

CENTRO UNIVERSITÁRIO UNIFACVEST

CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANDERSON TABALDI

**FisioTech: SOFTWARE DE ARMAZENAMENTO E GERENCIAMENTO DE  
INFORMAÇÕES PARA FISIOTERAPEUTAS**

LAGES

2021

ANDERSON TABALDI

**FisioTech: SOFTWARE DE ARMAZENAMENTO E GERENCIAMENTO DE  
INFORMAÇÕES PARA FISIOTERAPEUTAS**

Projeto apresentado à Banca Examinadora do  
Trabalho de Conclusão de Curso II de Ciência da  
Computação para análise e aprovação.

Orientação: Prof. Márcio José Sembay

Lages, SC \_\_\_/\_\_\_/2021, Nota\_\_\_\_\_

\_\_\_\_\_  
Márcio José Sembay

LAGES

2021

## RESUMO

As inovações tecnológicas estão sempre em alto desenvolvimento, com isso propiciar isso à área da saúde pode possibilitar diversos avanços e facilidades aos profissionais da área. Com isso nasce a importância da criação de um software que traga facilidade e agilidade ao dia a dia de profissionais como os fisioterapeutas, que baseiam seus tratamentos em informações colhidas desde o primeiro contato com o paciente, informações que com o passar dos atendimentos evoluem e se modificam. O FisioTech é um software desenvolvido com o objetivo de agregar e facilitar o dia a dia do Fisioterapeuta, com foco no armazenamento e gerenciamento de informações. Para desenvolvê-lo uma metodologia foi traçada, que se dividiu em duas vias, inicialmente foi realizada uma pesquisa buscando dados, informações e referências para fundamentar o desenvolvimento e criação do programa, e em um segundo momento foi realizado um desenho de como deveria se seguir a projeção do programa, e como ele seria desenvolvido, e então o desenvolvimento do software. Uma área com necessidades visíveis e com crescente destaque no mercado de trabalho, faz com que a utilização da tecnologia para armazenamentos agregue maior facilidade para prática clínica, com isso uma segurança maior evitando a perda de dados. O FisioTech traz como principal atrativo o foco nas individualidades do profissional de Fisioterapia, ele possibilita agilidade no processo de colher informações e segurança no armazenamento de dados, automatizando alguns processos e eliminando falhas de processos manuais.

**Palavras Chaves:** Tecnologia, Software, Fisioterapia, Desenvolvimento, Informação, Armazenamento.

## **ABSTRACT**

Technological innovations are always in high development, thus providing this to the health area can enable several advances and facilities for professionals in the area. With this, the importance of creating a software that brings ease and flexibility to the daily lives of professionals such as physiotherapists, who base their treatments on information collected from the first contact with the patient, information that evolves and becomes modify. FisioTech is a software developed with the objective of aggregating and facilitating the Physiotherapist's daily routine, with a focus on the storage and management of information. To develop it, a methodology was drawn up, which was divided into two ways, initially a survey was carried out seeking data, information and references to support the development and creation of the program, and in a second moment, a drawing was made of how it should follow. the design of the program, and how it would be developed, and then the development of the software. An area with visible needs and with growing prominence in the labor market, makes the use of technology for storage add greater ease to clinical practice, thus greater security, preventing data loss. FisioTech's main attraction is the focus on the individualities of the Physiotherapy professional, it allows for agility in the process of gathering information and security in data storage, automating some processes and eliminating failures in manual processes.

**Keywords:** Technology, Software, Physiotherapy, Development, Information, Storage.

## SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>7</b>
<b>LISTAS DE QUADROS .....</b>	<b>8</b>
<b>LISTA DE SIGLAS .....</b>	<b>9</b>
<b>1 INTRODUÇÃO .....</b>	<b>10</b>
1.2 JUSTIFICATIVA .....	10
1.3 OBJETIVOS .....	11
1.3.1 Objetivo Geral.....	11
1.3.2 Objetivos Específicos .....	11
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>11</b>
<b>3 PROCEDIMENTOS METODOLÓGICOS .....</b>	<b>16</b>
3.1 LEVANTAMENTO DE INFORMAÇÕES E DADOS .....	16
3.2 FERRAMENTAS E TECNOLOGIAS DE PROGRAMAÇÃO .....	16
3.3 CARACTERIZAÇÃO DA PESQUISA .....	17
3.3.1 Pesquisa documental.....	17
3.3.2 Pesquisa exploratória .....	17
<b>4 CRONOGRAMA.....</b>	<b>18</b>
<b>5 CARACTERIZAÇÃO DO PROJETO.....</b>	<b>18</b>
5.1 CADASTRO ELETRÔNICO DE PACIENTE .....	18
5.1.1 Filtro do cadastro .....	19
5.1.2 Criar novo cadastro .....	19
5.1.3 Editar cadastro.....	19
5.1.4 Cadastro eletrônico .....	19
5.1.5 Atendimentos .....	20
5.2 AGENDA DO PROFISSIONAL.....	22
5.3 RELATÓRIOS.....	23
5.3.1 Relatório de finanças.....	23
5.3.2 Relatório de evolução .....	23
5.3.3 Relatório por paciente .....	23
5.4 DADOS CADASTRAIS.....	23
<b>6 APRESENTAÇÃO DE DIAGRAMAS .....</b>	<b>24</b>

6.1	DIAGRAMA DE CASO DE USO GERAL.....	24
6.2	DIAGRAMA DE CLASSE GERAL.....	25
<b>7</b>	<b>FERRAMENTAS E METODOLOGIAS UTILIZADAS PARA DESENVOLVIMENTO.....</b>	<b>25</b>
7.1	C SHARP.....	25
7.2	SQL.....	27
<b>8</b>	<b>SOFTWARE .....</b>	<b>28</b>
8.1	TELAS DO SISTEMA .....	28
8.1.1	Tela de Login .....	28
8.1.2	Menu Principal.....	29
8.1.3	Cadastro de Pacientes.....	29
8.1.4	Atendimentos .....	30
8.1.5	Agenda .....	31
8.1.6	Relatórios .....	31
8.1.7	Relatório Financeiro.....	32
8.1.8	Relatório de evolução .....	32
8.1.9	Relatório do Paciente .....	33
<b>9</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>34</b>
<b>10</b>	<b>REFERÊNCIAS .....</b>	<b>35</b>
<b>11</b>	<b>ANEXOS .....</b>	<b>38</b>
	<b>ANEXO 1 – AUTORIZAÇÃO DE ENTREVISTA COM PROFISSIONAL .....</b>	<b>38</b>
	<b>ANEXO 2 – AUTORIZAÇÃO DE ENTREVISTA COM PROFISSIONAL .....</b>	<b>39</b>
	<b>ANEXO 3 – AUTORIZAÇÃO DE ENTREVISTA COM PROFISSIONAL .....</b>	<b>40</b>

## LISTA DE FIGURAS

Figura 1 - Código do processamento do cadastro e envio de alertas dos atendimentos.....	21
Figura 2. Código do processamento do cadastro e envio de alertas dos atendimentos.....	21
Figura 3. Código de configuração do HangFire.....	22
Figura 4. Diagrama de Caso de Uso Geral.....	24
Figura 5. Código de configuração do HangFire.....	25
Figura 6. Camadas do Projeto.....	26
Figura 7. Tela de Login.....	28
Figura 8. Menu Principal.....	29
Figura 9. Cadastro de Pacientes.....	29
Figura 10. Tela de Atendimentos.....	30
Figura 11. Tela de Evolução.....	30
Figura 12. Agenda do Dia e da Semana.....	31
Figura 13. Tela de Relatórios.....	31
Figura 14. Relatório de Finanças.....	32
Figura 15. Relatório de Evolução.....	33
Figura 16. Relatório Do Paciente.....	33

**LISTAS DE QUADROS**

Quadro 1 – Cronograma do TCC II.....	18
--------------------------------------	----



## **LISTA DE SIGLAS**

**C#** - C Sharp

**HTTP** - Hypertext Transfer Protocol

**SQL** - Structured Query Language

**IDE** - Ambiente de desenvolvimento integrado

# 1 INTRODUÇÃO

As inovações tecnológicas estão sempre em alto desenvolvimento, com isso propiciar esse desenvolvimento à área da saúde pode possibilitar diversos avanços e facilidades aos profissionais da área. A tecnologia agrega principalmente com a associação dos conceitos de tecnologia de informação e comunicação que integram em dispositivos com a finalidade de obter, armazenar e processar informações. (SANTOS, 2008).

Com isso nasce a importância da criação de um software que traga facilidade e agilidade ao dia a dia de profissionais como os fisioterapeutas, que baseiam seus tratamentos em informações colhidas desde o primeiro contato com o paciente, informações que com o passar dos atendimentos evoluem e se modificam, atualmente as informações são colhidas de forma manual e em alguns casos de maneira desorganizada.

A informação é um dos bens mais valioso de qualquer organização, e como tudo que é valioso, deve ser adequadamente guardado e protegido (WILLIAMS, 2001). De acordo com Araújo et al. (2013) por muito tempo a maioria das informações era armazenada e feita de maneira manual, e hoje com as novas tecnologias está cada vez mais informatizada, podendo auxiliar diversos setores da área da saúde.

Com isso o problema a ser discutido é a importância que tais softwares podem ter no desenvolvimento das melhorias no armazenamento e gerenciamento de informações, e nesse mesmo âmbito verificar as facilidades que eles apresentam, melhorando o dia a dia do Fisioterapeuta, agilizando tarefas que antes gastariam muito mais tempo e esforço.

O objetivo deste trabalho é apresentar e explicar como foi o desenvolvimento da criação de um software que pode ser utilizado no dia a dia do Fisioterapeuta, que diariamente lida com várias informações de diferentes pacientes.

## 1.2 JUSTIFICATIVA

Muitos profissionais da área da saúde já realizam suas atividades com o auxílio de softwares adequados a suas profissões, o que não é muito visto quando falamos do fisioterapeuta, isso de fato acontece por não existir um programa que atenda todas as exigências que se fazem necessárias. Com isso fica evidente a necessidade da criação de um software que

ajude a ter um controle total das informações do profissional para com seus pacientes, levando em conta as individualidades do profissional sendo algo que se encaixe e se adeque ao que a profissão exige.

### 1.3 OBJETIVOS

#### 1.3.1 Objetivo Geral

Desenvolver um software de armazenamento e gerenciamento de informações para profissionais fisioterapeutas.

#### 1.3.2 Objetivos Específicos

- a) Possibilitar maior facilidade para armazenar, visualizar e gerenciar informações dos pacientes.
- b) Auxiliar com o processo de agendamento de atendimentos automaticamente.
- c) Propiciar controle financeiro dos atendimentos realizados.
- d) Disponibilizar ao paciente maiores informações sobre o andamento dos atendimentos, e sobre futuro prognóstico.

## 2 FUNDAMENTAÇÃO TEÓRICA

Com base no tema e proposta de desenvolvimento de software é necessário entender que a nova realidade tecnológica, quanto aos procedimentos de coleta de informações, pode levar a dados de maior confiabilidade para gestão, geração de conhecimento e conseqüentemente, controle social. A atualização nos serviços de saúde vem sendo cada vez mais indispensável para os profissionais dessa área, e isso acaba sendo um processo ininterrupto para que o aprimoramento a assistência prestada ao indivíduo continue melhorando (BAKKER AR, 2007; CAVALCANTE RB et al., 2011).

Quando falamos de coleta de dados e informações, estamos citando também as fichas de avaliação fisioterapêuticas, e com isso observamos uma ausência de softwares com um olhar

mais focado, tendo uma visão baseada na Política Nacional de Saúde Funcional (CAVALCANTE RB et al., 2011).

Os softwares criados para serem utilizados na área da saúde têm por objetivo melhorar a assistência prestada, possibilitando uma maior disponibilidade de informações sobre o paciente e um acesso mais amplo aos dados armazenados, dessa forma sendo atualizados e de fácil entendimento. Com isso é possível apontar alguns benefícios que essa tecnologia traz, como o aumento da segurança dos usuários do serviço, possibilidade de comparar computacionalmente resultados de exames e a evolução do tratamento, auxiliando o diagnóstico e o plano assistencial, capacidade de rápido compartilhamento de informações quando necessário, geração de relatórios e indicadores de gestão e assistenciais tanto para fins epidemiológicos quanto estatísticos, colaborando para os processos de acreditação entre outros (LOURENÇÃO LG and FERREIRA JUNIOR CJ, 2016).

Uma técnica ainda muito utilizada no processo de coleta de informações na área da saúde é o uso do papel, sendo assim utilizados para informações, exames e prontuários. No entanto fica fácil entender que esse material atrasa alguns processos, podendo apresentar ilegibilidade, ambiguidade, perda de informação, multiplicidade de fichas do mesmo paciente, não padronização, dificuldade de acesso e, ainda necessita de um grande espaço para que seja arquivado (MAIA RS et al., 2016). Esse método acaba atrasando a coleta e muitas vezes duplicando o trabalho de registro rotineiro de dados (GUTIERREZ MA, 2011).

Atualmente a informática na área da saúde se faz essencial, agregando uma melhora da qualidade da informação, possibilitando aperfeiçoamento das atividades de armazenamento e recuperação de dados clínicos. O surgimento de novas tecnologias acompanha as mudanças sociais, e isso contribui para o desenvolvimento dos serviços e das ações de saúde, conseqüentemente, são importantes os estudos de instrumentos de trabalho que contribuam no processo de regulação da assistência, captando informações para a análise e adequação de ações (KALIL FILHO FA, 2008).

Justifica-se a utilização de um prontuário eletrônico em clínicas e estabelecimentos de fisioterapia no sentido de preencher a lacuna dos profissionais em relação à realização de avaliações completas, registros de dados, evolução e tratamento do paciente, obtenção de perfis epidemiológicos e substituição dos arquivos manuais por digitais, evitando entraves provocados pela dificuldade de leitura de fichas de avaliação, padronizando a informação (FRONZA CF AND OSÓRIO FCA, 2008).

Um sistema que possui uma boa usabilidade acaba por ter maior chance de ser utilizado por profissionais da área da saúde, porque por sua vez auxilia os processos de atendimento (REUSS E. et al., 2007). A avaliação de interfaces tem por objetivo identificar, classificar e contabilizar problemas de usabilidade de um sistema computacional. Esta atividade pretende fazer recomendações a partir do resultado da avaliação para aprimorar a usabilidade do sistema. Sistemas com boa usabilidade diminuem a ocorrência de erros e contribuem para produtividade e satisfação do usuário (DIX A. et al., 2007).

Em 2008, Zen et al. utilizou conceitos de engenharia de usabilidade para melhorar a produtividade de um sistema de prontuário eletrônico do paciente, dessa forma o sistema pode ser mais eficaz e também melhor aceito pelos profissionais que o utilizaram.

Reuss E. et al (2007), trabalhou mostrando que a interface em sistemas de registros eletrônicos de pacientes, devem sugerir as práticas realizadas pelos diversos profissionais da saúde, com isso possibilitando um trabalho multidisciplinar conectado.

Meyer and Skiba (2007) indicam a necessidade de equipamentos móveis para aprimorar a eficiência de uso e satisfazer os usuários. Estes apresentam suas experiências em usabilidade de sistemas de enfermagem.

Uma das ferramentas utilizadas para criação de um software é o ambiente de desenvolvimento integrado da Microsoft, o Visual Studio, que é um painel de inicialização criativo que você pode usar para editar, depurar e compilar o código e, em seguida, publicar um aplicativo. Um IDE (ambiente de desenvolvimento integrado) é um programa repleto de recursos que pode ser usado por muitos aspectos do desenvolvimento de software. Além do editor e do depurador padrão fornecidos pela maioria dos IDEs, o Visual Studio inclui compiladores, ferramentas de preenchimento de código, designers gráficos e muitos outros recursos para facilitar o processo de desenvolvimento de software (MICROSOFT, 2019).

A linguagem C# é uma linguagem elegante, orientada a objeto e fortemente tipada, que permite que os desenvolvedores criem uma variedade de aplicativos robustos e seguros executados no .NET Framework e no .NET Core. Você pode usar C# para criar aplicativos de cliente do Windows, serviços Web XML, componentes distribuídos, aplicativos cliente-servidor, aplicativos de banco de dados e muito, muito mais. O Visual Studio fornece um editor de código avançado, designers de interface do usuário convenientes, depurador integrado e muitas outras ferramentas para facilitar o desenvolvimento de aplicativos com base na

linguagem C#. A sintaxe de C# é altamente expressiva, mas também é simples e fácil de aprender. A sintaxe do C# será instantaneamente reconhecível para qualquer pessoa familiarizada com C, C++ ou Java (MICROSOFT, 2015).

Quando falamos dessa linguagem é preciso lembrar que por ser uma linguagem orientada a objeto, o C# oferece suporte aos conceitos de encapsulamento, herança e polimorfismo. Todas as variáveis e métodos, incluindo o método Main, o ponto de entrada do aplicativo, são encapsulados em definições de classe. Uma classe pode herdar diretamente de uma classe pai, mas pode implementar qualquer quantidade de interfaces. Métodos que substituem métodos virtuais em uma classe pai exigem a palavra-chave `override` como uma forma de evitar uma redefinição acidental. Em C#, um `struct` é como uma classe simplificada; é um tipo alocado na pilha que pode implementar interfaces, mas não oferece suporte a herança (MICROSOFT, 2015).

A plataforma de desenvolvimento modular, open-source chamada .NET Core é utilizada para múltiplos sistemas operacionais e se fundamenta na linguagem C#. Com ela é possível criar aplicações multiplataformas para console e web e também bibliotecas de classes com C# (MICROSOFT, 2021).

No gerenciamento do banco de dados através da linguagem C# que foi utilizado e devemos citar é o Entity Framework Core que aparece sendo uma versão leve e extensível, de software livre. Esse framework pode servir como mapeador relacional de objeto (O/RM), permitindo que os desenvolvedores do .NET trabalhem com um banco de dados usando objetos .NET, ela também elimina a necessidade da maior parte do código de acesso a dados que normalmente precisa ser escrito, sendo compatível com vários mecanismos de banco de dados. (MICROSOFT, 2021).

A Web API é a estrutura que deixa mais fácil a criação de serviços HTTP (Hypertext Transfer Protocol), sendo uma plataforma ideal para a construção de aplicações RESTful no .NET Core. A Web API disponibiliza informações do servidor por chamadas HTTP, assim os dados retornados são processados no cliente que consome esta Web API, aumentando assim a confiança entre cliente e servidor (MICROSOFT,2021 and DEVMEDIA, 2018).

O Hangfire para .NET Core aparece trazendo a possibilidade de criar trabalhos que são executados apenas uma vez, ou com uma recorrência configurável, além de trabalhos que são criados por usuários para serem processados em background, ele se encaixa em qualquer tipo

de aplicativo .NET Framework ou .NET Core. Agrega também um dashboard que contém todas as informações dos trabalhos realizados pela ferramenta, que utiliza um banco de dados para armazenamento e gerenciamento de seus trabalhos.

A plataforma Windows Forms possibilita a utilização de clientes inteligentes, que são aplicativos graficamente ricos que são fáceis de implantar e atualizar, podem funcionar quando eles estão conectados ou desconectados da Internet e podem acessar recursos no computador local de maneira mais segura do que os aplicativos tradicionais. Sendo um conjunto de bibliotecas gerenciadas que simplificam as tarefas comuns de aplicativos como leitura e gravação no sistema de arquivos. Ao usar um ambiente de desenvolvimento como o Visual Studio, você pode criar aplicativos cliente inteligente do Windows Forms que exibem informações, solicitam entrada de usuários e se comunicam com computadores remotos em uma rede, assim como comunicações externas por meio de protocolos como HTTP (MICROSOFT, 2021).

Nessa plataforma, formulários são criados exibindo informações para o usuário, com interfaces que exibem dados ou aceita a entrada de novos dados, dessa forma quando um usuário executa alguma ação em seu formulário ou em um de seus controles, a ação gera um evento. O Windows Forms tem controles avançados de interface do usuário que emulam recursos em aplicativos de alta tecnologia, podendo criar barras de ferramentas e menus que contêm texto e imagens, exibir submenus e hospedar outros controles, como caixas de texto e caixas de combinação (MICROSOFT, 2021).

O SQL Server é um banco de dados muito popular da Microsoft, que leva o título de ser o melhor da categoria, possui em sua bagagem muitos anos de desenvolvimento e aprimoramento. O SQL Server é um banco de dados relacional, possui uma arquitetura distribuída de gerenciamento, chamada de Distributed Management Framework ou DMF, composta de objetos, serviços e componentes. Através dela, vários servidores podem ser gerenciados completamente a partir de qualquer local na rede (JOBSTRAIBIZER F, 2009).

O SQL Server aceita a linguagem de programação, o que justifica seu nome. SQL ou Structured Query Language é uma linguagem declarativa para banco de dados relacional, sendo que muitas das suas características originais foram inspiradas na álgebra. Atualmente, a linguagem SQL é um grande padrão adotado pela maioria dos bancos de dados; tendo em vista sua simplicidade e facilidade de uso. Ela se diferencia de outras linguagens de consulta a banco de dados, pois uma consulta SQL especifica a forma do resultado e não o caminho para chegar

até ele. É uma linguagem declarativa em oposição a outras linguagens procedurais, nas quais é necessário informar o meio de chegar à informação (JOBSTRAIBIZER F, 2009).

### **3 PROCEDIMENTOS METODOLÓGICOS**

A metodologia traçada se dividiu em duas vias, inicialmente foi realizada uma pesquisa buscando dados, informações e referências para fundamentar o desenvolvimento e criação do programa, em um segundo momento foi realizado um desenho de como deveria se seguir a projeção do programa, e como ele seria desenvolvido. Foi realizado um estudo de caso, através de entrevistas com profissionais da área de fisioterapia com o intuito de sanar dúvidas, e verificar as necessidades e temas que o software deveria atender. Segundo Yin (2001) o estudo de caso é um método estratégico de pesquisa que permite compreender todas as abordagens específicas da coleta e análise de dados

#### **3.1 LEVANTAMENTO DE INFORMAÇÕES E DADOS**

Para obter esses dados foi realizada uma busca literária, propiciando uma fundamentação do desenvolvimento desse projeto. Esse tipo de busca possibilita o acesso a informações de diversos estudos anteriormente publicados, trazendo conclusões mais abrangentes quanto ao assunto. Uma busca por artigos foi realizada nas bases de dados Scielo, Lilacs, Medline e Google Acadêmico, durante o período de março a maio de 2021. Foram selecionados artigos e referências originais com publicação entre os anos de 2005 a 2020, nas línguas portuguesa e inglesa, que dissertassem sobre a atuação da tecnologia na área da saúde, e sobre softwares desenvolvidos com objetivos parecidos aos traçados nesse projeto. Essas informações ajudaram a definir quais programas seriam utilizados para o desenvolvimento do software

#### **3.2 FERRAMENTAS E TECNOLOGIAS DE PROGRAMAÇÃO**

Foram definidas duas ferramentas para o desenvolvimento do software, o Microsoft SQL Server Management Studio 2018 para gerenciamento do banco de dados SQL Server Express 2018, sendo usado como banco de dados para armazenar dados no servidor, e o Microsoft Visual Studio 2019 para o desenvolvimento do sistema com C# como linguagem de



programação, utilizando a tecnologia .NET Core 3.1, para codificação do lado servidor, além da linguagem C# com a tecnologia Windows Forms para a codificação do software apresentado ao usuário (lado cliente).

Para a execução do cadastro e alerta automático de atendimentos dos pacientes no lado servidor, foi utilizado o Framework Hangfire.AspNetCore 1.7.19 em conjunto a API .NET Core 3.1.

Este sistema foi desenvolvido para sistemas operacionais (S.O.) da plataforma Windows, no qual irá funcionar conforme o esperado caso este atenda os devidos pré-requisitos do sistema e das tecnologias utilizadas.

### 3.3 CARACTERIZAÇÃO DA PESQUISA

#### 3.3.1 Pesquisa documental

Inicialmente uma pesquisa documental foi realizada com o intuito de buscar fontes que ajudassem a criar uma problemática com real necessidade de solução, com fundamentação e contribuição de diversos autores, justificativas e objetivos para o desenvolvimento desse projeto foram devidamente entendidas. Uma pesquisa documental tem muita importância para realizar uma busca profunda não se baseando somente em material impresso disponíveis em bibliotecas e sim em diversos locais, tendo o objetivo de descrever e comparar os métodos de trabalho, comportamentos, entre outras características da problemática.

#### 3.3.2 Pesquisa exploratória

Quando a pesquisa é realizada em um campo onde o pesquisador possui pouco conhecimento a pesquisa exploratória se faz necessária (CERVO; BERVIAN; DA SILVA, 2007, p.61). Com base nisso três fisioterapeutas foram entrevistados e apresentaram suas visões e necessidades, mostrando a temática de trabalho, como realizam o cadastro de novos pacientes, agendamento de atendimentos, controle financeiro e evolução do tratamento dos pacientes.

## 4 CRONOGRAMA

Cronograma - ano 2021

<b>Atividades</b>	<b>MAR</b>	<b>ABR</b>	<b>MAI</b>	<b>JUN</b>	<b>JUL</b>
Busca de artigos em base de dados	X	X			
Fundamentação Teórica		X			
Desenvolvimento		X			
Consultoria com profissionais da área para coleta de dados		X	X		
Desenvolvido a base de conhecimento do sistema		X	X		
Organização dos dados encontrados			X		
Desenvolvimento do Software		X	X		
Revisão completa			X		
Produção dos Resultados			X		
Entrega TCC à coordenação				X	
Defesa TCC					X

**Quadro 1** – Cronograma do TCC II.

## 5 CARACTERIZAÇÃO DO PROJETO

### 5.1 CADASTRO ELETRÔNICO DE PACIENTE

Segundo informações colhidas através das entrevistas com os profissionais da área de fisioterapia, foi notado que cada profissional tem seu método para condução dos cadastros de seus pacientes. Levando isso em consideração, o cadastro foi desenvolvido para anexar as informações básicas dos pacientes, seus diagnósticos, objetivos, conduta de tratamento, além dos valores cobrados ao paciente, e também uma agenda semanal dos atendimentos.

### 5.1.1 Filtro do cadastro

Está disponível no cadastro eletrônico de pacientes a listagem de todos os pacientes cadastrados, com um campo para pesquisa por nome do paciente e uma opção para exibir ou não os pacientes cujo atendimentos já foram encerrados.

### 5.1.2 Criar novo cadastro

Ao criar um novo cadastro para o paciente, é possível informar seus dados pessoais básicos, informações referentes à cobrança dos atendimentos, informando o valor cobrado e o tipo da cobrança (mensal ou individual por atendimento), além dos diagnósticos, objetivos, conduta de tratamento e agenda de atendimentos semanal.

Cadastrando a agenda semanal, é possível informar em quais dias da semana os atendimentos serão realizados, além do horário inicial e final para cada atendimento. Esses dados serão usados pelo sistema para o cadastro automático de cada atendimento realizado, além de emitir um alerta minutos antes do atendimento ao profissional.

### 5.1.3 Editar cadastro

A edição do cadastro de pacientes consiste em alterar os dados cadastrados inicialmente para o paciente, além de poder desativá-lo, para considerar que este paciente já não será mais atendido.

### 5.1.4 Cadastro eletrônico

Podendo selecionar um ou mais cadastros de pacientes, a opção de excluir o cadastro removerá este do sistema.

### 5.1.5 Atendimentos

Nesta opção será aonde o profissional terá o controle dos atendimentos realizados e da evolução do paciente, podendo visualizar todos os atendimentos já realizados com este paciente ou filtrar atendimentos realizados em determinado mês do ano.

Quando um paciente tem sua agenda semanal de atendimentos registrada com pelo menos um dia de atendimento semanal, o cadastro do atendimento desse dia será feito de forma automática semanalmente pelo sistema, além de emitir um alerta por e-mail ao profissional 15 minutos antes do atendimento acontecer.

É possível também cadastrar um novo atendimento manualmente, em casos aonde houve um atendimento fora do planejado e que o sistema não consegue prever. Ao cadastrar, é possível informar a data e hora do atendimento, além da evolução ou observação individual do atendimento.

As funções de editar um atendimento já realizado e de excluir também estão disponíveis.

A codificação do cadastro automático e alertas dos atendimentos foi realizado levando em consideração o cadastro da agenda semanal do paciente. É realizado uma rotina percorrendo todos os usuários profissionais da fisioterapia cadastrados no sistema, e para cada usuário é buscado seus pacientes com a agenda semanal. Considerando o dia atual, caso a agenda do paciente tenha um cadastro para o dia atual, é validado se este atendimento já foi realizado e se um alerta já foi enviado, caso necessário, conforme código abaixo (Figura 1 e 2).

```

public async Task<string> ExecuteAlertQueueAsync()
{
    Log("Iniciando processamento");
    var dtNow = DateTime.Now;

    var usersAlerts = await _userRepository.RetrieveAsync(UserSpecification.RetrieveUserAlertsEnabled(), false, x => x.Clients);

    Log($"Buscou {usersAlerts.Count} usuários com alertas habilitados");

    foreach (var user in usersAlerts)
    {
        Log($"Iniciando processamento do usuário {user.UserName} com {user.Clients.Count} clientes.");
        foreach (var client in user.Clients)
        {
            Log($"Iniciando processamento do cliente {client.Name}");

            // Valida se tem atendimento hoje
            var attendanceToday = client.Recurrences.FirstOrDefault(p => p.WeekDay == (WeekDay)dtNow.DayOfWeek);

            // Caso não tenha, não precisa fazer nada
            if (attendanceToday == null)
            {
                Log($"Cliente não possui atendimento hoje, pulando para o próximo.");
                continue;
            }
        }
    }
}

```

**Figura 1.** Código do processamento do cadastro e envio de alertas dos atendimentos.

**Fonte:** Próprio Autor

```

// Caso tenha, valida se já criou o atendimento
var attendanceDate = new DateTime(dtNow.Year, dtNow.Month, dtNow.Day, 0, 0, 0);
var attendance = client.Attendances.FirstOrDefault(p => p.Date.Between(attendanceDate));

// Caso não tenha criado o atendimento, então cria
if (attendance == null)
{
    Log($"Atendimento ainda não criado, fazendo a criação...");

    attendance = new Attendance(client.ID, attendanceDate, attendanceToday.StartTime, attendanceToday.EndTime, string.Empty);
    Repository.Create(attendance);
}

// Valida se está na hora de mandar o e-mail e se já mandou
var startDate = new DateTime(dtNow.Year, dtNow.Month, dtNow.Day, attendanceToday.StartTime.Hour, attendanceToday.StartTime.Minute, 0);

Log($"Fazendo validação se deve enviar o email. O atendimento foi enviado?: {attendance.AlertHasSend}. " +
    $"Hora do atendimento: {startDate.ToString("dd/MM/yyyy HH:mm:ss")}, e hora atual: {dtNow.ToString("dd/MM/yyyy HH:mm:ss")}");

if (!attendance.AlertHasSend && startDate.AddMinutes(-30) <= dtNow)
{
    Log($"Enviando e-mail para {user.Mail}...");

    attendance.SetAlertSended();
    var emailHasSend = Email.Send(user.Mail, "Alerta de atendimento", BuildMailBody(user.FullName, client.Name, attendance.HourInitial));
    Log($"Email enviado?: {emailHasSend}");
}
}

await CommitAsync();

return bool.TrueString;
}

```

**Figura 2.** Código do processamento do cadastro e envio de alertas dos atendimentos.

**Fonte:** Próprio Autor

Para a execução deste processamento foi usado o Framework HangFire Server, que integrado juntamente à API .NET Core, realiza o enfileiramento do método selecionado (figura2) conforme o tempo configurado. A configuração para integração segue na imagem abaixo (figura 3).

```

0 referências | tabald98, há 57 dias | 1 autor, 2 alterações
public static class HangFireExtensions
{
    1 referência | tabald98, há 57 dias | 1 autor, 1 alteração
    public static void AddHangFire(this IServiceCollection services, IConfiguration configuration)
    {
        services.AddHangfire(p => p
            .SetDataCompatibilityLevel(CompatibilityLevel.Version_170)
            .UseSimpleAssemblyNameTypeSerializer()
            .UseRecommendedSerializerSettings()
            .UseActivator(new HangfireActivator(services.BuildServiceProvider()))
            .UseSqlServerStorage(configuration.GetSection("ConnectionStrings").GetSection("DbQueuesConnection").Value, new SqlServerStorageOptions()
            {
                QueuePollInterval = TimeSpan.FromSeconds(15),
                UsePageLocksOnDequeue = true,
                DisableGlobalLocks = true,
                EnableHeavyMigrations = false,
            }
        ));

        services.AddHangfireServer(p=>
        {
            p.ServerName = string.Format("{0}: Queues Manager", Environment.MachineName);
            p.WorkerCount = Environment.ProcessorCount * 5;
            p.StopTimeout = TimeSpan.FromSeconds(10);
        }
        ));
    }

    1 referência | tabald98, há 57 dias | 1 autor, 2 alterações
    public static void AddRecurringJobs(this IApplicationBuilder app)
    {
        RecurringJob.AddOrUpdate<IAttendanceAppService>("execute-recurrence-alert", p => p.ExecuteAlertQueueAsync(), Cron.MinuteInterval(15));
    }
}

```

**Figura 3.** Código de configuração do HangFire.

**Fonte:** Próprio Autor

## 5.2 AGENDA DO PROFISSIONAL

A agenda do profissional consiste em ter um calendário de seus atendimentos de fácil visualização, permitindo filtros por atendimentos no dia atual ou na semana atual. Este calendário é exibido conforme configurado a agenda semanal de atendimentos no cadastro eletrônico do paciente.

Filtrando pelos atendimentos do dia atual, é possível visualizar quais são os pacientes que tem ou terão atendimentos realizado no dia corrente, além do horário destes atendimentos.

Filtrando pelos atendimentos da semana atual, é possível visualizar quais são os pacientes que tem ou terão atendimentos realizado na semana, qual dia da semana cairá este atendimento, além do horário destes.

### 5.3 RELATÓRIOS

Conforme os levantamentos realizados durante as pesquisas, notou-se uma dificuldade do profissional da fisioterapia em controlar suas finanças, pois normalmente muitos pacientes são atendidos diariamente e um controle manual fica praticamente inviável. Também há uma dificuldade em comprovar os atendimentos junto ao paciente, para efetuar as devidas cobranças.

#### 5.3.1 Relatório de finanças

Na visualização dos relatórios é possível extrair um relatório das finanças, determinando um período inicial e final. Neste relatório exportado em PDF é possível visualizar quais pacientes foram atendidos no período, além da quantidade de atendimentos e valores cobrados individuais e totais do período.

#### 5.3.2 Relatório de evolução

Neste relatório, é possível selecionar qual paciente será extraído no relatório PDF. Ao escolher um paciente, os dados exibidos no relatório será os atendimentos realizados no período selecionado, podendo assim enviar estes dados ao paciente, para que este também consiga visualizar os atendimentos realizados e sua evolução.

#### 5.3.3 Relatório por paciente

Ao exportar o relatório do paciente, será possível selecionar individualmente o paciente necessário, aonde as informações apresentadas serão os dados cadastrais do paciente, assim o profissional poderá por exemplo, imprimir o relatório PDF conforme sua necessidade.

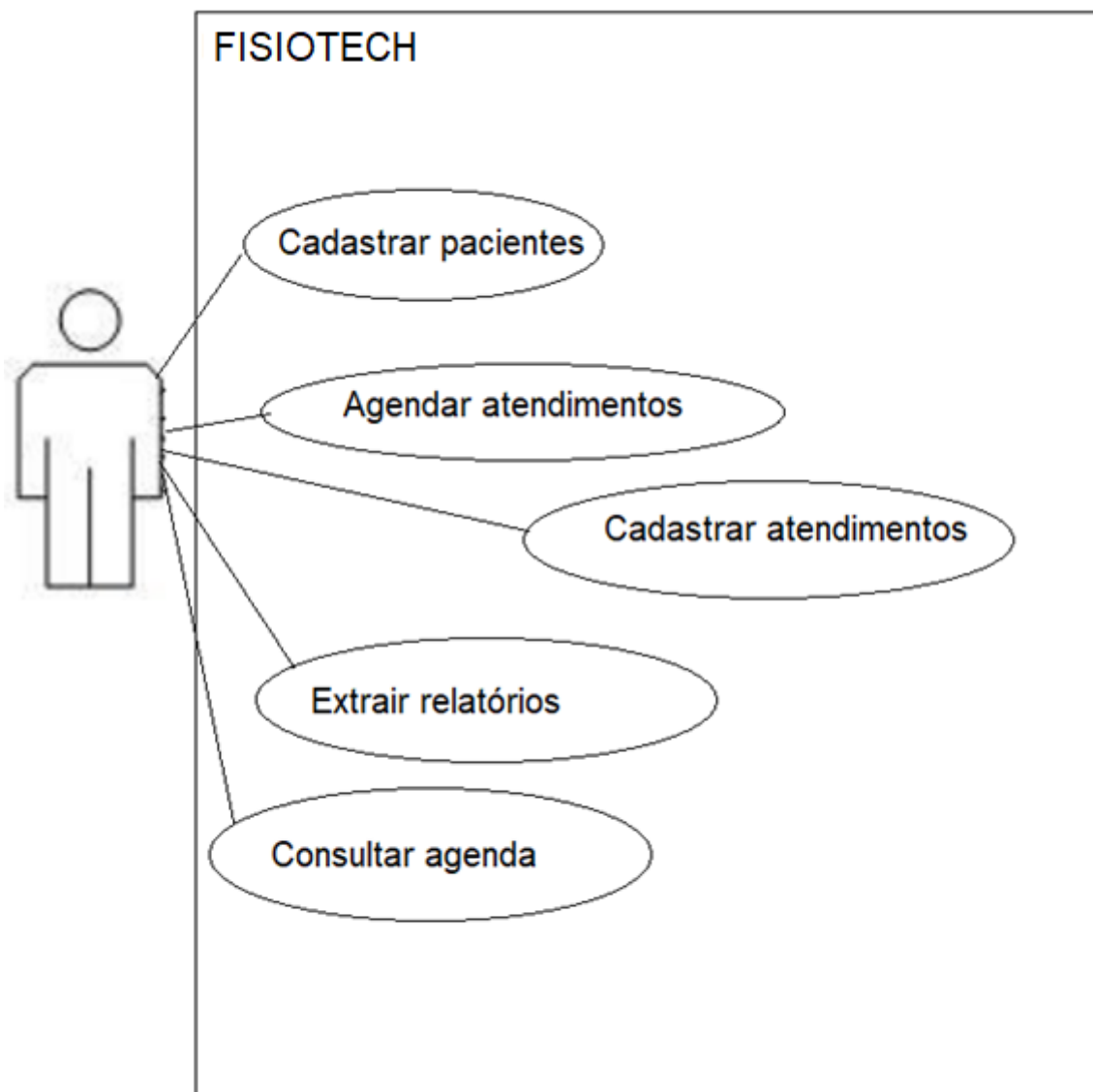
### 5.4 DADOS CADASTRAIS

Ao selecionar o nome do usuário logado, na tela principal do sistema, serão exibidos os dados do profissional da fisioterapia. Nesta tela poderá configurar seus dados pessoais, como o e-mail para recebimento dos alertas de atendimentos, alterar sua senha e também ativar ou desativar os alertas de atendimentos.

## 6 APRESENTAÇÃO DE DIAGRAMAS

### 6.1 DIAGRAMA DE CASO DE USO GERAL

Há apenas uma camada de acesso ao sistema, aonde o usuário poderá gerenciar seus pacientes. O gerenciamento dos usuários com acesso ao sistema é feito de forma manual através do cadastro no banco de dados.



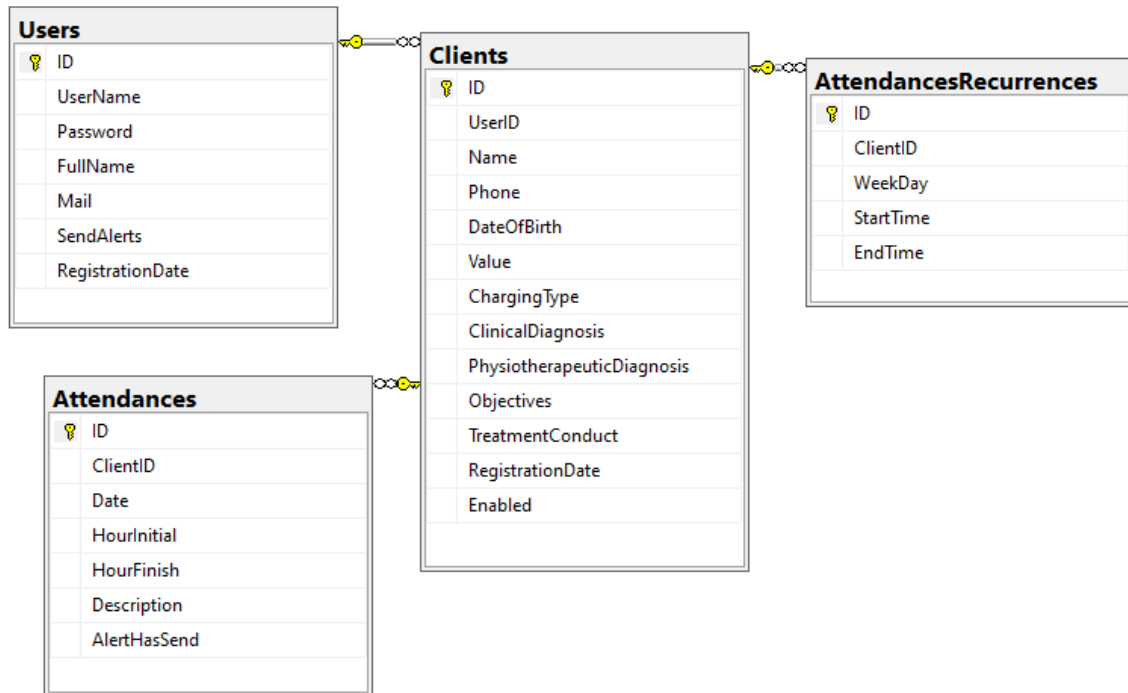
**Figura 4.** Diagrama de Caso de Uso Geral

**Fonte:** Próprio Autor



## 6.2 DIAGRAMA DE CLASSES GERAL

O Diagrama de classe será basicamente a estrutura do banco de dados e seus devidos relacionamentos.



**Figura 5.** Código de configuração do HangFire.

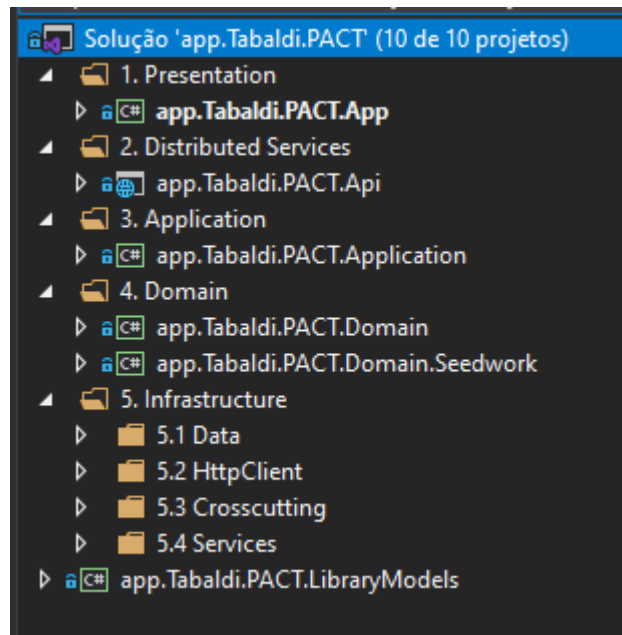
**Fonte:** Próprio Autor

## 7 FERRAMENTAS E METODOLOGIAS UTILIZADAS PARA DESENVOLVIMENTO

### 7.1 C SHARP

Para o desenvolvimento do FisioTech foi utilizado a linguagem C Sharp, criada pela Microsoft, relacionada a plataforma .NET é uma linguagem de programação orientada a objetos e baseia-se em conceitos de outras linguagens, principalmente C++ e JAVA, mesmo considerada recente, veem se destacando no mercado de trabalho devido a praticidade e inovação contida nela (ARAUJO, 2010).

A linguagem foi utilizada em níveis de camada:



**Figura 6.** Camadas do Projeto.

**Fonte:** Próprio Autor

a) Apresentação:

Nesta camada é aonde foi desenvolvido o lado cliente da aplicação, realizando a comunicação via API com o lado servidor, e também as telas de apresentação e interação com o usuário.

b) Serviços distribuídos

Nesta camada é aonde foi desenvolvido a API do servidor que recebe as chamadas HTTP dos seus clientes.

c) Aplicação

Nesta camada fica a lógica do sistema, como a atribuição aos objetos relacionados ao banco de dados, validações, gerenciamento automático dos atendimentos, entre outros.

d) Domínio

Nesta camada fica os objetos do sistema, como usuários e pacientes, bem como a regra de negócio do sistema.

e) Infraestrutura

Nesta camada fica a parte de comunicação do o banco de dados, utilizando o Entity Framework Core, bem como serviços comuns utilizados em todos os projetos.

## 7.2 SQL

O FisioTech foi desenvolvido de maneira que o mesmo tenha todas sua massa de dados armazenadas em um banco de dados, no qual foi utilizado linguagem SQL. A SQL ou Linguagem de Consulta Estruturada foi criada pela IBM Research, no início da década de 1970, para o protótipo de um sistema de banco de dados chamado System R (DATE, 2004, p. 71).

O SQL oferece além de uma linguagem de consulta, recursos para definir a estrutura dos dados, atualizar – incluir, excluir e alterar dados, especificar restrições de integridade e outros recursos mais (SILBERSCHATZ; KORTH; SUDARSHAN, 1999, p. 109).

Sua linguagem de fácil interpretação e edição, sua lógica não precisa de gerenciamento em sequência. É Open Source, atualizações e versões novas são outro ponto forte.

De acordo com SETZER e SILVA (2005) a linguagem SQL suporta comandos de criação e manipulação de tabelas, manipulação de objetos, manutenção de dados, realização de consultas, controle de acesso e administração geral do banco de dados, tornando-se assim ideal para desenvolvimento da aplicação e essencial para armamento e manipulação de dados que serão gerenciados através do FisioTech.

Para a comunicação entre C# ambiente servidor e o banco de dados, foi utilizado o framework Entity Framework Core, no qual cria e executa queries de forma automática de maneira que o SQL entenda. A sua utilização no C# é dada através da biblioteca System.Linq nativa C#, aonde é possível buscar, filtrar, editar e criar dados no banco de maneira simples em códigos C#.

## 8 SOFTWARE

### 8.1 TELAS DO SISTEMA

Para a interface do sistema, foi utilizado a tecnologia Windows Forms da linguagem C#, já integrados na IDE Visual Studio 2019, cuja autoria é da Microsoft.

#### 8.1.1 Tela de Login

O sistema conta com uma tela de login (Figura 7) para obter o controle de acesso para cada usuário conforme seus dados são disponibilizados. Cada usuário terá controle para seus pacientes individualmente, assim se faz necessário a identificação ao entrar no sistema.

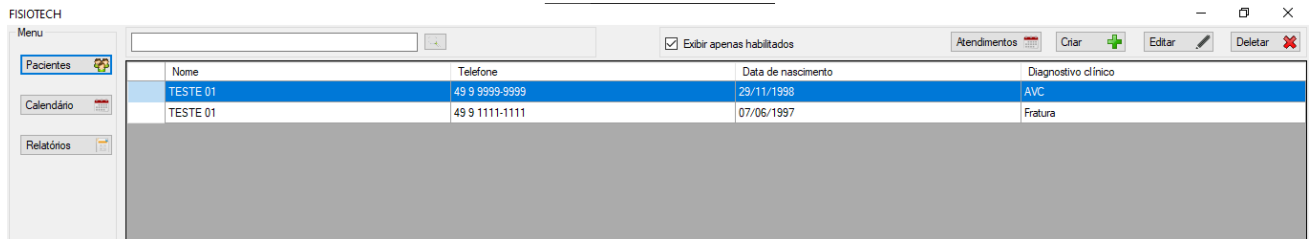


**Figura 7.** Tela de Login

**Fonte:** Próprio Autor

### 8.1.2 Menu Principal

Na tela principal (Figura 8), será exibido todos os pacientes cadastrados ao usuário logado. Também será possível interagir com as demais funções do sistema.



**Figura 8.** Menu Principal

**Fonte:** Próprio Autor

### 8.1.3 Cadastro de Pacientes

Nesta tela é possível cadastrar um novo paciente, com seus dados pessoais, clínicos e agenda dos atendimentos.

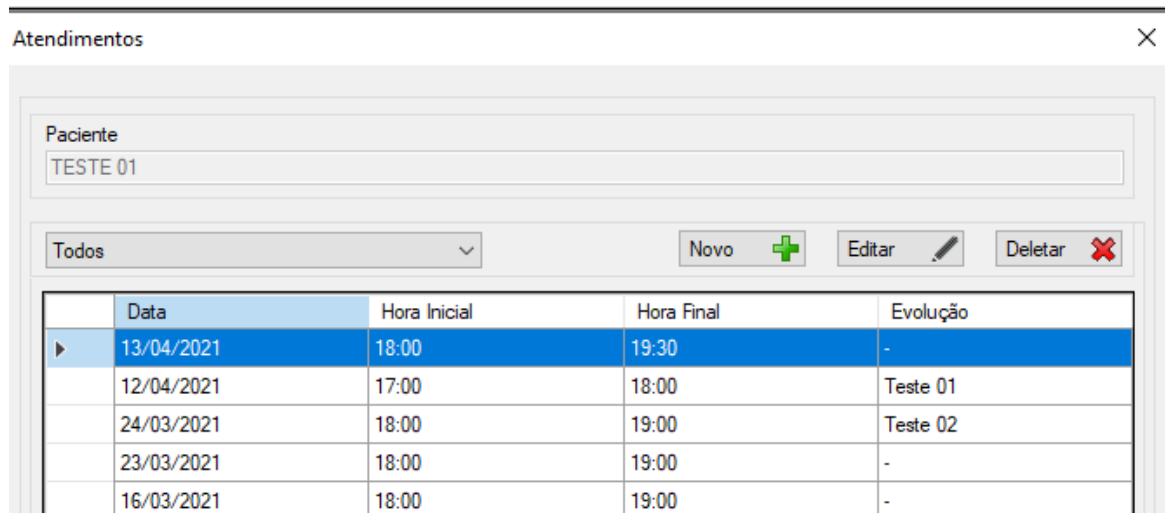
Dias	Das	Até
Segunda-feira:	17:52	17:52
Terça-feira:	17:52	17:52
Quarta-feira:	17:52	17:52
Quinta-feira:	17:52	17:52
Sexta-feira:	17:52	17:52
Sábado:	17:52	17:52
Domingo:	17:52	17:52

**Figura 9.** Cadastro de Pacientes

**Fonte:** Próprio Autor

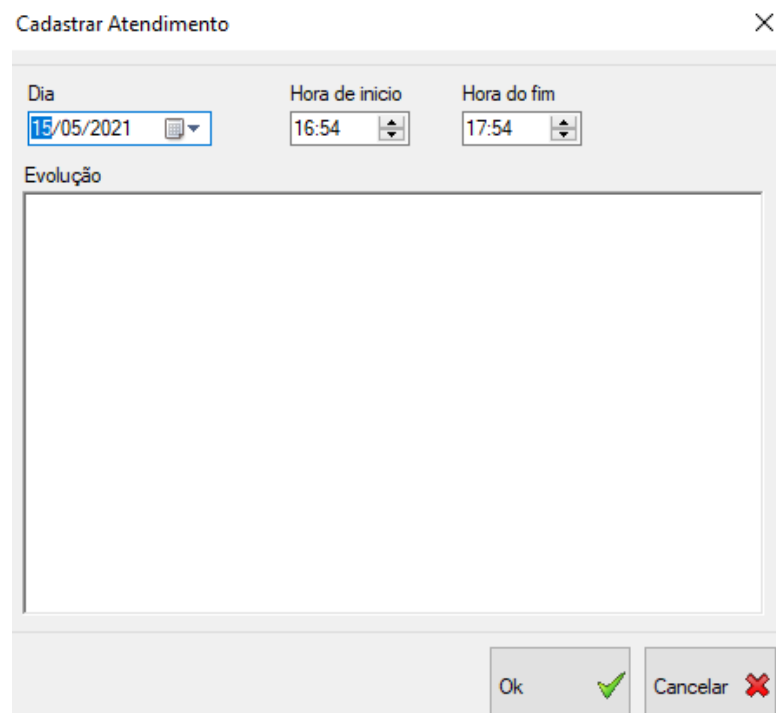
#### 8.1.4 Atendimentos

Essa é a tela de registro dos atendimentos individuais por paciente, aonde é possível visualizar, filtrar, cadastrar, editar e excluir.



**Figura 10.** Tela de Atendimentos

**Fonte:** Próprio Autor

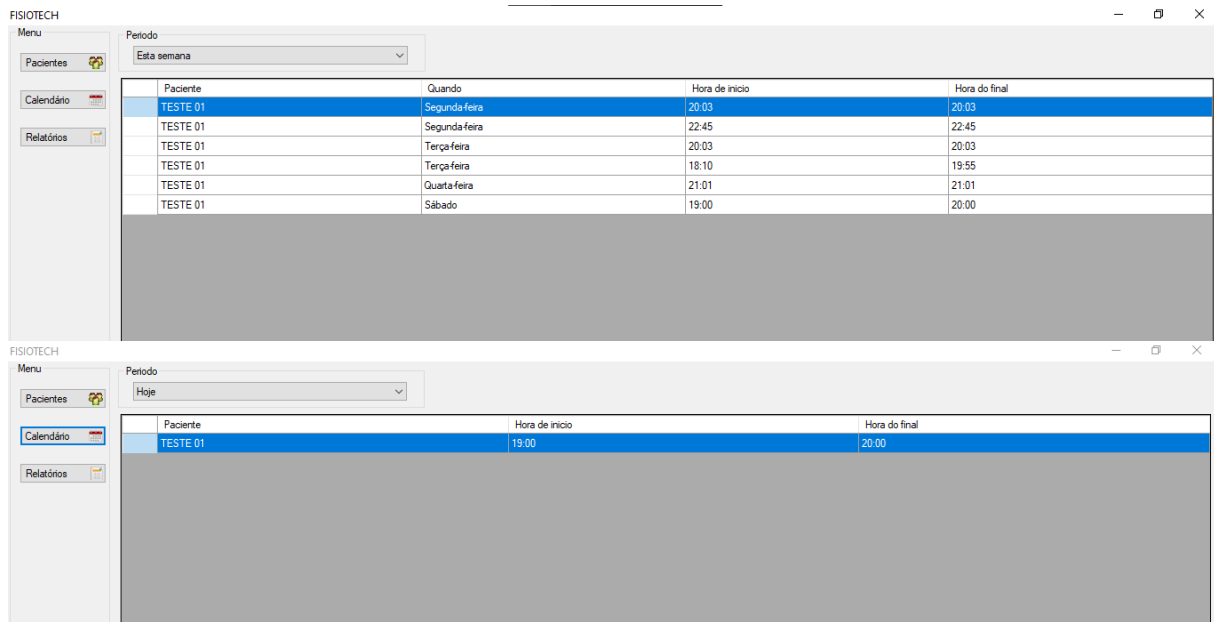


**Figura 11.** Tela de Evolução

**Fonte:** Próprio Autor

### 8.1.5 Agenda

A tela de agenda é aonde é possível visualizar quais são os atendimentos do dia atual e da semana atual.



**Figura 12.** Agenda do Dia e da Semana

**Fonte:** Próprio Autor

### 8.1.6 Relatórios

Nesta tela é possível selecionar qual será o relatório a ser extraído. Todos os relatórios são extraídos no formato PDF e salvos em um diretório que deve ser informado ao realizar a extração do relatório.

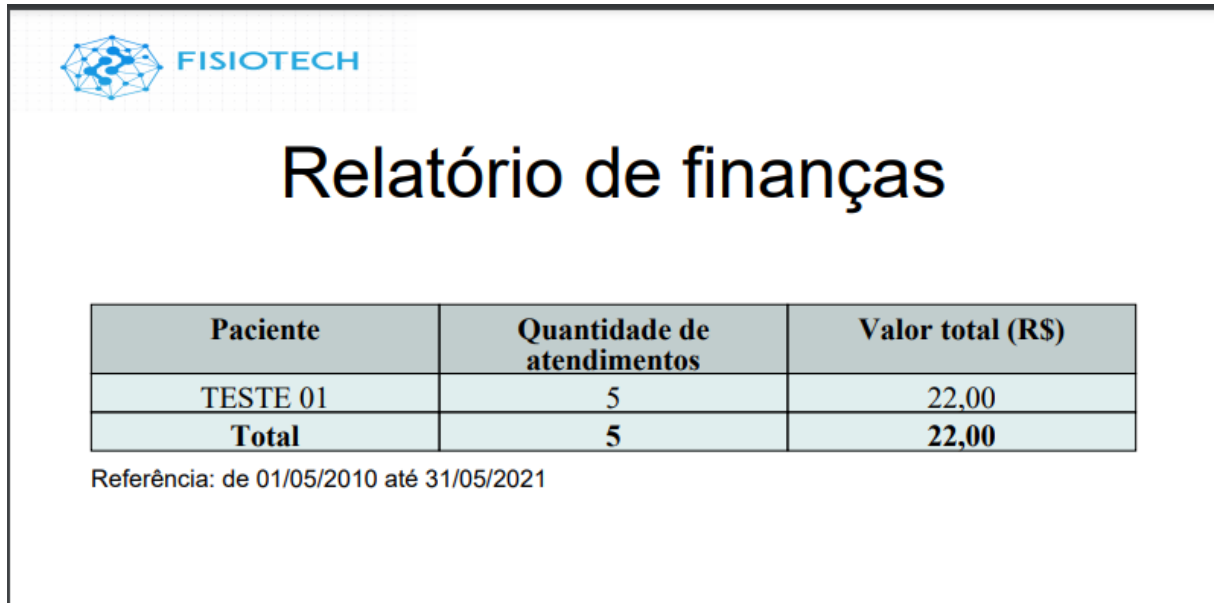


**Figura 13.** Tela de Relatórios

**Fonte:** Próprio Autor

### 8.1.7 Relatório Financeiro

Este é o relatório que sumariza todos os pacientes, seus atendimentos e os valores cobrados no período selecionado.



**Figura 14.** Relatório de Finanças

**Fonte:** Próprio Autor

### 8.1.8 Relatório de evolução

Neste relatório será visualizado os atendimentos do paciente selecionado no período, além de sua evolução.





FISIOTECH

## Relatório de evolução do(a) TESTE 01

Dia do atendimento	Hora inicial	Hora final	Evolução
16/03/2021	18:00	19:00	-
23/03/2021	18:00	19:00	-
24/03/2021	18:00	19:00	Teste 02
12/04/2021	17:00	18:00	Teste 01
13/04/2021	18:00	19:30	-

**Figura 15.** Relatório de Evolução

**Fonte:** Próprio Autor

### 8.1.9 Relatório do Paciente

Este é o relatório que sumariza as informações de cadastro do paciente.



FISIOTECH

## Relatório do paciente TESTE 01

**Data de nascimento:**

- 29/11/1998

**Telefone:**

- 49 9 9999-9999

**Diagnóstico clínico:**

- AVC

**Diagnóstico fisioterapêutico:**

- Teste

**Objetivos:**

- Teste

**Conduta de tratamento:**

- Teste

**Figura 16.** Relatório Do Paciente

**Fonte:** Próprio Autor

## **9 CONSIDERAÇÕES FINAIS**

A Fisioterapia é uma profissão relativamente nova no mercado de trabalho, tendo apenas 52 anos de regulamentação e reconhecimento no Brasil, mas vem em uma crescente que surpreende a cada dia, especialmente em 2021, ano onde os profissionais da área foram de suma importância para o sistema básico de saúde. Segundo a revista Exame, a profissão cresceu 725% na pandemia, números nunca vistos antes. O Fisioterapeuta é profissional de primeiro contato, podendo ele ser autônomo ou não, de qualquer maneira é ele quem tem todo o controle de seus atendimentos e de como sua conduta de tratamento virá a seguir, com isso muitas vezes o profissional fica sobrecarregado de informações, e o armazenamento delas na maioria das vezes e através de relatórios impressos e grande número de documentação, o que pode levar a perda de dados.

Uma área com necessidades visíveis e com crescente destaque no mercado de trabalho, faz com que a utilização da tecnologia para armazenamentos agregue maior facilidade para prática clínica, com isso uma segurança maior evitando a perda de dados. O FisioTech traz como principal atrativo o foco nas individualidades do profissional de Fisioterapia, ele possibilita agilidade no processo de colher informações e segurança no armazenamento de dados, automatizando alguns processos e eliminando falhas de processos manuais.

## 10 REFERÊNCIAS

A. DIX, J. FINLAY, G. ABOWD, AND R. BEALE. *Human-Computer-Interaction*. Pearson Education, 2004.

ARAÚJO, Bruno Gomes de et al. Processo de certificação de sistemas de registro eletrônico de saúde no brasil: uma abordagem abrangente e os principais desafios. *Revista Brasileira de Inovação Tecnológica em Saúde*, Natal, RN, v. 3, n. 3, p.11-26, 2013.

B. MEYER AND D. SKIBA. Usability-testing healthcare software with nursing informatics students in distance education: A case study. *HCI and Usability for Medicine and Health Care*, 1(1):447–452, 2007.

BAKKER AR. The need to know the history of the use of digital patient data, in particular the EHR. *Int J Med Inform*. 2007.

CAVALCANTE RB, SILVA PC, FERREIRA MN. Sistemas de informação em saúde: possibilidades e desafios. *Rev enferm UFSM*. 2011.

CERVO, Amado Luiz; BERVIAN, Pedro Alcino; DA SILVA, Roberto. *Metodologia científica*. 6ª. ed. São Paulo: Prentice Hall, 2007.

DATE, C. J. *INTRODUÇÃO A SISTEMAS DE BANCOS DE DADOS*. 8. ed. Rio de Janeiro: Elsevier, 2004.

DEVMEDIA. *Introdução ao ASP.NET Web API*. Acesso em 15 fevereiro 2021.

E. R. et al. Nurses's working practices: What can we learn for designing computerised patient record system? *HCI and Usability for Medicine and Health Care*, 1(1):55–82, 2007.

E. ZEN, R. NUNES, M. A. OLIVEIRA, M. DA SILVA FONSECA, AND S. PEREIRA. Melhora da produtividade no uso do prontuário eletrônico do paciente utilizando a engenharia de usabilidade. In *Workshop de Informática Médica*, pages 111–120, 2008.

FLÁVIA JOBSTRAIBIZER. Guia Profissional Microsoft SQL Server 2008 - São Paulo: Digerati Books, 2009. 128 p. - (v.1).

FRONZA CF, OSÓRIO FCA. O prontuário eletrônico como instrumento de avaliação e apoio à decisão clínica fisioterapêutica. 2008.

GUTIERREZ MA. Sistemas de Informação Hospitalares: progressos e avanços. JHealth Inform. 2011;3(2):115-7.

KALIL FILHO FA. Protocolo eletrônico de coleta de dados clínicos em fisioterapia respiratória para doenças pulmonares. [Dissertação]. Curitiba: Universidade Federal do Paraná; 2008.

LOURENÇÃO LG; FERREIRA JUNIOR CJ. Implantação do prontuário eletrônico do paciente no Brasil. Enfermagem Brasil. 2016;15(1):44-53.

MAIA RS, SANTOS LMA, MENDES NS, BARBOSA AS, OLIVEIRA RFR. Uso do prontuário eletrônico em Estratégia de Saúde da Família/Montes Claros—relato de experiência. 2016.

MICROSOFT. Bem-vindo ao IDE do Visual Studio, 2019. Disponível em: <https://docs.microsoft.com/pt-br/visualstudio/get-started/visual-studio-ide?view=vs-2019>. Acesso em: 18 de abril de 2021.

MICROSOFT. Introdução à língua C# e ao Quadro .NET, 2015. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>. Acesso em: 18 de abril de 2021.

MICROSOFT. Introdução ao .NET, 2020. Disponível em: <https://docs.microsoft.com/pt-br/ef/core/>. Acesso em: 20 de janeiro de 2021.

MICROSOFT. Entity Framework Core. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/core/introduction>. Acesso em: 20 de janeiro de 2021.

MICROSOFT. Learn About ASP.NET Web API. Acesso em 15 de fevereiro de 2021.

SANTOS, M. F. (2008) - Tecnologias da informação e comunicação. In SLIDESHARE - Slideshare. (27 de dezembro de 2008).

WILLIAMS, P. A. Information Security Governance, Information Security Technical Report. v. 6, n. 3 p.60-70, 2001.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. SISTEMA DE BANCO DE DADOS. 3. ed. São Paulo: Makron Books, 1999.

YIN, Roberto K. Estudo de caso: planejamento e métodos. 2ª Ed. Porto Alegre: Bookman, 2001.

**11 ANEXOS****ANEXO 1 – AUTORIZAÇÃO DE ENTREVISTA COM PROFISSIONAL**

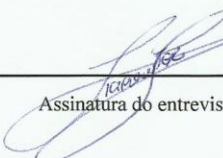
Fisioterapeuta Tiago Kuster Lopes



## AUTORIZAÇÃO

Eu TIAGO KUSTER LOPES, abaixo assinado(a), autorizo Anderson Tabaldi, estudante de Ciência da Computação, do Centro Universitário Unifacvest, a utilizar as informações por mim prestadas, para a elaboração de seu Trabalho de Conclusão de Curso.

Lages, 20 de MAIO de 2021.

  
Assinatura do entrevistado

**Dr. Tiago Kuster Lopes**  
Fisioterapeuta  
CREFITO 3011&31-F

**ANEXO 2 – AUTORIZAÇÃO DE ENTREVISTA COM PROFISSIONAL**

Fisioterapeuta Patrícia Caroline de Oliveira Lencina



## AUTORIZAÇÃO

Eu Patrícia Caroline de O. Lencina, abaixo assinado(a), autorizo Anderson Tabaldi, estudante de Ciência da Computação, do Centro Universitário Unifacvest, a utilizar as informações por mim prestadas, para a elaboração de seu Trabalho de Conclusão de Curso.

Lages, .....10..... de .....Maio..... de 2021.

Assinatura do entrevistado

**Dra. Patrícia Caroline**  
Fisioterapeuta  
CREFITO 10/309202-F

**ANEXO 3 – AUTORIZAÇÃO DE ENTREVISTA COM PROFISSIONAL**

Fisioterapeuta Gabriela de Liz

**AUTORIZAÇÃO**

Eu Gabriela de Liz, abaixo assinado(a), autorizo Anderson Tabaldi, estudante de Ciência da Computação, do Centro Universitário Unifacvest, a utilizar as informações por mim prestadas, para a elaboração de seu Trabalho de Conclusão de Curso.

Lages, 20 de Maio de 2021.



Assinatura do entrevistado