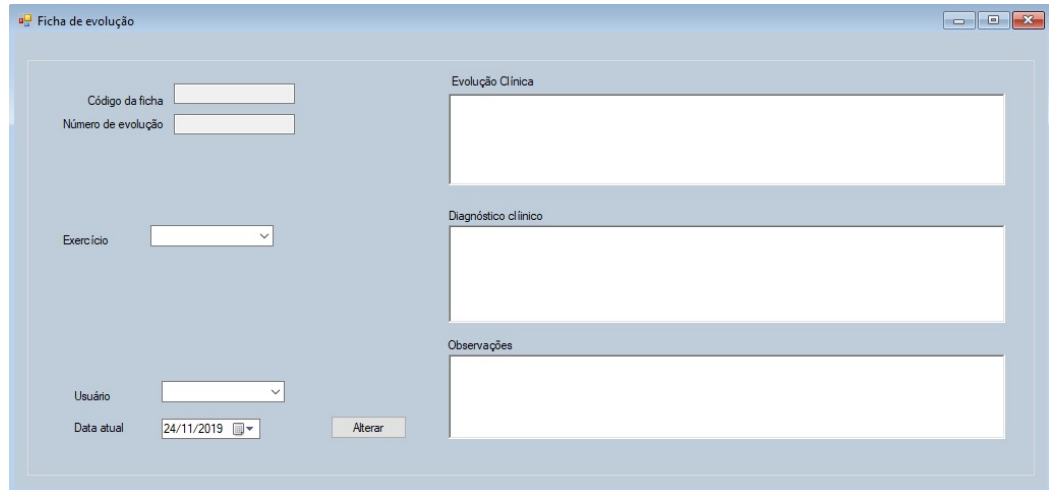
Figura 9: Ficha inicial do paciente

A imagem mostra uma janela de software intitulada "Ficha_de_paciente1". No topo, há uma barra de título com o ícone de uma pasta e o texto "Ficha_de_paciente1", além de botões de minimizar, maximizar e fechar. O formulário principal contém os seguintes elementos: "ID ficha:" seguido de um campo de entrada; "CPF do paciente:" seguido de um campo de entrada contendo o valor "03761804040"; "Data início tratamento:" seguido de um menu suspenso com o valor "21/11/2019"; "Final do tratamento:" seguido de um menu suspenso com o valor "quinta-feira"; "Observações posturais" seguido de uma grande área de texto; "Objetivo" seguido de uma grande área de texto; "Queixa principal" seguido de uma grande área de texto; e dois botões "Nova evolução" e "Salvar" na base esquerda.

Fonte: Elaborado pelo autor.

Na Figura 9, o cadastro das fichas de evolução, que será salvo no banco de dados do sistema.

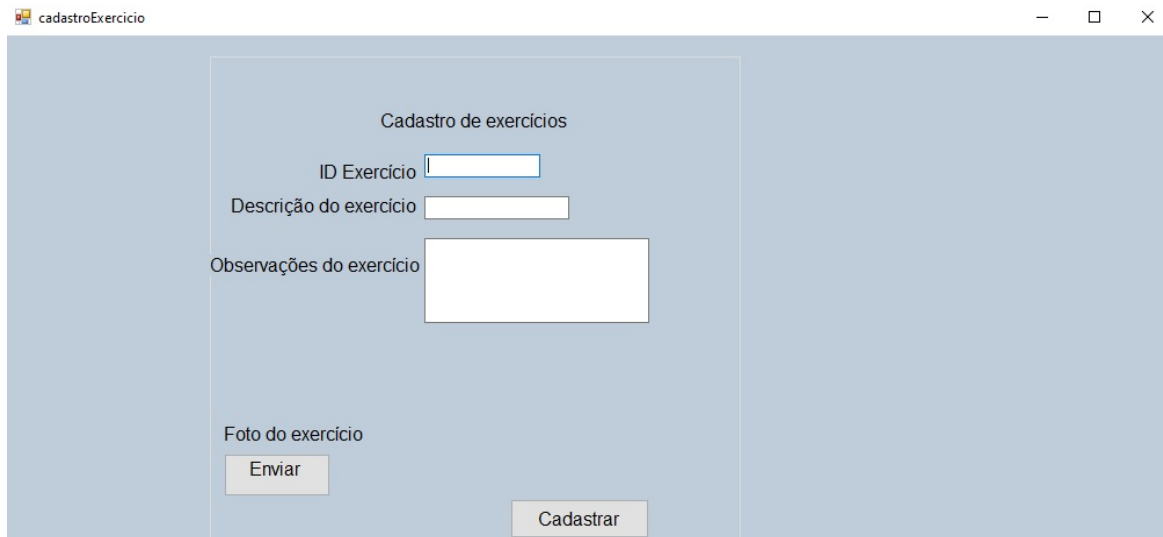
Figura 10: Ficha de evolução



Fonte: Elaborado pelo autor.

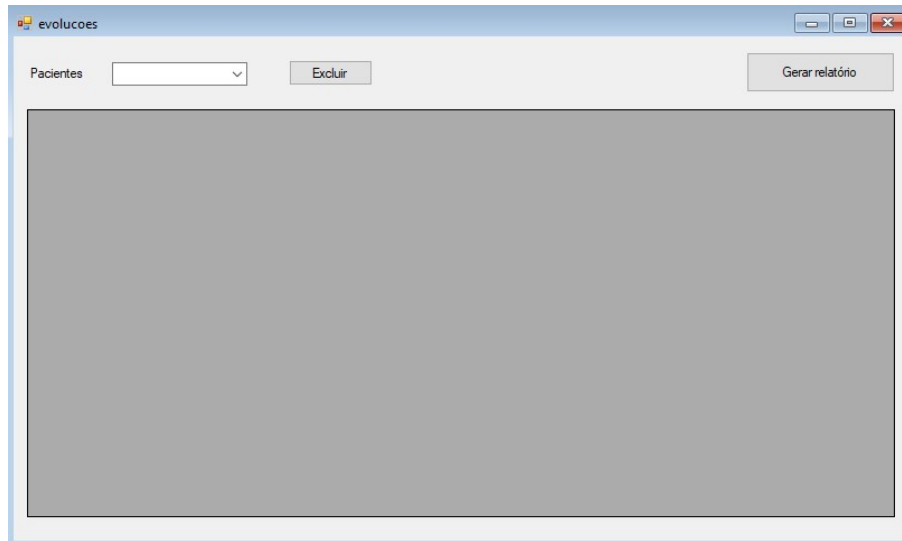
Na figura 10, tela de cadastro de exercícios, dentro do sistema, também é possível cadastrar exercícios, através desta tela.

Figura 11: Cadastro de exercícios



Fonte: Elaborado pelo autor.

A Figura 11, o usuário poderá ver todas as evoluções de um paciente, com a possibilidade de excluir a mesma e gerar um relatório de uma ficha x.

Figura 12: Evoluções do paciente

Fonte: Elaborado pelo autor.

6 CONSIDERAÇÕES FINAIS

Este projeto abrangeu o desenvolvimento de um sistema para gerenciar os dados pessoais e clínicos de pacientes, assim como os exercícios realizados pelos mesmos, em um estúdio de Pilates.

O desenvolvimento deste trabalho foi importante, pois, além de organizar as informações, tornando a gestão do estúdio mais automatizada, conseguiu resolver alguns problemas como a perda de informações, que antes ocorriam com frequência e acabavam gerando transtornos aos pacientes e à fisioterapeuta que necessitava da informação das informações atualizadas para dar andamento ao tratamento dos pacientes.

Observou-se que podem ocorrer mudanças ao longo do desenvolvimento de um sistema, isso só reforça a importância da elaboração de um bom projeto, sempre bem documentado. Na continuidade deste projeto entende-se como possibilidades de trabalhos futuros: (i) A possibilidade da criação de outro tipo de usuário, com menos permissões, não um usuário administrador somente, e (ii) a possibilidade da inserção de fotos dos exercícios dentro do cadastro de exercícios do sistema.

7 REFERÊNCIAS

COUGO, Paulo. **Modelagem Conceitual e Projeto de Bancos de Dados**. São Paulo: Elsevier, 1997. 283 p.

GALUPPO, Fabio. **Desenvolvendo com C#**. Bookman, 2004.

HEUSER, C. A. **Projeto de banco de dados**. 6. ed. Porto Alegre, RS: Bookman, 2009. 282 p.

LARMAN, C. **Utilizando UML e Padrões**. 3ª Edição. Porto Alegre: Bookman, 2011. 695 p.

LIBERTY, Jesse. **Programming C#: Building .NET Applications with C#**: Building .NET Applications. Programming C#, Cambridge, v. 4, p.302-605, fev. 2005.

PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. 6ª Edição. São Paulo: McGraw Hill, 2006.

SAADE, J. **C# - Guia do Programador**. São Paulo: Novatec, 2011.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

APÊNDICE I – Instruções SQL para a criação da base de dados do sistema

Neste apêndice são apresentadas as instruções SQL que permitem a criação da base de dados do sistema no SGBD MySQL.

```
CREATE DATABASE `estudio`
```

```
CREATE TABLE exercicio (  
    descricao_exercicio VARCHAR,  
    id_exercicio VARCHAR PRIMARY KEY,  
    observacoes_ex VARCHAR,  
    foto_exercicio VARCHAR,  
    FK_usuario_id_usuario VARCHAR  
);
```

```
CREATE TABLE ficha_evolucao (  
    num_evolucao VARCHAR,  
    data_avaliacao VARCHAR,  
    id_fichaev VARCHAR PRIMARY KEY,  
    observacoes VARCHAR,  
    diag_clinico VARCHAR,  
    ev_clinica VARCHAR,  
    FK_paciente_id_paciente VARCHAR,  
    FK_usuario_id_usuario VARCHAR  
);
```

```
CREATE TABLE ficha_paciente (  
    fim_tratamento TIMESTAMP,  
    inicio_tratamento VARCHAR,  
    id_ficha VARCHAR PRIMARY KEY,  
    objetivo VARCHAR,  
    obs_posturais VARCHAR,  
    queixa_principal VARCHAR,  
    FK_paciente_id_paciente VARCHAR,  
    FK_usuario_id_usuario VARCHAR  
);
```

```
CREATE TABLE paciente (  
    telefone VARCHAR,  
    ocupacao VARCHAR,  
    cpf VARCHAR,  
    data_nasc VARCHAR,  
    sexo VARCHAR,  
    id_paciente VARCHAR PRIMARY KEY,  
    nome VARCHAR,  
    endereco VARCHAR,  
    email VARCHAR,  
    Foto_paciente VARCHAR,  
    FK_usuario_id_usuario VARCHAR  
);
```

```
CREATE TABLE usuario (  
    nome_usuario VARCHAR,  
    nome_deusu VARCHAR,  
    senha_usuario VARCHAR,  
    sobre_usu VARCHAR,  
    id_usuario VARCHAR PRIMARY KEY  
);
```

```
ALTER TABLE exercicio ADD CONSTRAINT FK_exercicio_1  
    FOREIGN KEY (FK_usuario_id_usuario)  
    REFERENCES usuario (id_usuario)  
    ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE ficha_evolucao ADD CONSTRAINT FK_ficha_evolucao_1  
    FOREIGN KEY (FK_paciente_id_paciente)  
    REFERENCES paciente (id_paciente)  
    ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE ficha_evolucao ADD CONSTRAINT FK_ficha_evolucao_2
```

```
FOREIGN KEY (FK_usuario_id_usuario)  
REFERENCES usuario (id_usuario)  
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE ficha_paciente ADD CONSTRAINT FK_ficha_paciente_1  
FOREIGN KEY (FK_paciente_id_paciente)  
REFERENCES paciente (id_paciente)  
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE ficha_paciente ADD CONSTRAINT FK_ficha_paciente_2  
FOREIGN KEY (FK_usuario_id_usuario)  
REFERENCES usuario (id_usuario)  
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE paciente ADD CONSTRAINT FK_paciente_1  
FOREIGN KEY (FK_usuario_id_usuario)  
REFERENCES usuario (id_usuario)  
ON DELETE RESTRICT ON UPDATE RESTRICT;
```