

**UNIVERSIDADE BRASIL  
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOENGENHARIA  
CAMPUS ITAQUERA**

**RAMÃO DE SOUZA BALTA**

**SISTEMA DE COMPARTILHAMENTO DE HIPÓTESE DE  
DIAGNÓSTICO**

**DIAGNOSTIC HYPOTHESIS SHARING SYSTEM**

São Paulo – SP

2023

## PROGRAMA DE PÓS-GRADUAÇÃO EM BIOENGENHARIA

### RAMÃO DE SOUZA BALTA

#### SISTEMA DE COMPARTILHAMENTO DE HIPÓTESE DE DIAGNÓSTICO

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Bioengenharia da Universidade Brasil, como parte dos requisitos necessários para obtenção do título de Mestre em Bioengenharia.

Prof. Dr. Daniel Souza Ferreira Magalhães  
**Orientador**

Prof. Dr. José Carlos Cogo  
**Coorientador**

**Área de concentração:** Reabilitação

**Linha de pesquisa:** Inovação em reabilitação

São Paulo –SP  
2023

**Ficha catalográfica elaborada pelo Sistema de Bibliotecas da Universidade Brasil,  
com os dados fornecidos pelo (a) autor (a).**

B158s      BALTA, Ramão de Souza.  
              Sistema de compartilhamento de hipótese de diagnóstico / Ramão de  
              Souza Balta -- São Paulo: Universidade Brasil, 2023.  
              48 f.: il. color.

              Dissertação de Mestrado defendida no Programa de Pós-graduação  
              do Curso de Bioengenharia da Universidade Brasil.  
              Orientação: Prof. Dr. Daniel Souza Ferreira Magalhães.  
              Coorientação: Prof. Dr. José Carlos Cogo.

              1. Diagnóstico compartilhado. 2. Radiologia. 3. Tomografia. 4.  
              Software de hipótese de diagnóstico. I. Magalhães, Daniel Souza  
              Ferreira. II. Cogo, José Carlos. III. Título.

CDD 610.28



UNIVERSIDADE  
BRASIL

TERMO DE APROVAÇÃO

RAMÃO DE SOUZA BALTA

"SISTEMA DE COMPARTILHAMENTO DE HIPÓTESE DE DIAGNÓSTICO"

Dissertação aprovada como requisito parcial para obtenção do título de **Mestre no Programa de Pós-Graduação em Bioengenharia** da Universidade Brasil, pela seguinte banca examinadora:

Prof.(a) Dr.(a) Daniel Souza Ferreira Magalhães (presidente-orientadora)

Prof.(a). Dr.(a). Alessandro Marco Hakme da Silva (UNIVERSIDADE BRASIL)

Prof.(a) Dr.(a) Fedul Ferrari Rodor (UNIVERSIDADE FEDERAL DE ITAJUBÁ)

São Paulo, 24 de março de 2023

Presidente da Banca Prof.(a) Dr.(a). Daniel Souza Ferreira Magalhães

Houve alteração do Título: sim ( ) não (X):

---

---

---



### Termo de Autorização

#### **Para Publicação de Dissertações e Teses no Formato Eletrônico na Página WWW do Respectivo Programa da Universidade Brasil e no Banco de Teses da CAPES**

Na qualidade de titular(es) dos direitos de autor da publicação, e de acordo com a Portaria CAPES no. 13, de 15 de fevereiro de 2006, autorizo(amos) a Universidade Brasil a disponibilizar através do site <http://www.universidadebrasil.edu.br>, na página do respectivo Programa de Pós-Graduação *Stricto Sensu*, bem como no Banco de Dissertações e Teses da CAPES, através do site <http://bancodeteses.capes.gov.br>, a versão digital do texto integral da Dissertação/Tese abaixo citada, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira.

A utilização do conteúdo deste texto, exclusivamente para fins acadêmicos e científicos, fica condicionada à citação da fonte.

Título do Trabalho: **“SISTEMA DE COMPARTILHAMENTO DE HIPÓTESE DE DIAGNÓSTICO”**

Houve alteração do Título: sim ( ) não (X):

\_\_\_\_\_

\_\_\_\_\_

Autor(es):

Discente: **Ramão de Souza Balta**

Assinatura: RS Balta

Orientador(a): **Prof.(a) Dr.(a) Daniel Souza Ferreira Magalhães**

Assinatura: DS Magalhães

Coorientador(a):

Assinatura: Dr. José Carlos Cojo

Data: 24/03/2023

**Aos pesquisadores, em especial aos da área da saúde**

Que provaram o seu valor durante a pandemia

**Dedico**

## **AGRADECIMENTOS**

Agradeço a Deus pelo dom da vida e por tudo que nos proporcionou até aqui e, sobretudo suas bênçãos futuras para todos nós.

Agradeço a meus pais Milton da Silva Balta, Maria de Souza Lopes e Romilda da Cruz Balta que sempre estiveram presentes no meu aprendizado.

Agradeço aos meus irmãos e amigos pelo incentivo em continuar sempre na busca pelo conhecimento.

Agradeço aos colegas de mestrado que demonstraram bravura durante esse longo período de pandemia.

Agradeço ao meu amigo e professor Doutor José Carlos Cogo pelo convite em fazer o mestrado, suas palavras me convenceram de que era o caminho.

Agradeço ao meu orientador, professor Doutor Daniel Magalhães pela ajuda em todas as fases do desenvolvimento deste trabalho.

E por fim, agradeço a gentileza de cada professor deste mestrado em dividir o conhecimento conosco.

*“Fazer o bem sem olhar a quem”*  
(Autor desconhecido)

## RESUMO

Um software com ferramentas para compartilhar e discutir hipóteses de diagnóstico permite a integração multidisciplinar entre os profissionais da odontologia, servindo como um elo que liga o radiologista aos demais especialistas. Este trabalho apresenta o desenvolvimento de um Sistema de Compartilhamento de Hipótese de Diagnóstico, uma ferramenta que permite publicar imagens de um determinado caso e receber ajuda dos membros da comunidade científica e assim formular um diagnóstico preciso. A base de desenvolvimento é a linguagem C#, com o ASP.NET para trazer recursos de aplicação web ao sistema e Javascript para interação. Um dos pontos fortes do software para compartilhamento de hipótese de diagnóstico é a possibilidade de manter sigilo do autor das perguntas e a forma livre do compartilhamento com outros profissionais e assim, resolver juntos um dado problema com a melhor solução e o segundo ponto forte é contemplar a Lei Geral de Proteção de Dados.

**Palavras-chave:** Diagnóstico compartilhado. Radiologia. Tomografia. Software de hipótese de diagnóstico.

## **ABSTRACT**

A software with tools to share and discuss diagnostic hypotheses allows multidisciplinary integration between dentistry professionals, serving as a link connecting radiologist to other specialists. This work presents the development of a Diagnosis Hypothesis Sharing System, a tool that allows publishing images of a given case and receiving help from members of the scientific community and thus formulating an accurate diagnosis. The development base is C# programming language, with ASP.NET to bring web application capabilities to the system and Javascript to manipulate user events. One strong point of the software is the possibility of keeping the author of the questions confidential and the free way of sharing with other professionals and thus solving a given problem together finding the best solution, the General Data Protection Law is applied in the software.

**Keywords:** Shared diagnosis. Radiology. Tomography. Diagnostic hypothesis software.

## **DIVULGAÇÃO E TRANSFERÊNCIA DE CONHECIMENTO**

Esta dissertação mostra as ferramentas necessárias para desenvolver um software web para postagem de imagens e com recursos de comentários sobre as anomalias ou divergências encontradas. O passo a passo do desenvolvimento não está no escopo deste trabalho.

## LISTA DE FIGURAS

Figura 1- Fluxograma de metodologia de desenvolvimento do software .....	26
Figura 2 – Caso de Uso.....	28
Figura 3 - Diagrama de Entidade Relacionamento .....	30
Figura 4 – Diagrama de Classe.....	31
Figura 5 - Projeto no Visual Studio.....	32
Figura 6 – String de conexão .....	33
Figura 7 – Exemplo de classe de dados .....	34
Figura 8 – Exemplo de classe de lógica de negócio .....	35
Figura 9 – Adicionar Web Form.....	36
Figura 10 – Exemplo de arquivo Aspx.....	37
Figura 11 – Exemplo de <i>code behind</i> .....	37
Figura 12 – Inserção de componente do Aspx.....	38
Figura 13 – Manipulação de eventos do componente Aspx.....	38
Figura 14 – Exemplo de código javascript.....	39
Figura 15 – Postagem no Sistema de Hipótese de diagnóstico .....	40
Figura 16 – Zoom no visualizador de Imagens .....	41
Figura 17 – Exemplo de uso em outras áreas.....	42
Figura 18 – Postagem no grupo de discussões .....	43

## LISTA DE TABELAS

Tabela 1 – Caso de uso “Registra-se” .....	28
Tabela 2 – Caso de uso “Loga” .....	29
Tabela 3 – Caso de uso “Pergunta” .....	29
Tabela 4 – Caso de uso “Responde” .....	29

## LISTA DE SIGLAS E ABREVIATURAS

<i>.Net</i>	<i>Dot Net</i> – Ponto Net
ANPD	Autoridade Nacional de Proteção de Dados Pessoais
API	<i>Application Program Interface</i> – Interface de programação de aplicação
ASP.NET	<i>Active Server Pages dot Net</i> – Páginas de Servidor Ativas
DBMS	<i>Data Base Management System</i> – Sistema de Gerenciamento de Banco de Dados
HTML	<i>Hypertext Markup Language</i> – Linguagem de Marcação de Hipertexto
LGPD	Lei Geral de Proteção de Dados
Npgsql	<i>.Net Postgresql (provider)</i> – Provedor de Banco de dados .Net
SSL	<i>Secure Sockets Layer</i> – Camada de conexão segura
UML	<i>Unified Modeling Language</i> – Linguagem de Modelagem Unificada

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>16</b>
<b>2 OBJETIVOS</b> .....	<b>18</b>
2.1 OBJETIVOS ESPECÍFICOS.....	18
<b>3 REVISÃO DA LITERATURA</b> .....	<b>19</b>
3.1 LAUDO.....	19
3.2 ABSTRAÇÃO E ANÁLISE DE SISTEMAS .....	19
3.3 LINGUAGEM DE PROGRAMAÇÃO.....	20
3.4 BANCO DE DADOS .....	20
3.5 HOSPEDAGEM DA APLICAÇÃO.....	21
3.6 TÉCNICAS DE DESENVOLVIMENTO DO SOFTWARE .....	22
3.7 LEI GERAL DE PROTEÇÃO DE DADOS.....	24
<b>4 METODOLOGIA</b> .....	<b>25</b>
4.1 VISÃO GERAL.....	25
4.2 DEFINIÇÃO DO FLUXO .....	26
4.3 SELEÇÃO DE LINGUAGEM E BANCO DE DADOS .....	27
4.4 MODELAGEM DO SISTEMA .....	27
4.5 CASO DE USO .....	28
4.6 ESPECIFICAÇÃO DE CASO DE USO.....	28
4.7 DEFINIÇÃO DE DADOS.....	30
4.8 DIAGRAMA DE CLASSE.....	30
4.9 VISUAL STUDIO – AMBIENTE DE TRABALHO.....	31
4.10 FERRAMENTAS EXTERNAS .....	32
4.11 CLASSES DE ACESSO A DADOS .....	33
4.12 INTERFACE GRÁFICA PARA O USUÁRIO.....	35
<b>5 RESULTADOS E DISCUSSÃO</b> .....	<b>40</b>
5.1 RECURSOS DISPONÍVEIS.....	40
5.2 FUTURO DO PROJETO.....	43
<b>6 CONCLUSÃO</b> .....	<b>45</b>
<b>8 REFERÊNCIAS</b> .....	<b>46</b>

## 1 INTRODUÇÃO

Os softwares de compartilhamento de diagnóstico na radiologia odontológica brasileira se resumem às redes sociais e alguns fóruns de discussão. Não existe, ao melhor de nosso conhecimento, nenhum sistema oficial e específico para o fim de troca de opiniões sobre o diagnóstico por imagem entre os profissionais da área com profissionais de outras categorias da odontologia, como por exemplo os implantodontistas e ortodontistas.

Segundo Lee JS, et. al. (2003):

Os sistemas de colaboração em tempo real, nos quais os participantes compartilham dados e aplicativos multimídia em tempo real, fazem com que um radiologista de plantão possa fazer diagnósticos e relatar facilmente, visualizando as imagens transferidas em casa. Devido à acessibilidade da imagem, todos os usuários podem examinar e manipular as imagens de forma consistente, de modo que um hospital isolado possa receber assistência para realizar uma consulta remota.

A principal abordagem do projeto é desenvolver um sistema para troca de hipóteses de diagnóstico entre radiologistas e outras especialidades com a possibilidade de perguntas anônimas ou com sigilo de autor.

As discussões se estendem e conduzem a uma segunda opinião médica quando se tratar de profissionais da mesma área ou consulta avançada quando temos dúvidas a respeito de uma área complementar, um sistema web para múltiplas especialidades, ajustado para a radiologia odontológica, se torna uma ferramenta perfeita para a composição de um diagnóstico (KAMMERER et al, 2015).

Um software com ferramentas para compartilhar e discutir hipóteses de diagnóstico permite a integração multidisciplinar entre os profissionais da odontologia, servindo como um elo que liga o radiologista as demais especialidades. Neste caso, um paciente pode passar por estes profissionais em sequencias diferentes e com a possibilidade de reiteração. Os casos comuns consistem em um indivíduo que busca atendimento odontológico clínico e é enviado a um ortodontista para correção da mordida. Havendo necessidade, é encaminhado novamente a um implantodontista, fazendo um ciclo de tratamento em várias especialidades que dependem uma das outras. Qualquer especialista neste ciclo pode solicitar as radiografias ou tomografias (MONTEIRO E, OLIVEIRA C, 2016; MONIER et al, 2018; NEIRA et al, 2010).

Um software de compartilhamento de hipóteses de diagnóstico permite que seus usuários tenham acesso às imagens ao mesmo tempo e interajam entre si para conseguir o melhor para seus pacientes, também pode fazer com que a imagem disponibilizada possa ser analisada por outros profissionais que não fazem parte do tratamento, inclusive de forma anônima, caso necessário, e assim não deixa quem perguntou em situação embaraçosa quanto ao seu conhecimento.

O sistema foi desenvolvido de acordo com estas necessidades destes profissionais, e para suprir as necessidades de mercado, ao longo de sua leitura, apresentaremos o que é necessário para desenvolver uma aplicação deste tipo e quais ferramentas precisam ser utilizadas para realizar o trabalho, tendo como objetivo demonstrar a possibilidade de desenvolvimento sem a intenção de ser um guia de programação, mas um atalho que mostra a viabilidade da parceria entre a Tecnologia da Informação e as Ciências Biológicas e Médicas.

## 2 OBJETIVOS

Desenvolver um sistema para inserção de imagens, e, com elas, formular hipóteses de diagnóstico em laudos e compartilhar casos clínicos entre especialidades.

### 2.1 OBJETIVOS ESPECÍFICOS

- Desenvolver uma aplicação web utilizando a linguagem de programação C# na estrutura web disponível no ASP.NET
- Hospedar a aplicação em um servidor na World Wide Web
- Possibilitar a interação multidisciplinar e a colaboração dos colegas de profissão e demais especialistas da área.
- Possibilitar a inserção de perguntas com sigilo de autor a fim de incentivar qualquer tipo de pergunta, sem a preocupação com a reputação.

### 3 REVISÃO DA LITERATURA

#### 3.1 LAUDO

Os resultados de um estudo de caso radiológico são apresentados através de um laudo descritivo que pode vir acompanhado das imagens utilizadas para a sua confecção. Este laudo consiste em interpretar as imagens médicas, como as radiografias e tomografias, e transcrevê-lo levando em conta a anatomia e as anomalias encontradas nos exames, a normalidade também pode ser descrita neste laudo caso tenha sido solicitada pelo profissional a fim de tirar dúvidas sobre a saúde do indivíduo.

O laudo radiológico é um documento assinado por um dentista especialista em radiologia onde o profissional descreve os componentes encontrados em um raio x ou tomografia, observando o estado de normalidade ou alterações existentes de acordo com a anatomia. A confecção destes documentos é uma das principais atividades de clínicas de radiologia que realizam exames diariamente. E, melhorar este processo, tendo um modelo de laudo e recursos de autotexto, garante um bom fluxo e velocidade no detalhamento dos achados radiográficos, deixando o profissional livre para cuidar apenas da parte clínica e a parte computacional deve ficar transparente (TRONCOSO, 2023).

#### 3.2 ABSTRAÇÃO E ANÁLISE DE SISTEMAS

A abstração em computação é o ato de transformar uma demanda do mundo real em uma solução computacional para resolver um problema ou simplesmente realizar um cálculo complexo. Pode ser definido como o momento de concepção do software. Para projetar o software, em conjunto com a abstração, utilizamos técnicas de análise de sistema para confeccionar os principais diagramas da UML e descrever as necessidades desta demanda para um programador que deve ler o diagrama e codificar em uma linguagem de programação (MOZILLA, 2023A).

UML é a sigla em inglês de “*Unified Modeling Language*” ou, traduzinho, “Linguagem de Modelagem Unificada”, o UML é uma linguagem de notação e diagramas para uso em projetos de sistemas computacionais (ANDRADE, 2023).

### 3.3 LINGUAGEM DE PROGRAMAÇÃO

As linguagens de programação foram criadas para simplificar a forma de como damos instruções para um computador, elas transformam códigos de uma linguagem natural para um formato binário que as máquinas entendem e, com eles, realizam tarefas ou cálculos de forma sequencial e organizada.

As primeiras linguagens surgiram para construir as calculadoras programáveis, e traduzir fórmulas complexas. ADA e FORTRAN, entre outras menos famosas, formam a vanguarda da programação. Estas linguagens evoluíram com o avanço da tecnologia chegando até as linguagens contemporâneas como Java, Visual Basic, C++, Python e C#, amplamente utilizadas em tudo que temos contato neste mundo guiado pela tecnologia (PACIEVITCH, 2023).

Para conseguir migrar programadores de linguagens populares e muito bem difundidas e para parar a grande escalada do Java, a Microsoft criou o C#, uma nova linguagem de programação baseada em C++ e com a aparência do Java, totalmente orientada a objetos, e foi muito bem recebida pelos órfãos do Delphi e outros adeptos de outras plataformas, trazendo-os para o Visual Studio (MICROSOFT, 2021).

Para interagir com o usuário e criar aplicações dinâmicas e com recursos similares a de um programa de computador, grande parte dos softwares que roda em um navegador usa a linguagem Javascript, em especial o visualizador de imagens que precisa tratar os eventos do mouse e traduzi-los em ações como o zoom na imagem ou para arrastar uma imagem grande e assim mostrar uma estrutura anatômica ou região.

JavaScript é uma linguagem interpretada, orientada a objetos e rica em funções que interagem com o usuário e os outros elementos da computação na internet. Ela é mais conhecida como a linguagem de script para páginas Web, mas também é utilizada em vários outros ambientes sem navegadores (MOZILLA, 2023B).

### 3.4 BANCO DE DADOS

Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de

gerenciamento de banco de dados (DBMS, do inglês *Data Base Management system*). Juntos, os dados e o DBMS, e com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados (ORACLE, 2023).

Tudo que utilizamos em nosso cotidiano, desde uma lista de clientes aos lançamentos em nossa conta corrente são armazenados em banco de dados, um mecanismo baseado em linhas e colunas que de forma relacional, com uma combinação de tabelas torna o processamento e a consulta de dados transparente para os diversos aplicativos exibirem em suas telas. Assim, os dados podem ser facilmente acessados, manuseados, alterados e organizados. Quase todos os bancos de dados usam a linguagem SQL, linguagem consulta estruturada para escrever e consultar dados, algum banco antigo usa a sua própria notação.

O banco de dados PostgreSQL por ser um software gratuito e bem avaliado costuma ser uma das primeiras opções na escolha dos desenvolvedores, assim como SQL Server, MySQL, Oracle e outros. A escolha depende também do tamanho do projeto e dos recursos necessários pela aplicação (POSTGRESQL, 2023).

### 3.5 HOSPEDAGEM DA APLICAÇÃO

Uma vez desenvolvida e testada, uma aplicação web precisa ser disponibilizada para seu público, é necessário criar um domínio, endereço de internet utilizado para acessar um site, e apontar este domínio para o servidor onde o software ficará hospedado e disponível na *world wide web*. No Brasil, a responsabilidade de gerenciar domínios está atribuída ao registro.br. O Registro.br é o departamento do NIC.br responsável pelas atividades de registro e manutenção dos nomes de domínios que usam o .br. Para manter o serviço, uma taxa anual é cobrada pelo serviço de registro de domínio (REGISTRO.BR, 2023).

A hospedagem fica a cargo de empresas especializadas, elas são responsáveis por armazenar a sua aplicação e seus dados em um servidor compartilhado, popularmente chamado de “nuvem” na atualidade, e funciona em forma de serviço, devendo ser pago mensalmente. Como principais empresas do setor, podemos citar a GoDaddy, HostGator, Locaweb, e a própria Microsoft através do Windows Azure, elas estão entre as mais populares no nosso país.

Um servidor compartilhado nos termos gerais da internet é um computador com alto poder de processamento e armazenamento de dados que hospeda vários

sites, inclusive pode hospedar páginas de empresas distintas, separados apenas pelos seus endereços virtuais. O serviço pode ser pago e algumas vezes até gratuitos, e o valor varia de acordo com a tecnologia escolhida, os dados, a quantidade de acesso e o espaço em disco utilizado (LOCAWEB, 2023).

### 3.6 TÉCNICAS DE DESENVOLVIMENTO DO SOFTWARE

O desenvolvimento do sistema é a técnica de abstrair uma tarefa cotidiana em uma tarefa executada por um computador fazendo com que passos repetitivos sejam realizados pela máquina de forma organizada e sequencial e produzindo algum tipo de saída, como por exemplo um relatório contábil ou um gráfico de resultados.

Um projeto de sistema, visto do ponto de vista da análise de sistemas, é a descrição detalhada da estrutura de como software será implementado, e é composto pelos dados, as interfaces de interação com o usuário e os diagramas da UML e, às vezes, uma pequena parte dos algoritmos usados.

A partir da documentação, o desenvolver um software começa a escrever instruções em uma linguagem de alto nível, as linguagens de programação, essas instruções são convertidas em um código que poderá ser interpretado pela máquina através da compilação. Em software modernos, a programação é apenas parte do processo, é necessário envolver vários profissionais de diferentes ramos da computação e criar uma ligação entre as linhas de código e o ambiente que o usuário interage com o sistema, isso é feito por um designer de formulários ou um designer web, são eles os responsáveis por fazer a parte visual do software e tornar a nossa experiência com ele bem mais agradável.

As principais características de um software e o que o mercado espera encontrar depende do tipo do sistema, um sistema desktop, onde o software roda na máquina local é diferente de um sistema web que depende de um servidor para ser hospedado e roda de forma distribuída processando informações remotamente e na máquina local. Dentre outras, podemos citar:

**Robustez** - o sistema deve processar rapidamente as informações e devolver o resultado de forma precisa nas mais diversas situações.

**Amigabilidade** - ele deve ser de fácil entendimento e permitir que o usuário interaja de forma confortável com suas funções.

**Responsividade** - um software web precisa roda e se apresentar bem em um computador e também em celulares sem perder sua essência.

**Alta disponibilidade** - é desejável que o sistema seja operado tanto de forma online quanto off line, devendo ele enviar os dados ao servidor assim que houver conexão.

**Segurança** - em virtude de um software de internet estar disponível para todo o globo, é imprescindível que ele tenha seus recursos para impedir invasões e exposição de seus dados a quem não possui o direito.

Ainda sobre o desenvolvimento de sistemas, existem algumas fases no desenvolvimento de software, isso varia de acordo com o tipo do projeto e da equipe que irá trabalhar no projeto, as mais comuns são: análise de requisitos, análise, design (projeto), programação e testes. Estas fases podem exigir uma sequência de execução, mas algumas delas podem ser sobrepostas para acelerar a conclusão do trabalho e a elaboração do produto.

**Análise de requisitos** – Esta fase faz o levantamento de todas as necessidades do processo geral, ela pode ser dividida em tarefas ou ter um comportamento geral, isso é feito junto a um possível usuário e consiste em coletar documentos e entrevistas para compreender as necessidades e possíveis recursos de uso.

**Análise** – Esta fase projeta as primeiras abstrações, transforma objetos reais em classes computacionais, é a forma com que o analista de sistema deixa o projeto compreensível para quem vai executá-lo na fase da programação, essas classes são descritas em forma de diagramas, podemos citar o Diagrama de Classe, o Diagrama de sequência e o Diagrama de Casos de Uso para ilustrar a tarefa.

**Design** – Separadamente do projeto do sistema, podemos ter o design da apresentação, as telas e páginas do sistema precisam ser criadas por um profissional habilidoso com a comunicação com o usuário e muita vez deve escolher e optar por um ambiente bonito ou funcional, de acordo com as expectativas de uso do sistema.

**Programação** – Esta fase consiste em converter as classes e diagramas criadas nas duas fases anteriores em códigos em linguagem de programação orientada a objetos, C# por exemplo ou Java estão entre as mais utilizadas no momento.

**Testes** – Um sistema precisa passar por um rigoroso controle antes de ser distribuído, estes testes podem ser testes de unidade, integração e aceitação, e são feitos em etapas distintas e por profissionais distintos. O programador faz a primeira testagem e assim que conclui, um outro profissional testa o conjunto completo.

### 3.7 LEI GERAL DE PROTEÇÃO DE DADOS

Desde setembro de 2020 está em vigor a Lei Geral de Proteção de Dados Pessoais (LGPD), Lei n. 13.709, de 14 de agosto de 2018. Esta lei regula o tratamento de dados pessoais no Brasil, tanto em meios físicos, impressos quanto de modo digital, publicado na internet, isso se aplica para instituições públicas e privadas e pode ser utilizada em conjunto com outras leis internacionais para garantir a segurança dos dados. Os dados passam a ser controlados pelos proprietários e não por quem os armazena, cabendo assim, a necessidade de um termo de consentimento para uso de dados e mesmo as imagens de um indivíduo. O órgão responsável pela fiscalização e aplicação de penalidades é a ANPD, Autoridade Nacional de Proteção de Dados Pessoais (MINISTÉRIO PÚBLICO FEDERAL, 2023; PRESIDÊNCIA DA REPÚBLICA, 2023).

## 4 METODOLOGIA

### 4.1 VISÃO GERAL

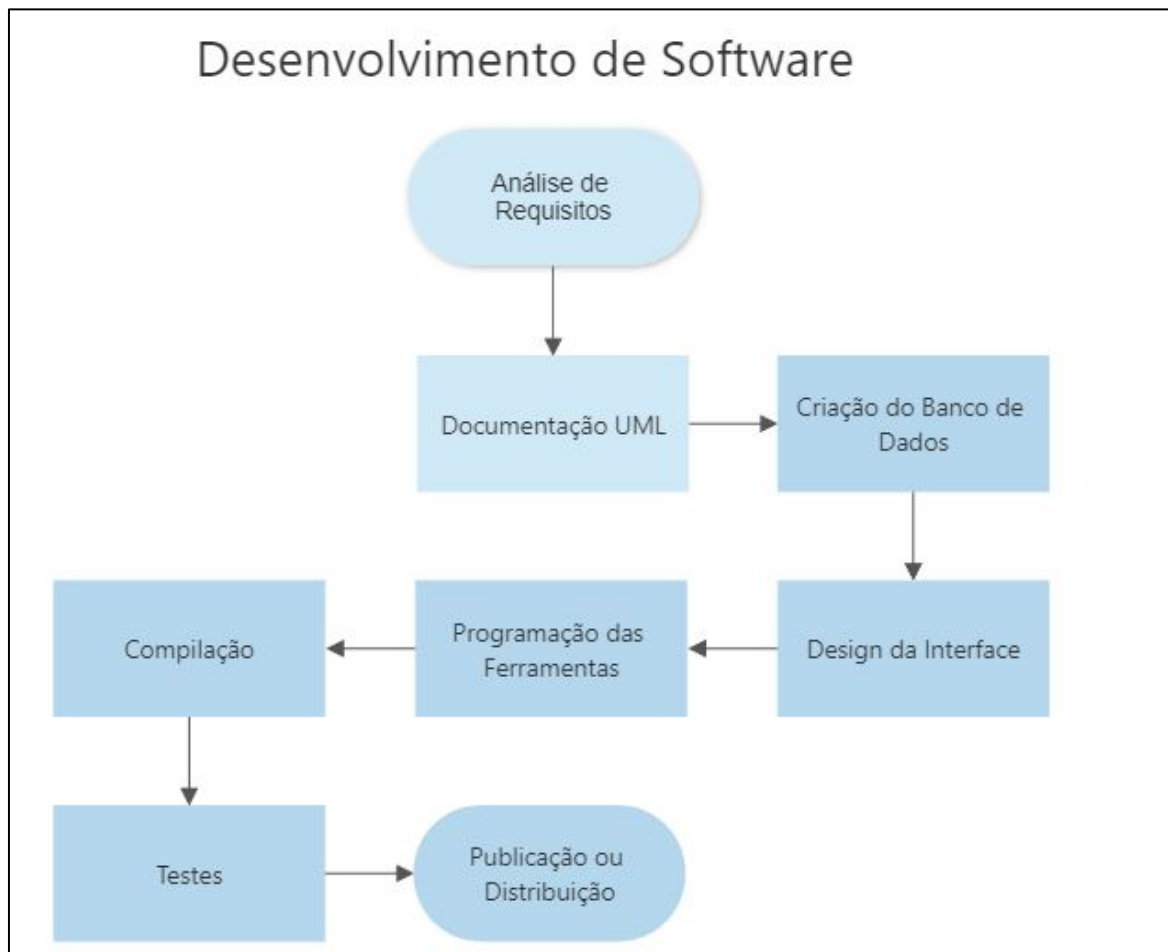
O desenvolvimento do referido software seguiu as práticas de análise de sistemas e de desenvolvimento de forma simplificada e direta, separado em etapas distintas, por se tratar de um software de porte pequeno, não precisou dispor de vários profissionais, o mesmo profissional desenvolveu o papel de analista e programador e fez uso de web design padrão do ambiente de desenvolvimento.

Para iniciar o processo, o analista de sistemas iniciou a sua tarefa pelo levantamento de requisitos junto a pessoa retentora do conhecimento e dos processos, e coletou documentos para servir como base. No Sistema de Compartilhamento de Hipótese de Diagnóstico, a fonte de informação foi um radiologista e os documentos coletados foram laudos e imagens radiográficas. Ainda faz parte do trabalho do analista de sistemas, o desenho dos diagramas UML e a escrita de parte da documentação do software.

Utilizando os diagramas e a documentação, os processos de criação de banco de dados e a programação foram desempenhadas por um administrador de banco de dados e um programador em conjunto com um *web designer* produz as telas que interagem com o usuário. Por fim, um testador deve testar o funcionamento geral do sistema e relatar na documentação do sistema os possíveis problemas.

A Figura 1 mostra um fluxograma com as fases do desenvolvimento de software.

Figura 1- Fluxo de metodologia de desenvolvimento do software



Fonte: criação própria

#### 4.2 DEFINIÇÃO DO FLUXO

O fluxo de uso do software começa pela tela de login, onde o usuário digita seu usuário e senha antes de interagir com as demais funções do sistema. Caso não tenha um login e senha, é possível criar um novo usuário e assim usar a aplicação.

As tarefas principais disponíveis após a entrada são divididas em fazer uma pergunta ou responder a elas, clicando no botão “fazer pergunta” ou utilizar uma das caixas de texto abaixo de cada pergunta para respondê-la. Durante a navegação entre as perguntas, é possível clicar em uma imagem para abrir o visualizador de imagens e assim ampliar a imagem e dar zoom e arrastá-la para uma região de interesse.

A tarefa de fazer uma pergunta é subdividida em escolher o tipo da pergunta, selecionar as imagens para enviar para o servidor e escolher a opção de anonimato,

antes de salvar é preciso concordar com os termos da Lei Geral de Proteção de dados, e inserir o documento de aceite de forma digital.

#### 4.3 SELEÇÃO DE LINGUAGEM E BANCO DE DADOS

O software terá como base de desenvolvimento a linguagem C#, desenvolvida pela Microsoft, e o conjunto de ferramentas para criação de páginas contidas no .Net Framework e ASP.NET, a empresa disponibiliza uma versão gratuita do Visual Studio para desenvolvedores que precisam dessa modalidade (MICROSOFT, 2023).

Para interagir com o usuário foi usada a linguagem JavaScript, com ela é possível criar os eventos para controlar o mouse para dar zoom e arrastar a imagem assim como dar brilho e contraste na imagem.

O banco de dados para armazenamento foi o *Postgres SQL* e os arquivos seguiram a técnica de repositório com base nas pastas do sistema operacional utilizando a chave primária do paciente, um número sequencial que individualiza o registro, e um sufixo com o nome da imagem ou do exame.

#### 4.4 MODELAGEM DO SISTEMA

A modelagem de sistemas é o processo de desenvolvimento de modelos abstratos de um sistema, cada modelo apresenta uma maneira diferente de visualização do sistema, ela ajuda o analista a entender a funcionalidade do projeto e apresentar à equipe de desenvolvimento e ao cliente de forma que ambos a entendam.

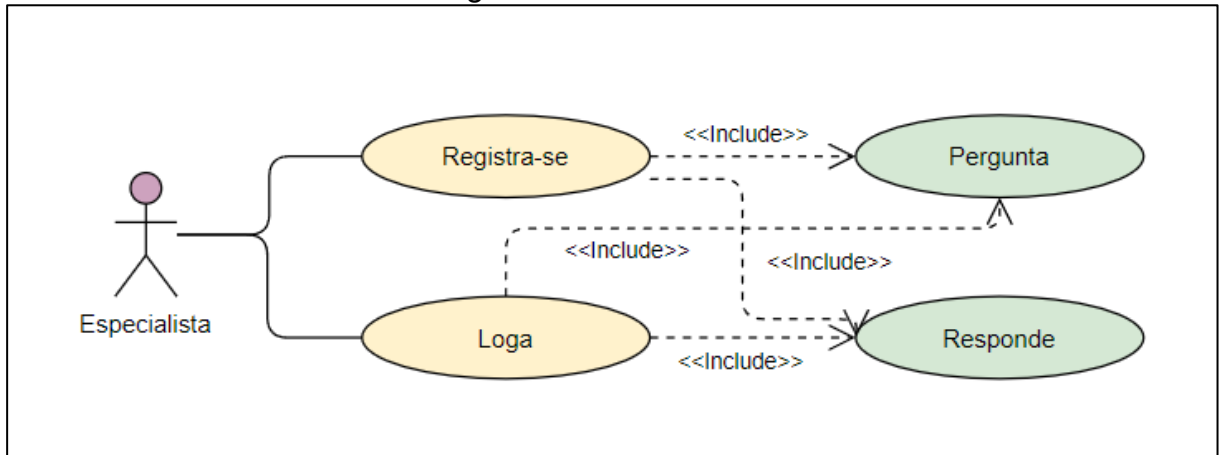
A linguagem utilizada para elaborar a modelagem é a UML, *Unified Modeling Language*, termo em inglês para Linguagem de modelagem unificada. Ela é composta por diagramas, documentos e planilhas relativos ao negócio do cliente.

Os principais diagramas são: diagrama de casos de uso, diagrama de classes, diagrama de objetos, diagrama de colaboração, diagrama de sequência, diagrama de atividades e ainda outros, o uso dos diagramas depende do tipo de desenvolvimento e nem todos são obrigatórios. Na metodologia de desenvolvimento ágil, utilizado no desenvolvimento deste projeto, o mínimo de diagramas é requerido.

## 4.5 CASO DE USO

A Figura 2 ilustra o diagrama de caso de uso principal, onde um especialista se cadastra no sistema e insere a sua primeira pergunta ou responde uma pergunta já existente na base de dados, os demais serão apresentados em tabelas.

Figura 2 – Caso de Uso



Fonte: criação própria

## 4.6 ESPECIFICAÇÃO DE CASO DE USO

### 4.6.1 Registra-se

Tabela 1 – Caso de uso “Registra-se”

Nome do caso de uso	Registra-se
Atores:	Especialista
Pré condições:	Não haver um cadastro
Fluxo principal:	1 – O usuário digita o endereço da aplicação web no navegador 2 – O usuário seleciona a opção registrar-se 3 – O usuário preenche os dados e salva

Fonte: Autoria própria

## 4.6.2 Loga

Tabela 2 – Caso de uso “Loga”

Nome do caso de uso	Loga
Atores:	Especialista
Pré condições:	Haver um cadastro
Fluxo principal:	1 – O usuário digita o endereço da aplicação web no navegador 2 – O usuário entra com seu nome de usuário e senha 3 – O sistema o encaminha para a lista de perguntas

Fonte: Autoria própria

## 4.6.3 Pergunta

Tabela 3 – Caso de uso “Pergunta”

Nome do caso de uso	Pergunta
Atores:	Especialista
Pré condições:	Estar logado
Fluxo principal:	1 – O usuário clica no botão fazer pergunta 2 – Preenche o formulário de dados 3 – O sistema disponibiliza a pergunta para outra pessoa responder

Fonte: Autoria própria

## 4.6.4 Responde

Tabela 4 – Caso de uso “Responde”

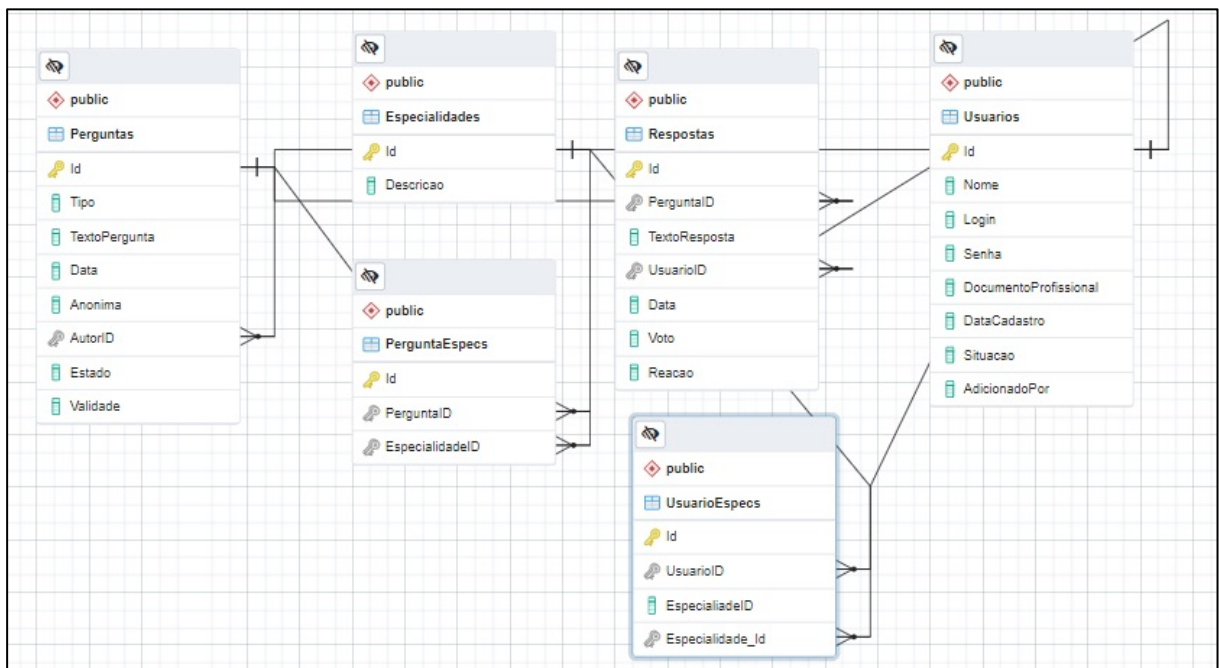
Nome do caso de uso	Responde
Atores:	Especialista
Pré condições:	Haver uma pergunta
Fluxo principal:	1 – O usuário seleciona uma pergunta 2 – O usuário analisa as imagens disponíveis e descreve na caixa de texto 3 – O sistema o encaminha para a lista de perguntas e repostas

Fonte: Autoria própria

#### 4.7 DEFINIÇÃO DE DADOS

O banco de dados conta com uma tabela de usuários, a tabela de perguntas, a tabela de respostas, a tabela de Especialidades, e seus relacionamentos e campos são mostrados em detalhes no diagrama de entidade relacionamento na Figura 3.

Figura 3 - Diagrama de Entidade Relacionamento

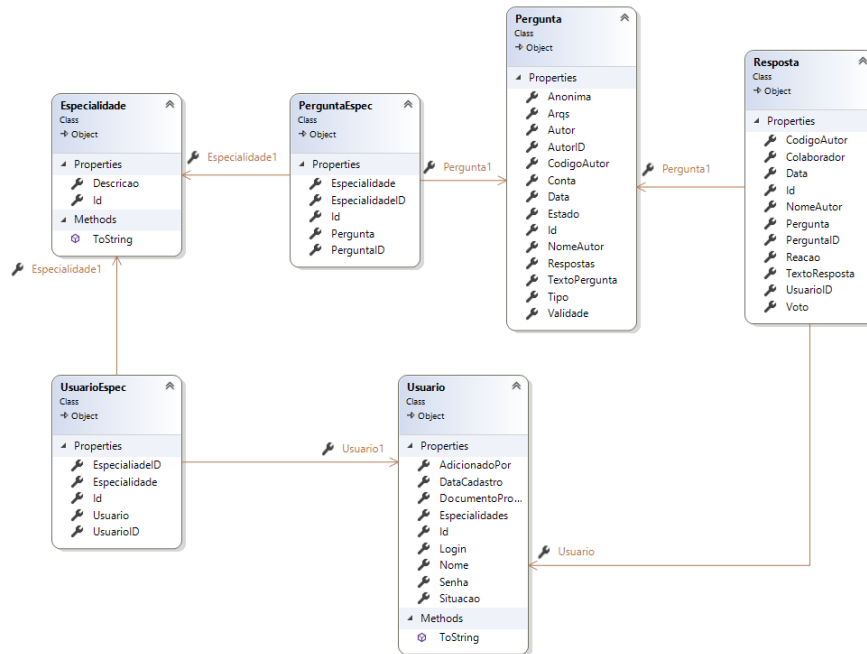


Fonte: Criação própria

#### 4.8 DIAGRAMA DE CLASSE

O diagrama de classe, apresentado na Figura 4, mostra ao desenvolvedor a composição e a relação das classes de um projeto e descrever todos os seus atributos e métodos, ele também serve para a construção de outros diagramas, como o de comunicação, sequência e o diagrama de estado. Como os outros recursos da UML, ele é construído independente da linguagem alvo e pode ser implementado na maioria das linguagens atuais direcionadas a softwares comerciais.

Figura 4 – Diagrama de Classe



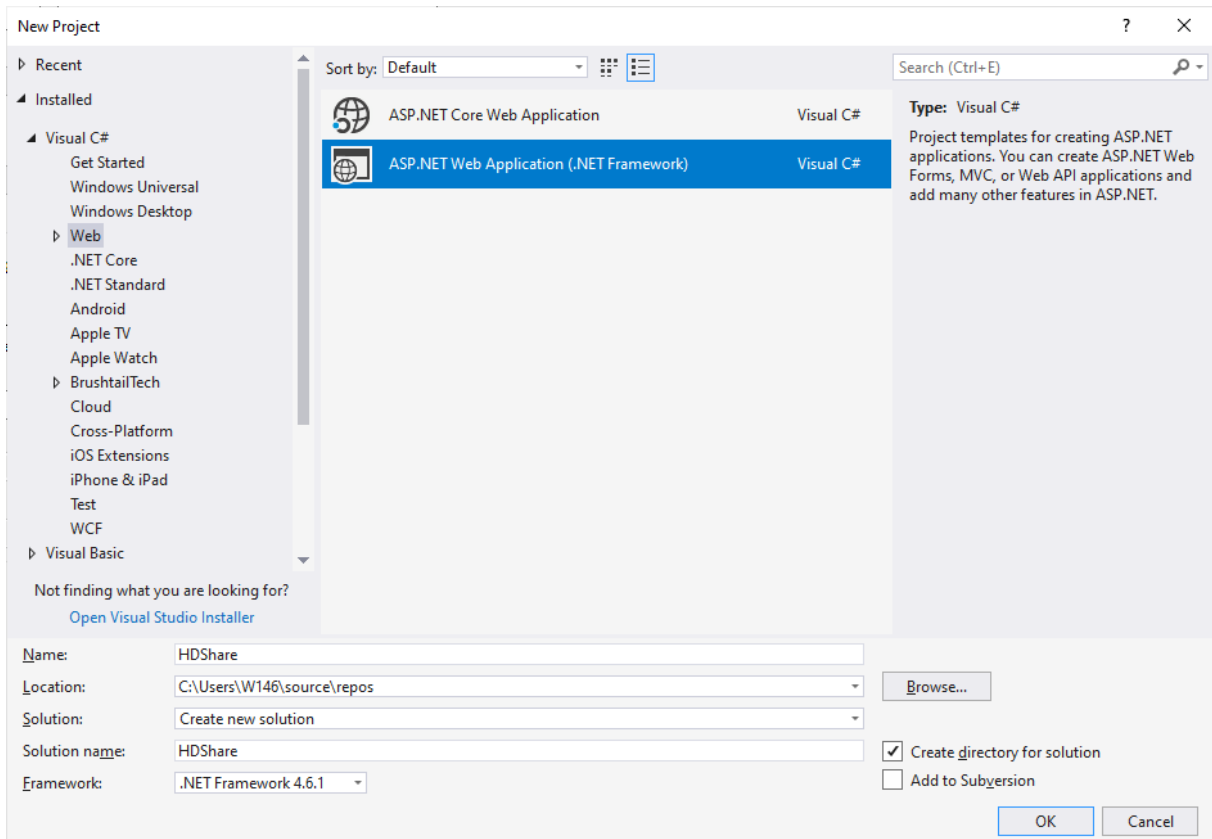
Fonte: criação própria

#### 4.9 VISUAL STUDIO – AMBIENTE DE TRABALHO

Para construir o sistema, foi utilizado o Visual Studio Community 2017, que é uma ferramenta de desenvolvimento onde é possível além de editar o código fonte, criar a interface com o usuário e depurar possíveis erros de programação. O tipo de projeto, ou, *project type*, utilizado no desenvolvimento de software foi *ASP.NET Web Application* com C# e o .Net Framework 4.6.1, conforme mostra a Figura 5.

Este tipo de projeto insere automaticamente vários arquivos à aplicação e configura o ambiente para o programador se preocupar apenas em desenhar suas telas e escrever o código que faz com que tudo funcione.

Figura 5 - Projeto no Visual Studio



Fonte: criação própria

#### 4.10 FERRAMENTAS EXTERNAS

O projeto padrão do Visual Studio instala os pacotes de componentes básicos para desenvolver uma aplicação web, devendo ao desenvolvedor instalar os pacotes opcionais. Uma das coisas criadas para o Visual Studio que facilita muito a vida é o gerenciador de pacotes *Nuget*, com ele podemos pesquisar entre inúmeras bibliotecas e inserir ferramentas úteis ao nosso desenvolvimento, o *Nuget* também nos ajuda a escolher a melhor versão de um determinado produto. Vejamos um exemplo de uso: como o nosso projeto usa o banco de dados PostgreSQL, foi necessário adicionar os pacotes *Npgsql* e *Npgsql.EntityFrameworkCore*, eles são usados para acessar o banco de dados e não são instalados como padrão no Visual Studio, pois usa como padrão o Microsoft SQL Server.

#### 4.11 CLASSES DE ACESSO A DADOS

A forma como o sistema acessa o banco de dados varia de acordo com o as necessidades de programação ou da análise de sistema software, o mais comum para desenvolvedores Microsoft é utilizar o *Entity Framework*, ferramenta visual para referenciar as tabelas no banco, ele liga as tabelas do banco as aplicações diretamente de forma visual, utilizando o modo interativo do Entity Framework, ou via código, chamado de *code first*, este último é o modo utilizado para desenvolver o sistema, e consiste em escrever primeiro as classes em C# e ao executar o sistema pela primeira vez, o mecanismo cria o banco de dados, as tabelas e suas dependências. Em ambos os casos é necessário ter uma *string* de conexão previamente configurada no arquivo web.config e mostrada na Figura 6.

Figura 6 – String de conexão

```
<connectionStrings>
    <add name="Default"
        connectionString="server=localhost;database=ubrquest;userid=postgres;password=
        xxx;port=5432" providerName="Npgsql" />
</connectionStrings>
```

Fonte: Código fonte do sistema

O trecho de código na Figura 7 mostra a classe Usuario, escrita em C#, onde o ASP.NET e o .Net framework fazem o papel de transformar esse código de programação em um mecanismo que cria e mantém um acesso ao banco de dados de forma transparente e automática.

Figura 7 – Exemplo de classe de dados

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations;

namespace Projeto_Mestrado.Data
{
    public class Usuario
    {
        [Key]
        public int Id { get; set; }
        public string Nome { get; set; }
        public string Login { get; set; }
        public string Senha { get; set; }
        public string DocumentoProfissional { get; set; }
        public DateTime DataCadastro { get; set; }
        public int Situacao { get; set; }
        public int AdicionadoPor { get; set; }

        public virtual List<UsuarioEspec> Especialidades { get; set; }

        public override string ToString()
        {
            return Nome;
        }
    }
}
```

Fonte: Código fonte do sistema

Durante a codificação, a estrutura de programação em camadas precisa ser dividida em várias outras classes para acessar o banco, assim como a técnica de ligação da aplicação ao banco, ela também pode ser feita com o uso de abordagens diversas, a mais simples, composta de três camadas, consiste em criar uma camada de ligação entre a interface e o acesso ao banco, é camada de negócio, ela é a responsável por validar os dados antes que eles sejam inseridos ou alterados no banco e possui métodos para listar, inserir, alterar e excluir registros das tabelas. O trecho de código na Figura 8 ilustra uma dessas classes formatada no editor de código do Visual Studio.

Figura 8 – Exemplo de classe de lógica de negócio

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Projeto_Mestrado.Data.BL
{
    public class UsuarioEspecialidadesBLL
    {
        QuestContext db = new QuestContext();

        public IQueryable<Especialidade> listarPorUsuario(int usr)
        {
            var lista= from u in db.UsuarioEspec
                join q in db.Especialidades on u.EspecialiadeID equals
q.Id
                where u.UsuarioID == usr
                orderby u.Especialidade.Descricao
                select q;

            return lista;
        }

        public IQueryable<Especialidade> listarDisponiveis(int usr)
        {
            var lista = listarPorUsuario(usr);
            var result = db.Especialidades.Where(p => !lista.Any(p2 => p2.Id ==
p.Id));

            return result.OrderBy(p => p.Descricao);
        }
    }
}

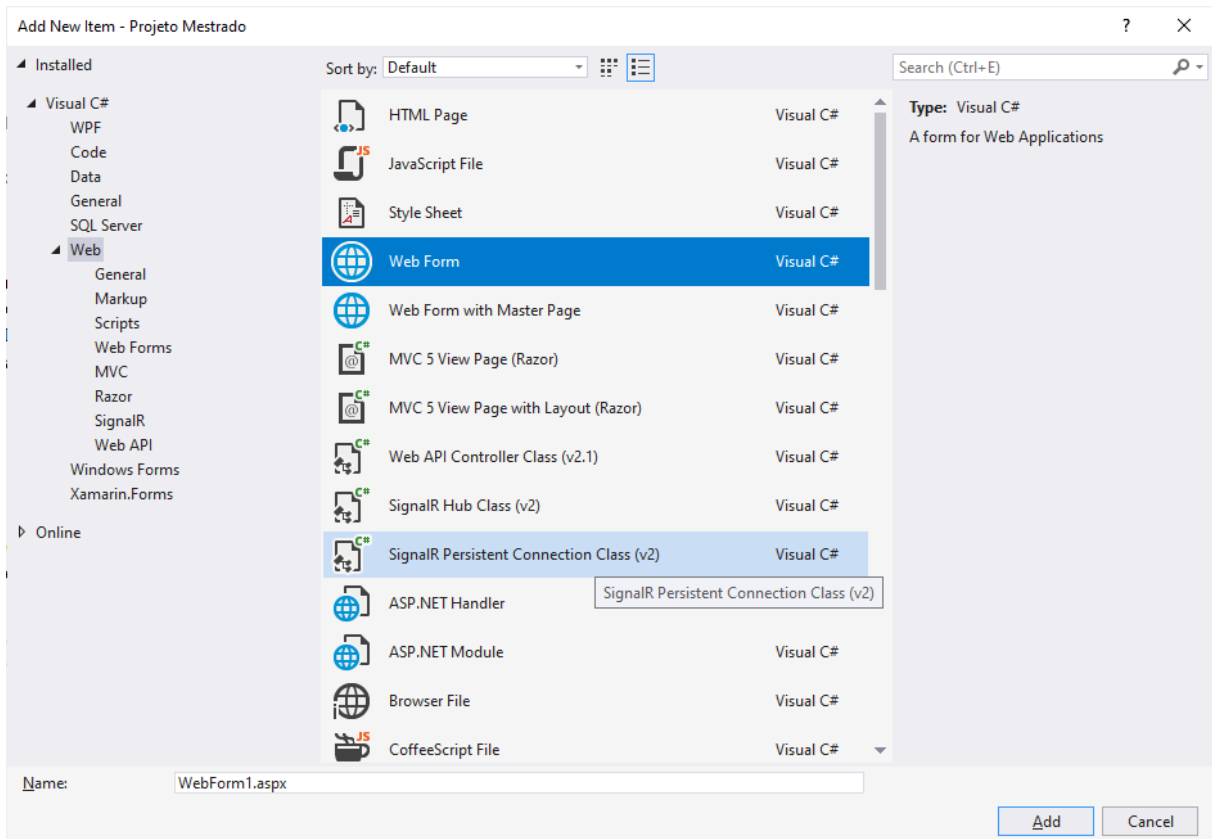
```

Fonte: código fonte do sistema

#### 4.12 INTERFACE GRÁFICA PARA O USUÁRIO

A interface com o usuário corresponde a maior parte da programação de um sistema como um todo, tratando-se de um sistema web criado pelo .Net Framework, a estrutura de uma única página é dividida em duas partes, uma contendo tags XML específicas do ASP.NET e outra contendo códigos em C# para manipular os eventos e criar métodos para tratar as ações do usuário. Para tanto, o Visual Studio se encarrega por criar os arquivos com essa divisão e o programador apenas acrescenta os elementos visuais no arquivo chamado Design e os códigos no segundo arquivo, o que chamamos de *code behind*. A Figura 9 mostra no ambiente de desenvolvimento a criação de uma nova página ASP.NET utilizando o modelo Web Form. Esse processo é repetido para cada formulário do sistema, como o cadastro de usuários por exemplo ou o cadastro de especialidades.

Figura 9 – Adicionar Web Form



Fonte: Microsoft Visual Studio

Quando um formulário é inserido, ele não vem um arquivo em branco, o modelo oferecido pelo *ASP.NET* traz o arquivo de design com algumas *tags* HTML, conforme a Figura 10 e a partir dele é preciso acrescentar os componentes adicionais para dar vida à página.

Figura 10 – Exemplo de arquivo Aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Figuras.aspx.cs"
Inherits="Projeto_Mestrado.Figuras" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      </div>
  </form>
</body>
</html>

```

Fonte: código fonte do sistema

O mesmo acontece com o arquivo do *code behind*, ele já vem com a declaração da classe e com o método *page\_load* declarado, ilustrado na Figura 11. O próximo passo é colocar os eventos para tratar o comportamento do sistema. O código do *code behind* é quase sempre processado no servidor e devolvido ao usuário com as alterações no formato HTML para ser exibido no navegador.

Figura 11 – Exemplo de *code behind*

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Projeto_Mestrado
{
  public partial class Figuras : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {

    }
  }
}

```

Fonte: código fonte do sistema

Um exemplo prático é colocar um componente *Label*, um rótulo para inserir texto numa página ASP, e um componente *Button*, um botão, dentro da tag *div* no formulário principal do arquivo *design* e tratar o evento *click* deste botão no *code behind*. Veja o código alterado na Figura 12 e o respectivo evento na Figura 13.

Figura 12 – Inserção de componente do Aspx

```
<form id="form1" runat="server">
  <div>
    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
    <asp:Button ID="Button1" runat="server" Text="Button"
OnClick="Button1_Click"/>
  </div>
</form>
```

Fonte: código fonte do sistema

Figura 13 – Manipulação de eventos do componente Aspx

```
public partial class Figuras : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = "hora :" + DateTime.Now.ToString("HH:mm:ss");
    }
}
```

Fonte: código fonte do sistema

#### 4.13 O VISUALIZADOR DE IMAGENS

O coração do sistema e a parte mais complexa do desenvolvimento está na interface, o .Net framework, utilizado pelo ASP.NET faz a maior parte do trabalho para o desenvolvedor, é este framework que renderiza o HTML através do código escrito em *c#*, e das *tags* específicas contidas no design das páginas. Mas isso não é suficiente, uma aplicação rica em interações com o usuário requer algum esforço e precisa de recursos específicos que são encontrados com facilidade na linguagem Javascript. Este trabalho não tem como objetivo detalhar o desenvolvimento do sistema, mas apresentar o que é necessário para construir uma aplicação capaz de mostrar uma imagem na tela do computador e manipulá-la de forma que o radiologista possa encontrar estruturas anatômicas na radiografia ou alguma anormalidade nelas.

Na Figura 14 temos parte do código em Javascript responsável por manipular os eventos do mouse e produzir os movimentos nas imagens ou aplicando o zoom nas mesmas. O código completo estará disponível no anexo A.

Figura 14 – Exemplo de código javascript

```
function inicializaContexto() {  
  
    ctx = canvas.getContext('2d');  
    trackTransforms(ctx);  
  
    redraw();  
  
    var lastX = canvas.width / 2,  
        lastY = canvas.height / 2;  
  
    var dragStart, dragged;  
  
    canvas.addEventListener('mousedown', function (evt) {  
        document.body.style.mozUserSelect = document.body.style.webkitUserSelect =  
document.body.style.userSelect = 'none';  
        lastX = evt.offsetX || (evt.pageX - canvas.offsetLeft);  
        lastY = evt.offsetY || (evt.pageY - canvas.offsetTop);  
  
        if (lastX < 50) {
```

Fonte: código fonte do sistema

Todas as etapas descritas neste capítulo foram utilizadas para a elaboração do software, a análise de sistema e seus diagramas, os componentes escolhidos e os softwares adicionais são necessários para que o sistema seja construído e possa funcionar adequadamente. O código fonte completo está disponível no site da aplicação.

## 5 RESULTADOS E DISCUSSÃO

### 5.1 RECURSOS DISPONÍVEIS

Após desenvolvimento e publicação, o software pode ser acessado através do endereço [www.hdshare.com.br](http://www.hdshare.com.br), e conta com os recursos de controle de usuários, inserção e comentário das perguntas, visualização de imagens com zoom e possibilidade de arrastar para posicionar, modo de sigilo de autor para perguntas, enquete, separação por especialidade, compartilhamento por especialidade, armazenamento de casos para base de conhecimento, entre outros (BALTA, 2023).

O sistema de Compartilhamento de Hipótese de Diagnóstico foi desenvolvido para retirar a carência da ferramenta e criar recursos adicionais para o diagnóstico por imagem. O foco principal do software é a possibilidade de fazer perguntas de forma anônima, preservando a identificação do profissional com o objetivo de deixar o usuário livre para perguntar qualquer coisa, e sem se preocupar se está fazendo uma pergunta muito óbvia a ponto de manchar sua reputação. A Figura 15 ilustra a lista de perguntas do sistema.


Figura 15 – Postagem no Sistema de Hipótese de diagnóstico

↑  
0  
↓

**Anônimo**

19/02/2023 17:24:08

Boa tarde colegas! Gostaria da ajuda de vcs em relação a esse caso: paciente sexo feminino, 16 anos e 4 meses, sente dor na região. Qual o possível HD?



**Ramao**  
Acredito ser uma lesao benigna, tipo amelo ou querato. Pedir tomo para melhor avaliação

**Ramao**  
Descrevê-la detalhadamente, sugerir TC e histopatológico. Integridade radicular dos elementos envolvidos, expansão de tábua óssea, lesão radiolúcida multilocular.

**Resposta:**

Enviar

Fonte: [www.hdshare.com.br](http://www.hdshare.com.br)

Um recurso importante para o diagnóstico por imagem é o Visualizador de imagens, que é ativado clicando na miniatura da imagem na lista de perguntas. Esta opção dispõe de recursos de zoom, arrastar imagem, brilho e contraste utilizando os botões do mouse combinado com as teclas *shift* ou *control* conforme a Figura 16.

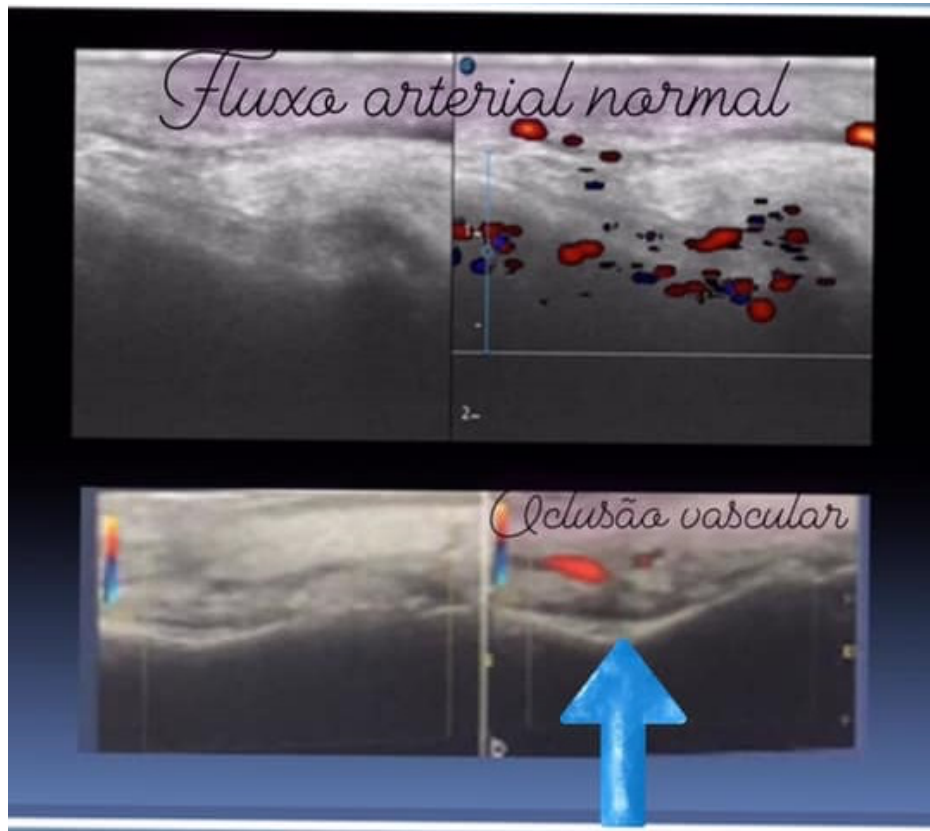
Figura 16 – Zoom no visualizador de Imagens



Fonte: [www.hdshare.com.br](http://www.hdshare.com.br)

O software não se limita a radiologia odontológica e suas especialidades correlacionadas, com o cadastro de especialidades, é possível direcionar o sistema para outras áreas que, utilizando imagens ou outro recurso multimídia, possam tirar dúvidas de uma população específica e compartilhá-las com outras pessoas. Cardiologia, como a da Figura 17, ultrassonografia e até mesmo a estética pode ganhar poder de discussão com a separação por grupos.

Figura 17 – Exemplo de uso em outras áreas



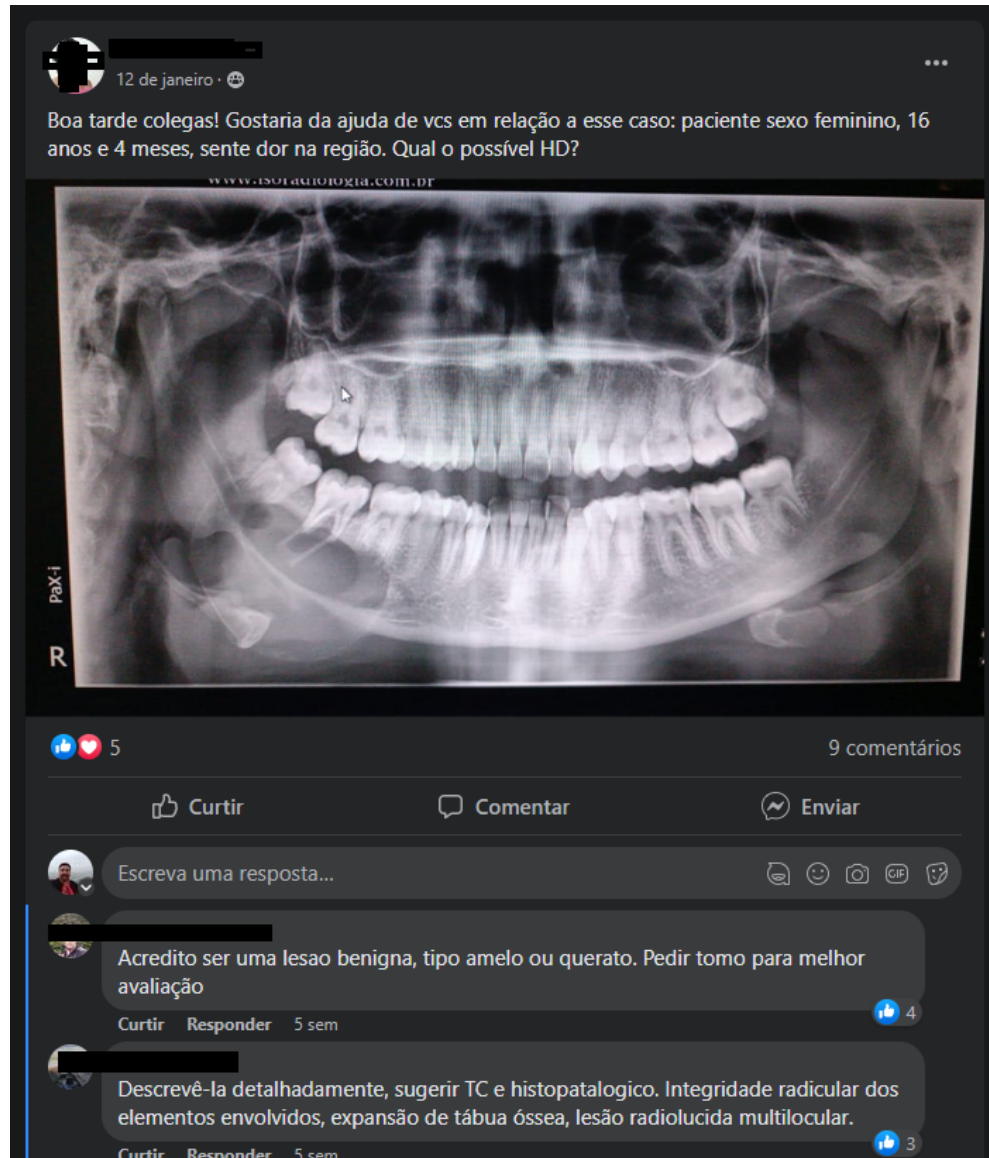
Fonte: <https://www.facebook.com/groups/radiologiamedicabrasileira>

O sistema conta com o compartilhamento das hipóteses de diagnóstico entre as especialidades, com isso, é esperado melhora na quantidade de informação encontrada nos exames dos pacientes e uma contribuição de outras especialidades para um diagnóstico mais preciso. Isso acontece quando um profissional de uma área relacionada, complementa o laudo com informações específicas com sua formação. Um ortodontista pode pedir ajuda a um implantodontista sobre quanto espaço seria necessário deixar para fazer um procedimento posterior, assim, fazem o planejamento em conjunto.

A principal ferramenta utilizada para compartilhamento de hipótese de diagnóstico atualmente é um grupo de radiologistas no Facebook, os membros postam as imagens, mas se deparam com as deficiências da plataforma, a versão desktop da rede social não dispõe de um zoom eficiente para analisar uma região específica da radiografia e os botões de reação são utilizados para mostrar quem concorda com a postagem de forma amadora. O recurso de comentários é um dos poucos recursos utilizados sem alguma limitação. E por se tratar de um grupo, é

limitado a uma única especialidade. A Figura 18 mostra uma foto da tela do grupo de discussão sobre radiologia.

Figura 18 – Postagem no grupo de discussões



Fonte: [www.facebook.com/groups/radiologiaodontologica](http://www.facebook.com/groups/radiologiaodontologica)

## 5.2 FUTURO DO PROJETO

A etapa de testes do software foi realizada apenas em ambiente de desenvolvimento e produção, um dos próximos passos seria a implantação do serviço junto ao grupo da radiologia odontológica para validar a proposta, isso demandaria esforço de convencimento de uso, o que esbarra em tirar o usuário de

sua zona de conforto com a troca de tecnologia e ainda negociar com os fundadores do grupo, ou seja, tirar o controle das mãos dos mesmos, sendo assim, a troca deveria ter benefícios também a essas pessoas.

Pensando na continuidade do projeto, o software tem potencial de monetização, a plataforma de *Facebook* é gratuita e ganha dinheiro com propagandas e com a venda de informações, o mesmo tipo de negócio pode ser aplicado. Baseando-se na radiologia odontológica como público alvo, temos clientes com poder de compra significativo e que investe em equipamentos caros e têm histórico de uso de softwares pagos para gerenciamento de suas clínicas e, entre outros, softwares científicos para cálculos de cefalometria e simuladores de implante. Isso pode atrair patrocinadores para o projeto ou empresas interessadas em expor seus produtos para este grupo privilegiado de clientes e tornar viável a manutenção do sistema de forma profissional e com evolução constante.

## 6 CONCLUSÃO

Utilizando as técnicas de análise e desenvolvimento de sistemas aplicadas à linguagem C# e o ASP.NET foi possível desenvolver um software para compartilhamento de hipótese de diagnóstico com a possibilidade de manter sigilo do autor das perguntas.

Os dados armazenados em um banco de dados possibilitam a sua exibição e alteração de forma dinâmica e uma vez publicado em um servidor, foi possível ter acesso a eles via internet.

O modo como o sistema foi projetado garante que várias especialidades possam compartilhar um mesmo problema, e em conjunto, encontrar a melhor solução para o paciente, como foi apresentado para a radiologia, ou outro tipo de indivíduo em outras áreas do conhecimento.

## 8 REFERÊNCIAS

- ANDRADE, A. TreinaWeb. **O que é UML?** Disponível em <<https://www.treinaweb.com.br/blog/o-que-e-uml>> Acessado em 27 de janeiro de 2023
- BALTA, R. **Hipótese Diagnóstica**. Disponível em <<http://hdshare.com.br>> acessado em 27 de janeiro de 2023
- KAMMERER, F.J.; HAMMON, M.; SCHLECHTWEQ, P.M., UDER, M, SCHWAB, S.A.. **A web based cross-platform application for teleconsultation in radiology**. J Telemed Telecare. 2015 Sep;21(6):355-63. doi: 10.1177/1357633X15575237. Epub 2015 May 10. PMID: 25962651.
- LEE JS, TSAI CT, PEN CH, LU HC. **A real time collaboration system for teleradiology consultation**. Int J Med Inform. 2003 Dec;72(1-3):73-9. doi: 10.1016/s1386-5056(03)00130-8. PMID: 14644308.
- LOCAWEB. **O que é e como funciona um Servidor de Hospedagem de Sites?** Disponível em <https://www.locaweb.com.br/conteudos/servidor-de-hospedagem/>. Acessado em 23 de fevereiro de 2023
- MELÍCIO MONTEIRO EJ, COSTA C, OLIVEIRA JL. **A Cloud Architecture for Teleradiology-as-a-Service**. Methods Inf Med. 2016 May 17;55(3):203-14. doi: 10.3414/ME14-01-0052. Epub 2016 Mar 4. PMID: 26940635.
- MICROSOFT. **Microsoft - Documentação do C#**. Disponível em <<https://docs.microsoft.com/pt-br/dotnet/csharp/>> Acessado em 11 de junho de 2021.
- MICROSOFT. **O que é ASP.NET?** Disponível em <https://dotnet.microsoft.com/pt-br/learn/aspnet/what-is-aspnet>. Acessado em 27 de janeiro de 2023
- MICROSOFT. **ASP.NET Free. Cross-platform. Open source**.<disponível em <https://dotnet.microsoft.com/apps/aspnet>> acessado em 11 de junho de 2021.
- MINISTÉRIO PÚBLICO FEDERAL. **Lei Geral de Proteção de Dados**,. Disponível em <<https://www.mpf.mp.br/servicos/lcpd/o-que-e-a-lcpd>>. Acessado em 3 de março de 2023
- MONIER, ELZA BERNARDES; ARAUJO, DENIZAR VIANNA; OLIVEIRA, ANA EMILIA FIGUEIREDO DE; FERNANDES, ANA CAROLINA URUÇU REGO; CANTANHEDE, LUANA MARTINS; FURTADO, MANUELA ALMEIDA MONTENEGRO. **O uso de recursos digitais no ensino de Radiologia Odontológica: uma revisão integrativa de literatura**. Rev. ABENO ; 18(3): 75-83, jul.-set. 2018. ilus, tab
- MOZILLA. **Abstração** <disponível em <https://developer.mozilla.org/pt-BR/docs/Glossary/Abstraction>> acessado em 27 de janeiro de 2023A
- MOZILLA. **Javascript**. Disponível em <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acessado em 27 de janeiro de 2023B
- NEIRA, RICARDO ALFREDO QUINTANO; PUCHNICK, ANDREA; COHRS, FREDERICO MOLINA; LOPES, PAULO ROBERTO DE LIMA; LEDERMAN, HENRIQUE MANOEL; PISA, IVAN TORRES. **Evaluation of a second opinion system in radiology**. Radiol. bras ; 43(3): 179-183, maio-jun. 2010. ilus, tab
- ORACLE. **O que é um Banco de Dados?** disponível em <https://www.oracle.com/br/database/what-is-database/>. Acessado em 23 de fevereiro de 2023
- PACIEVITCH, Y. C#. **Info Escola**. Disponível em <https://www.infoescola.com/informatica/c-sharp/>. Acessado em 27 de janeiro de 2023

POSTGRESQL. **PostgreSQL: The World's Most Advanced Open Source Relational Database.** Disponível em <https://www.postgresql.org/>. Acessado em 27 de janeiro de 2023

PRESIDÊNCIA DA REPÚBLICA. **Lei Nº 13.709, de 14 de agosto 2018**, Disponível em <[https://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2018/lei/L13709.htm](https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm)>. Acessado em 9 de março de 2023

REGISTRO.BR. **Sobre o Registro.br.** Disponível em <https://registro.br/quem-somos/>. Acessado em 23 de fevereiro de 2023

SOMMERVILLE, I. **Engenharia de Software.** Addison Wesley, 2003.

TRONCOSO, R. Modelo de laudo médico. **Portal Medicina, 2022.** Disponível em <https://portalelemedicina.com.br/modelo-de-laudo-medico-padronizar-entrega>. Acessado em 27 de janeiro de 2023