

SISTEMA PARA GERENCIAMENTO CLÍNICO

Gabriel Rodrigues Garcia; Alexandre Borges Gonçalves²; Mateus de Sousa Valente³; Leonardo Campos de Assis⁴

^{1,2, 3 e 4} Universidade de Uberaba

E-mail do responsável: garcia.gabrielrodrigues1@gmail.com

RESUMO

Atualmente o cenário empresarial é bastante concorrido, principalmente por causa da transformação digital. O acesso à informação e o uso de sistemas informatizados tornaram os clientes cada vez mais exigentes, que buscam produtos com maior qualidade. Sendo assim, a gestão de qualidade se tornou essencial para um que a empresa torne competitiva no mercado e isso é uma realidade principalmente na área da saúde. Existe um grande volume de clínicas médicas espalhadas pelo país, e muitas delas enfrentam problemas com relação à gestão e qualidade. Com isso, esse projeto tem como objetivo, ajudar o processo de gerenciamento de modo simples e objetivo, seguindo a teoria da gestão de qualidade. Construindo de um modelo otimizado de organização de fluxos de dados em um sistema de agendamento de horários, com embasamento na qualidade oferecendo uma melhor gestão durante o acompanhamento médico.

Palavras-chave: gestão de qualidade; sistema de agendamento.

ABSTRACT

Currently the business scenario is quite competitive, mainly because of digital transformation. Access to information and the use of softwares have made customers increasingly demanding, seeking higher quality products. Therefore, quality management has become essential for the company to become competitive in the market and this is a reality especially in the healthcare area. There is a large volume of medical clinics spread across the country, and many of them face problems regarding management and quality. Therefore, this project aims to help the management process be simple and objective, following the theory of quality management. Building an optimized model for organizing data flows in an appointment scheduling system, based on quality, offering better management during medical monitoring.

Keywords: quality management; scheduling system.

1. INTRODUÇÃO

No século XXI, a humanidade vive a era da informação. Em poucos segundos qualquer notícia ou acontecimento é facilmente acessada pelos meios digitais. Os dados através da utilização da rede de computadores mundial, a Internet, todo conhecimento e informações produzidos são disponibilizados para qualquer pessoa em qualquer lugar que tenha acesso a rede. A produção intensa de informações e conhecimentos tanto científicos como tecnológicos é experienciada pela sociedade dando a esta uma característica que podemos chamar de sociedade do conhecimento (Castells, 2002).

Quando se fala em informação nas organizações é natural que também se mencione os Sistemas de Informação (SI) que permitem coletar, armazenar, recuperar e disseminar informações para fins específicos (Turban; Ranier; Potter, 2007).

Deste modo, para empresas, indústrias, consultórios e instituições se manter no mercado de maneira competitiva vem sendo um grande desafio. Principalmente por causa da facilidade com que a informação é acessada. Clientes cada vez mais exigentes, e a possibilidade de expor na internet qualquer tipo de contrariedade com o produto. É motivo suficiente para que a empresa tenha que cuidar da qualidade dos seus produtos. Faz-se necessário um gerenciamento eficaz focado na qualidade. Segundo Prazeres, “Certificação é a atividade de comprovação da qualificação de itens, produtos, serviços, procedimentos, processos, pessoal ou de sistema da qualidade, no todo ou em parte”.

Especificamente, a área da saúde é uma das mais desafiadoras, cuidar de consultórios médicos e geri-los com qualidade é o desafio que este trabalho visa contribuir por meio de sistemas informatizados haja vista que a gestão de qualidade neste contexto se torna imprescindível para qualquer pessoa que queira estar buscando uma excelência na qualidade de seu atendimento. Desta forma, a gestão da qualidade torna-se uma alternativa poderosa para controlar processos e através da contínua melhora de seu funcionamento, está de encontro com a melhoria do atendimento e fazendo com que as expectativas dos clientes sejam superadas. Assim, promovendo uma melhora na empresa aumentando sua competitividade. (Galdámez, Caepinetti; Gerolamo, 2009; Oliveira, Martins, 2008).

Muito se fala do tempo gasto nas atividades do dia-a-dia e sua interferência na produtividade. Levando isso para o ambiente de consultórios e clínicas multiprofissionais, verifica-se que muito tempo é gasto em atividades como no registro e na busca de dados e documentos de pacientes, além do tempo consumido pelo próprio atendimento ou consulta. Segundo Juran e Gryna (1991, p. 43),

[...] para a maioria dos clientes, qualidade relaciona-se às características do produto que atendem suas necessidades.

Além disso, qualidade quer dizer ausência de falhas, bem como um bom serviço ao cliente [...]. Uma definição abrangente para isso é “adequação ao uso.

Ao realizar agendamentos manuais, com auxílio de uma agenda física, o profissional fica preso à marcação de horários de 30 em 30 minutos ou de hora em hora. Já um sistema automatizado, pode realizar agendamentos de procedimentos levando em conta o tempo médio que este pode ser realizado, bem como, uma média baseada nos atendimentos anteriores, diminuindo assim o tempo de espera, otimizando o tempo do profissional e diminuindo o cancelamento do atendimento e assim aumentando a qualidade do serviço. A gestão da qualidade tem como objetivos a padronização de processos e, por meio de planejamento, controle e aprimoramento, a garantia da qualidade de produtos e serviços. Segundo Srdoc, Sluga e Bratko (2005), Lagrosen (2007), Lagrosen e Lagrosen (2003), para se manter sempre à frente dos concorrentes, é necessário que se adote critérios para tomada de decisões, buscando a qualidade dos processos e serviços.

Sistemas de gerenciamento e agendamento manuais são susceptíveis a falhas que podem levar a perda de dados importantes, o que pode ser mais difícil de ocorrer em processos automatizados, pois permitem backups da base de dados em um ou mais locais remotos. Os profissionais que conseguem manter um histórico com todas as informações do paciente, demonstra eficiência e excelência ao realizar seu trabalho, fazendo com que o seu cliente tenha confiança na execução dos procedimentos. Conforme mencionado por Divardim (2017, p. 1):

A busca por qualidade no setor de atendimento ao público em geral deixa de ser um diferencial e se torna essencial, tendo em vista a alta competitividade do mercado e pacientes cada vez mais exigentes.

Quando automatizamos este processo, pode-se criar informações com as médias de tempo de cada procedimento, porém, como cada pessoa pode reagir de um jeito diferente, a experiência de tempo decorrido desta pessoa pode ser utilizada para se agendar outro paciente após o procedimento e com isso minimizar o seu tempo de espera. Precisão nas marcações de consulta e o pouco tempo de espera demonstram respeito, seriedade e principalmente a excelência na prestação de serviços ofertados para as pessoas, o que aumenta a credibilidade no profissional. Este artigo tem como objetivo demonstrar a importância da qualidade e excelência do atendimento ao cliente através do uso do software de Gerenciamento clínico. Uma organização precisa ser gerenciada para uma melhor eficiência, buscando aperfeiçoar a qualidade através dos saberes e das informações adquiridas, utilizando como vantagem competitiva, proporcionando êxitos em suas atividades (North, 2010).

Um Sistema Digital, pode trazer benefícios para uma clínica médica, tais como a coleta e armazenamento de dados serem atividades mais simples, rápidas e eficientes. Fazendo com que, a qualidade da consulta seja o principal foco do

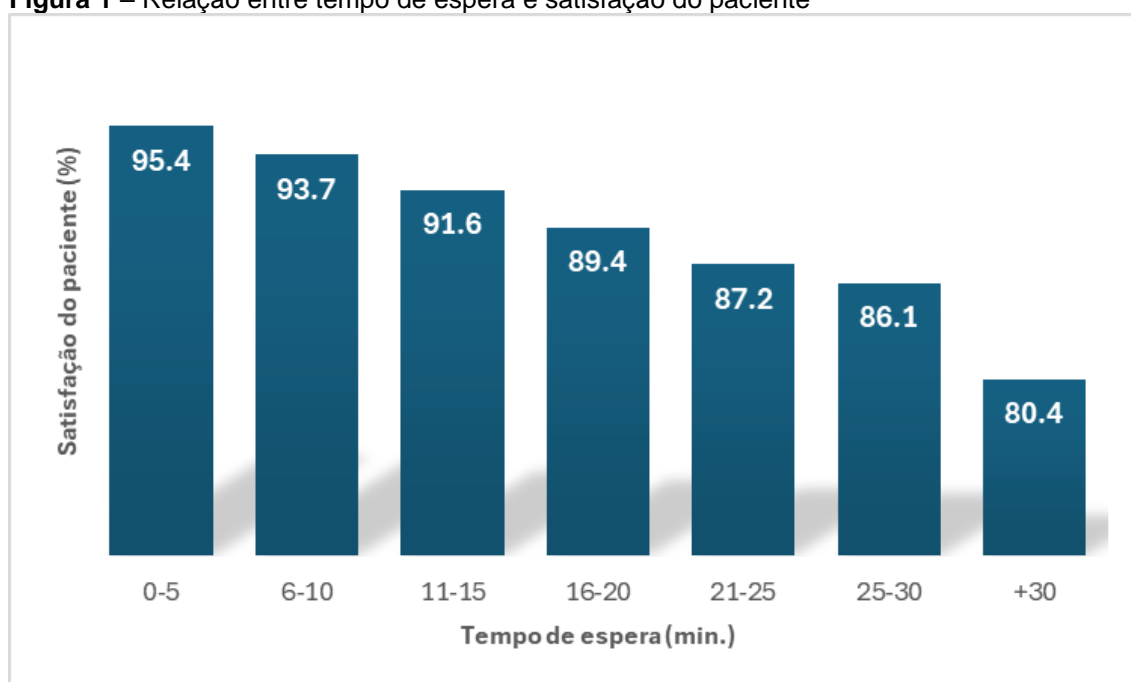
serviço prestado pela clínica, pois a qualidade dos produtos e serviços inclui não apenas sua função e desempenho pretendidos, mas também seu valor percebido e o benefício para o cliente (Associação [...], 2015).

2. REFERENCIAL TEÓRICO

2.1 SETOR DE ATENDIMENTO CLÍNICO

Em um mercado acirrado com pacientes cada vez mais demandantes, a busca por qualidade no setor deixou de ser uma vantagem competitiva e tornou-se imprescindível, sendo assim, a gestão da qualidade é um conceito essencial para qualquer organização que busca se manter competitiva no mercado atual. Segundo a Associação Brasileira de Normas e Técnicas (2015, p. 2), "uma organização focada em qualidade promove uma cultura que resulta em comportamentos, atitudes, atividades e processos que agregam valor através da satisfação das necessidades e expectativas dos clientes e de outras partes interessadas pertinentes". No contexto da área da saúde, as clínicas médicas devem ser eficientes principalmente em relação ao tempo de espera do paciente ao ser atendido. Como mostra o gráfico (Figura 1) a seguir da pesquisa feita pelo setor saúde, quanto mais aumenta o tempo de espera, maior é a insatisfação do paciente. No gráfico é possível observar que um atraso de meia hora, impacta em aproximadamente 20% dos pacientes insatisfeitos.

Figura 1 – Relação entre tempo de espera e satisfação do paciente



Fonte: Adaptado de Setor Saúde (2024).

Sendo assim a gestão da qualidade assume um papel ainda mais crítico quando se trata de clínicas e consultórios médicos. Que necessitam de um gerenciamento eficiente para garantir a segurança e a eficácia dos atendimentos prestados. Sistemas informatizados de gestão clínica são ferramentas valiosas para alcançar esses objetivos. Esses sistemas permitem a automação de processos administrativos e clínicos, reduzindo erros humanos, otimizando o tempo dos profissionais e melhorando a experiência do paciente (Oliveira *et al.*, 2011)

A automação de processos, como o agendamento de consultas, é uma prática que tem mostrado resultados significativos na melhoria da eficiência e da qualidade dos serviços de saúde. Estudos demonstram que a utilização de sistemas automatizados de agendamento pode reduzir significativamente o tempo de espera dos pacientes e aumentar a precisão nas marcações de consulta. Esses sistemas consideram o tempo médio necessário para cada procedimento e ajustam automaticamente a disponibilidade do profissional, baseando-se em dados históricos e na experiência acumulada (Silva *et al.*, 2020).

Outra ferramenta poderosa nesse cenário é a implementação de um sistema de avaliação de satisfação do cliente. Pois utilizar esse tipo de ferramenta possibilita coletar dados sobre a experiência do cliente assim permitindo identificar áreas passíveis de melhorias e executar medidas de correção adequadas. Esses procedimentos não apenas ajudam as empresas minimizarem falhas encontradas na prestação dos serviços ou produção dos seus produtos, mas também elevam os graus satisfatórios declarados pelos clientes para com esses serviços oferecidos pela organização levando-a para excelência nos produtos fornecidos e atendimento aos consumidores.

2.2. SEGURANÇA DE DADOS

É importante lembrar que hoje existe a Lei Geral de Proteção de Dados Pessoais (LGPD), Lei nº 13.709/2018, que foi promulgada para proteger os direitos fundamentais de liberdade e de privacidade, e a livre formação da personalidade de cada indivíduo. A Lei fala sobre o tratamento de dados pessoais, dispostos em meio físico ou digital, feito por pessoa física ou jurídica de direito público ou privado, englobando um amplo conjunto de operações que podem ocorrer em meios manuais ou digitais (Brasil, 2024).

Sendo assim, o projeto deste artigo aborda sobre a segurança da informação. Que também é muito relevante nos dias de hoje, e garante a qualidade do serviço através das diretrizes da LGPD. Pois garantir a segurança e o armazenamento adequado de dados é crucial para o gerenciamento efetivo de clínicas médicas. Perder informações pode ter consequências graves, tanto em termos clínicos quanto legais. Sistemas digitais oferecem vantagens significativas nesse

aspecto ao permitirem *backups* regulares e um armazenamento seguro dos dados em locais remotos. Como relata Cruz *et al.* (2018), digitalizar e automatizar os processos relativos à coleta e guarda das informações do paciente eleva sua confiabilidade, garantindo sua integridade.

Segundo Almeida e Soares (2022):

A Lei nº 13.709/18 - LGPD, se vale da boa-fé e do interesse público, quando se trata da disponibilidade e do acesso aos dados pessoais, resguardados e previsto na referida Lei, [...] 'mediante o fornecimento de consentimento pelo titular' (inciso I do Art. 7º, Lei nº 13.709/18), que, caso venha ocorrer o compartilhamento de dados pessoais com outros controladores, nesse caso, os dados pessoais somente poderão ser compartilhados mediante consentimento do titular para esse fim, [...].

Desta forma, de acordo com a Lei nº 13.709/18 - LGPD, deve-se ter cuidado com relação a alguns registros que necessitam de maior proteção das informações, conhecidos como dados sensíveis, pode-se definir estes como aqueles que identificam o indivíduo e assim no Art 5º, inciso I, essas informações caracterizam a origem racial, religião, política, informações sobre saúde, vida sexual ou dado genético. (Almeida e Soares, 2022).

2.3. ENGENHARIA DE SOFTWARE

Para começarmos falando de Engenharia de Software, podemos desmembrar essas palavras para obter total significado deste estudo. De acordo com o Dicionário Aurélio assim define Engenharia e *software* respectivamente:

Engenharia:

Arte de aplicar conhecimentos científicos e empíricos e certas habilitações específicas à criação de estruturas, dispositivos e processos que se utilizam para converter recursos naturais em formas adequadas ao atendimento das necessidades humanas.

Software:

Reunião dos procedimentos e/ou instruções que determinam o funcionamento de um computador. Conjunto dos elementos que, num computador, compõem o sistema de processamento de dados; todo programa que se encontra armazenado no disco rígido.

A Engenharia de *Software* busca criar produtos em *software*, para atender as necessidades de pessoas e instituições, buscando atingir sempre o maior alto

nível de qualidade, visando tecnologias que sejam as mais adequadas para suprir a necessidade de quem deseja o *software*, usando de processos calculados para extrair maior produtividade de quem o usa (Cortés, 2013).

A modularidade consiste em separar partes de funcionalidades complexas do sistema em partes menores visando contribuir a uma resolução de um problema do sistema de uma forma mais clara e simples, também vale ressaltar que a modularidade ajuda não só na resolução mais simples de problemas, mas também na reutilização de códigos (David, 2024).

O conceito de incrementabilidade é fundamentar o processo de produção em um modo de passo-a-passo, com isso o objetivo final fica mais fácil de ser implementado, pois há uma constante modificação nas funcionalidades até que chegue no resultado esperado (Cortés, 2013).

2.4. FRAMEWORK .NET CORE

A .NET Core é uma evolução da .NET Framework, a implementação original do .NET, é desenvolvida exclusivamente para sistemas Windows, para criação de aplicações, sites e serviços para clientes de área de trabalho e servidores (Microsoft, 2024).

.NET Core é um *framework* de código aberto, multiplataforma, mantido pela Microsoft e pela comunidade de desenvolvedores, sendo uma plataforma de desenvolvimento gratuita usada para a criação de aplicações *web*, *desktop*, móvel, jogos, e IoT e mais.

Esta plataforma foi projetada com o intuito principal de fornecer desempenho, segurança, confiabilidade e principalmente produtividade, incluindo vastas bibliotecas com diversas funcionalidades otimizadas para desempenhar em variados sistemas operacionais (Gewarren, 2024).

.NET Core compartilha muitos componentes do NET Framework e possui compatibilidade com a maioria de seus códigos, e conta muitas melhorias e novidades (Microsoft, 2024).

Entre as melhorias, permite portabilidade de aplicações para diversos sistemas operacionais, expansão para containers, alto desempenho, flexibilidade permitindo a utilização apenas dos componentes necessários, ativamente mantido e atualizado pela Microsoft e pela comunidade. Além da implementação de novas ferramentas para desenvolvimento multiplataforma como Integração com o MAUI (Multi-platform App UI) evolução do Xamarin.Forms, que facilita o desenvolvimento de aplicações móveis e *desktop* com uma única base de código (Gewarren, 2024; Davidbritch, 2024).

2.4.1. C#

A linguagem C# é uma das mais populares para a plataforma .NET, utilizada por milhões de desenvolvedores, ela permite a criação de aplicações para uma variedade de dispositivos, desde IoT até a nuvem (Billwagner, 2024).

C# é uma linguagem de programação de uso geral, sua sintaxe é semelhante a outras linguagens da família C, como C++ e Java.

É uma linguagem fortemente tipada, baseada em princípios orientados a objetos, mas que também incorpora características de outros paradigmas, como orientação a eventos e programação funcional. Ela foi desenvolvida para ser eficiente e segura, oferecendo suporte a cenários de alta eficiência em recursos de baixo nível sem a necessidade de escrever código não seguro (Billwagner, 2024).

C# oferece diversos recursos avançados que incluem gerenciamento automático de memória do .NET Runtime, bibliotecas de tempo de execução fornecidas pelo .NET SDK, além de outras bibliotecas independentes da plataforma como gerenciamento de arquivos (Billwagner, 2024).

Outras cargas de trabalho possuem componentes exclusivos, como a biblioteca ASP.NET Core para desenvolvimento de aplicações *web* e a .NET MAUI para desenvolvimento de aplicações multiplataforma (Billwagner, 2024).

Também conta com um grande ecossistema de open-source NuGet, que aumenta as bibliotecas disponíveis para desenvolvedores, que fazem parte da execução nativa do projeto (Billwagner, 2024).

Possui um extenso suporte a técnicas orientadas a objetos, como herança e polimorfismo. A linguagem também possui suporte robusto a LINQ (Language Integrated Query), programação assíncrona e multi-thread (Billwagner, 2024).

2.4.2. Windows Forms .NET

O Windows Form é uma estrutura de interface do usuário usada na criação de aplicativos na área de trabalho permitindo maneiras mais produtivas de criar interface com através do Designer visual que é fornecido pelo Visual Studio, que permite a criação de interfaces através do método "arrastar e soltar", facilitando a construção de aplicativos com controle preciso sobre a interface do usuário (Adegeo, 2024).

O Windows Forms fornece uma ampla gama de controles pré-construídos, como botões, caixas de texto, menus e muito mais, permitindo a construção rápida e

eficiente de interfaces de usuário complexas. Ele também suporta a criação de controles personalizados, possuindo uma grande coleção de classes para renderizar objetos, como linhas, círculos entre outras formas diretamente nos controles de usuário (Adegeo, 2024).

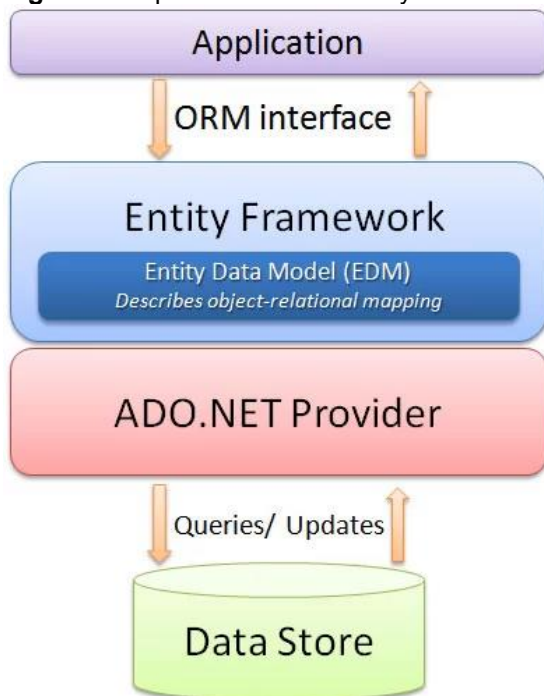
Com essa estrutura, é possível desenvolver sistemas graficamente avançados, com facilidade na sua implementação e atualização. Além disso, as aplicações Windows Form, contam com acesso ao Hardware local e o sistema de arquivos do computador em execução (Adegeo, 2024).

2.4.3. ENTITY FRAMEWORK CORE

O Entity Framework Core é uma evolução do ecossistema Entity Framework, sendo uma versão de código aberto e multiplataforma. (Ajcvickers, 2023)

Esse framework é uma ferramenta da Microsoft permitindo trabalhar com o banco de dados relacional de forma abstrata, fazendo com que o desenvolvimento seja focado muito mais na parte de negócios. Utilizando uma técnica chamada ORM (*Object-Relational Mapping*) (Redspark, 2015).

Figura 2 – Aplicabilidade do Entity Framework



Fonte: Redspark (2024).

De acordo com Redspark, 2015 o ORM é uma técnica para diminuir a impedância da programação orientada a objetos com bancos de dados relacionais, apresentando as tabelas como classes e os registros de cada tabela como registros nas classes correspondentes.

Para lidar com a quantidade de dados proposta, é utilizado o LINQ (Language Integrated Query). O LINQ é uma forma programática de realizar consultas de dados, permitindo que, em vez de escrever SQL como uma *string*, as consultas sejam feitas utilizando uma estrutura de programação que pode ser mais facilmente interpretada pelo compilador e transformada no código necessário automaticamente, sendo uma das formas mais utilizadas por bibliotecas ORMs. Essa abordagem é aplicável a diversas estruturas de dados, como bancos de dados SQL, XML, JSON, entre outros (Billwagner, 2024).

2.5. Model-View-Controller

Em arquitetura de software, o padrão MVC (Model-View-Controller) é amplamente reconhecido pela comunidade, especialmente no contexto de programação orientada a objetos. O sucesso de uma aplicação orientada a objetos está intimamente ligado ao padrão de arquitetura utilizado como base. Conforme destacado por Josué Luciano, padrões de projetos são soluções para problemas que alguém enfrentou e resolveu aplicando um modelo documentado, que pode ser adaptado conforme as necessidades da aplicação. O grande desafio das equipes de desenvolvimento de software é produzir aplicativos seguros, eficientes, de fácil manutenção, reutilizáveis e em prazos cada vez menores.

Na teoria, a aplicação do MVC é bastante simples: todas as requisições da aplicação são verificadas e direcionadas pela camada Controller, que se comunica com o Model para processar a requisição e, conforme necessário, exibir a View. O MVC separa as camadas de gerenciamento de fluxos, lógica e apresentação, facilitando a manutenção e a reutilização de código. O principal objetivo do padrão MVC é isolar ao máximo a camada de apresentação (Luciano, 2024).

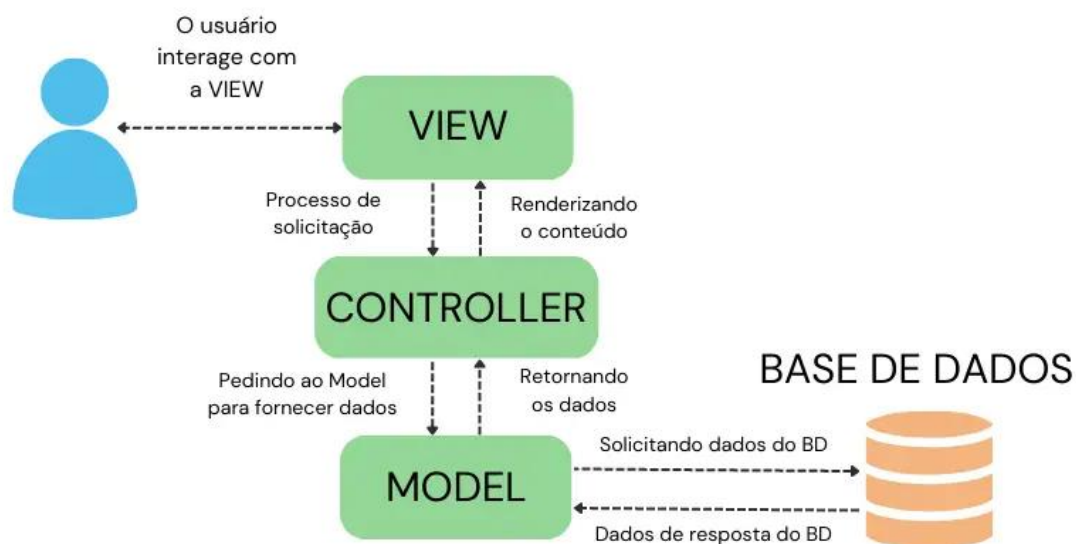
O componente Model é a parte mais fundamental da arquitetura, pois é responsável pela comunicação com os dados que serão visualizados pela View. Na camada Model, são realizadas as operações de negócios, como operações CRUD (Usandopy, 2023).

O Controller administra todo o fluxo de dados da aplicação, movendo os dados para a View com base em uma lógica específica (Usandopy, 2023).

Por fim, a View é definida como a interface visualizada pelo usuário, sendo a camada de apresentação que lida com a comunicação entre o Controller e o Model (Usandopy, 2023).

Abaixo pode ser visualizada uma imagem representando o modelo MVC:

Figura 3 - Modelo MVC



Fonte: Usandopy (2023).

2.6. BANCO DE DADOS RELACIONAL

Um banco de dados relacional é informações organizadas em forma de dados que possuem relação entre si. Esses dados são armazenados em tabelas para ajudar na visualização dos mesmos (Cloud [...], 2024).

Assim como está previsto no Cloud Google (2024, p. 1), “esse relacionamento é definido como a conexão lógica entre determinadas tabelas, estabelecidas com base nas interações entre tabelas”.

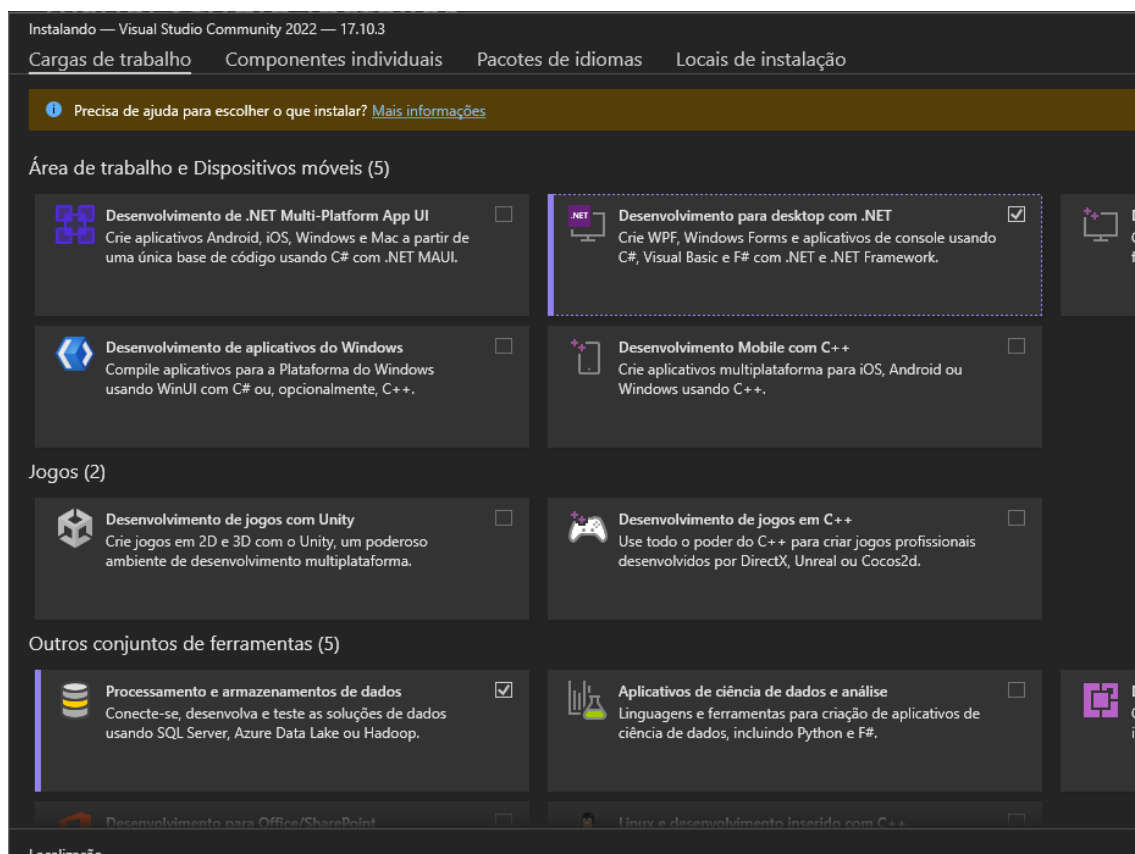
Este modelo foi desenvolvido em 1970 por EF Codd da IBM a fim de parar de usar estruturas hierárquicas para a organização dos dados, propondo permitir que tabelas se relacionam com outras através de atributos em comum. Dentro das tabelas, existe um atributo chamado de chave primária, na qual é um identificador exclusivo para uma linha específica e a partir dessa linha exclusiva é possível criar relação entre tabelas diferentes, recebendo outro atributo conhecido como chave estrangeira (Cloud [...], 2024).

Sua vantagem está na utilização simples, permitindo que qualquer pessoa faça consultas complexas, permitindo uma curva de aprendizado baixa e muito efetiva. Além disso, permite deletar e inserir novos dados sem causar dano na estrutura do banco contribuindo para a flexibilidade (Cloud [...], 2024, p. 1).

3. METODOLOGIA

A plataforma de desenvolvimento escolhida foi o Visual Studio 2022 Community, com as cargas de trabalho, Desenvolvimento para desktop com .NET e Processamento e armazenamento de dados instalados.

Figura 4 – Tela de instalação das cargas de trabalho do Visual Studio.



Fonte: Elaborada pelos autores.

A linguagem escolhida para a execução do projeto foi o C# usando o .NET Core 8 LTS (*Long-term support*), é uma linguagem repleta de recursos e funcionalidades, permite flexibilidade e portabilidade para outros sistemas operacionais e *web* com máxima reutilização de código. A .NET é constantemente mantida e atualizada pela Microsoft, e possui versão LTS, sendo um dos frameworks com melhor suporte e documentação atualmente.

No *front end* foi utilizado estrutura baseada em telas de aplicações *mobile* e *web* onde cada nova tela é empilhada sobre a outra, no caso da aplicação, as telas ou *views* do sistema aparecem todas em um único formulário, para executar essa funcionalidade foi utilizado os componentes UserControl do Windows Forms, cada view é um UserControl exclusivo.

As views *UserControl* estendem de uma interface *IViews* que possui métodos comuns para eventos de atualização, como quando uma view é fechada a anterior a ela deve ser atualizada.

Uma classe para gestão das *views* foi construída, a *ViewsManager*, já que o *Windows Forms* não possui suporte nativo a comunicação de telas empilhadas dentro de um formulário.

A classe *ViewsManager* faz a gestão da dimensão e precedência das *views*, passagem de parâmetros, disparar eventos de criação, fechamento e atualização de *views*.

Como padrão de projeto foi utilizado o padrão MVC (*Model-View-Controller*), algumas separações extras também foram adicionadas, como o *Public*, *Service* e o *DAO (Data Access Object)*.

Objetos que são únicos e compartilhados ficam na pasta *public* como o *ViewManager* e *IViews*.

Componentes reutilizáveis também foram utilizados, como a barra de título de cada view é um *UserControl* com suas funcionalidades básicas já programada que fica presente na pasta *Generic/Components/TopBarPack* onde contém a classe *UserControl* e a classe *Button* customizado.

Também foi criado componentes do menu lateral, barra horizontal estilo em abas, lista de arquivos e o calendário, todos esses componentes customizados ficam em *Generic/Components*, cada um em sua pasta *Pack*.

Também foi criado a pasta *Generic/Objects* onde possui mais objetos customizados de propósito geral, como caixas de texto, botões, renderizadores de imagens e linhas.

Na pasta *View* foram colocados os *UserControls* responsáveis pelas *views* da aplicação, onde nelas ficam toda a responsabilidade de receber os dados do *controller* e formatá-los de forma a exibí-los ao usuário. Nessa pasta também está o formulário principal que exibe as *views*, a tela de login que também é um formulário.

Para o *back end*, foi criado a pasta *Controller*, que foram colocados os *controllers* das *views* que são responsáveis por chamar o *model* e processar o retorno com a regra da aplicação e devolvê-lo a *view*.

Para *Model* foi utilizado uma abordagem diferente, foi colocado em um projeto de Biblioteca de Classes, que pode ser reutilizada em qualquer projeto .NET, sendo assim multiplataforma. A Biblioteca de Classe é compilada em um arquivo DLL (*Dynamic-link library*) que pode ser acoplado como dependência de projetos .NET. Assim permitindo a reutilização de código da regra de negócio para possíveis portes ou migrações de plataforma.

Dentro do projeto do Model, foi criada a pasta Service, DAO e Model, onde Service é responsável pela regra de negócio e buscar os dados do DAO, a DAO é responsável por buscar os dados no banco de dados usando o ORM e a Model é onde ficam os objetos que representam a estrutura do banco de dados no ORM.

Na pasta raiz do projeto do Model também foi colocado a classe DataContext, que é a classe de configuração e conexão do Entity Framework.

No projeto do Model foi colocada a dependência do Entity Framework Core via pacotes NuGet da comunidade de bibliotecas de classes mantida pela Microsoft.

Foi adicionado os pacotes Microsoft.EntityFrameworkCore, Microsoft.EntityFrameworkCore.Design, Microsoft.EntityFrameworkCore.Tools e Pomelo.EntityFrameworkCore.MySql todos na versão 7.0.0, o último sendo um complemento para o MySql e por consequência para o MariaDB, ele não é um complemento oficial da Microsoft, porém é um projeto estável de longa data, possui código aberto e uso livre, mantido por uma quantidade considerável de contribuidores.

Essa abordagem permite além da escalabilidade de ser consumido por outros projetos .NET, também permite o isolamento do ORM do restante da aplicação.

Para o banco de dados foi utilizado o MariaDB um banco de dados relacional, que possui paridade com MySql, é de código aberto e uso gratuito, sendo uma das soluções mais robustas com essas características.

Para a modelagem do banco de dados foi utilizado o método *Code First* do Entity Framework, onde criamos as classes que representam o banco de dados e utilizamos da função *Migration* do Entity Framework que é capaz de criar automaticamente as tabelas e relações no banco de dados. Com essa abordagem reduz a necessidade de configurações manuais do banco de dados, já que Entity Framework consegue na maioria das vezes alterar posteriormente a estrutura do banco de dados sem perda de dados.

Esta abordagem acelera o desenvolvimento inicial, mas deve ter cuidado ao ser utilizada; felizmente o Entity Framework facilita a migração para o método *database first* a qualquer momento com pouco retrabalho no código e sem perda de dados do banco.

Outro projeto de Biblioteca de Classes foi implementado, uma biblioteca Utils, que possui utilitários para o projeto, como ferramenta de criptografia *hash* SHA256 e validação do hash, e ferramenta de manipulação de arquivos *json*.

O projeto do Model faz uso da biblioteca Utils para criptografar senha dos usuários e validá-los, também utiliza o *json* para armazenar dados de configurações locais como o endereço do banco de dados.

4. Resultados

O sistema já é capaz de realizar suas funcionalidades na parte de gestão de qualidade a partir do momento em que você faz o login, pois nele é feita a validação do responsável pelo login e atrela as próximas informações de forma correta.

Figura 5 – Tela inicial da aplicação

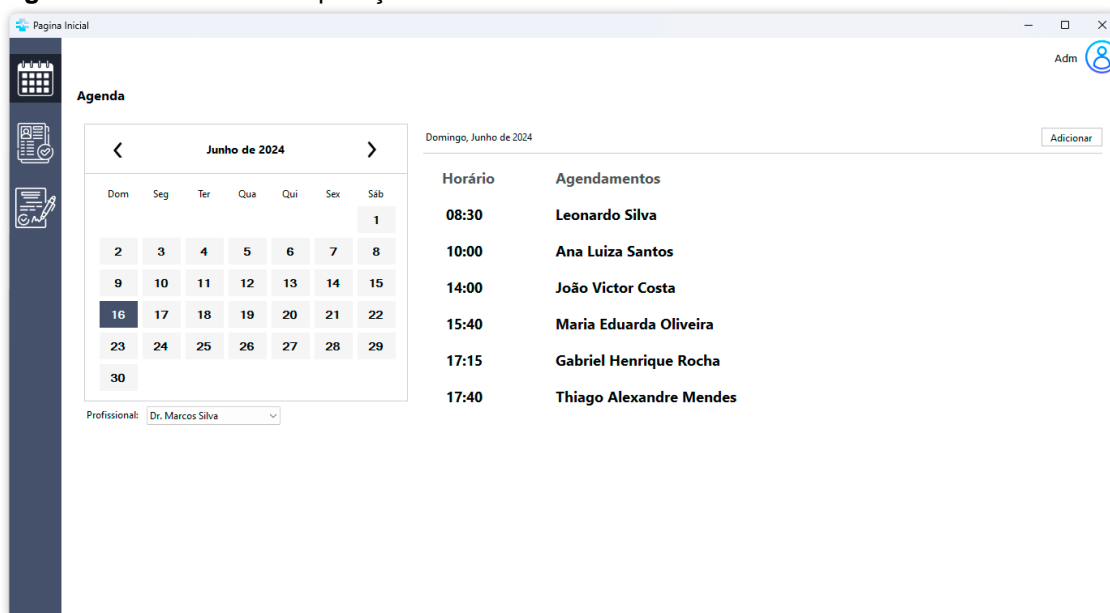


A imagem mostra uma janela de login com o título "Login". À esquerda, há um ícone de usuário em azul. À direita, há campos de entrada para "Usuário:" e "Senha:", e um botão "Entrar".

Fonte: Elaborada pelos autores.

Usando o *Windows forms* para confecção da tela e o restante das tecnologias anteriormente citadas, foi criada uma tela na qual é permitido a realização do agendamento.

Figura 6 – Tela inicial da aplicação



Fonte: Elaborada pelos autores.

Esta tela está programada para apenas mostrar informações relativas ao profissional que faça *login*. Nessa parte é possível visualizar quem são os pacientes do dia, renunciando o uso de papéis, nos quais podem não estar organizados e de alguma forma em um lugar inseguro, prejudicando os pacientes com perda de seus dados sensíveis.

Na tela de adicionar, além de suspender o uso manual de agendamento, faz uma análise sobre a média sobre o paciente que procura uma consulta, localizando o dia e o horário mais adequado, deixando a qualidade da consulta sempre em primeiro lugar.

Esse sistema informatizado também supre a falta de informação dos pacientes que os profissionais necessitam, pois a partir do primeiro cadastro é recolhida observações médicas dos pacientes, juntamente com documentos de consultas anteriores, receitas prescritas, dentre outros tipos. O responsável pelos prontuários consegue editar tais mecanismo, possibilitando deixar a consulta o mais fluida possível, pois seus dados estarão sempre atualizados.

Figura 7 – Tela criação de agendamento

The screenshot shows a web application window titled 'Pagina Inicial'. On the left is a dark blue sidebar with three icons: a calendar, a document with a checkmark, and a document with a pencil. The main content area is titled 'Criar Agendamento' with a back arrow. It contains several form fields: 'Nome do paciente' (a dropdown menu), 'Status' (radio buttons for 'Confirmado', 'Pendente', and 'Cancelado'), 'Observações' (a large text area), and 'Profissional' (a dropdown menu). On the right side, there is a 'Data' section with a date picker set to '27/06/2024', a calendar for 'Junho de 2024' (with the 27th highlighted), and a time range selector with 'De' set to '09:00:00' and 'Até' set to '10:30:00'. A user profile icon labeled 'Adm' is in the top right corner.

Fonte: Elaborada pelos autores.

Para que o tempo da consulta seja sempre muito objetivo, foi posto uma tela na qual documentos tais como receitas, já estarão pré-preenchidas e logo encaminhadas para pacientes via *e-mail* se for o caso. Isso contribui para uma rápida interação final entre o médico e paciente e também ajuda o paciente caso perca a guia que antes estava escrita em forma física.

Figura 8 – Tela de documentos diversos

The screenshot shows a web application window titled 'Pagina Inicial'. On the left is a dark blue sidebar with three icons: a calendar, a document with a checkmark, and a document with a pencil. The main content area is titled 'Documentos' and contains three rounded rectangular buttons: 'Arquivo1' with a document icon, 'Arquivo2' with a document icon, and 'Adicionar' with a plus sign icon. A user profile icon labeled 'Adm' is in the top right corner.

Fonte: Elaborada pelos autores.

O sistema através das tecnologias utilizadas juntamente dessa análise em vários parâmetros para gerenciar ao máximo a qualidade do ambiente clínico, pode contribuir para a melhora em redes clínicas de todas as áreas.

5. Conclusão

A criação de um sistema informatizado voltado para clínica com inclusão de uma gestão de qualidade, de fato é uma solução para a carência no atendimento na área da saúde. Isso porque, com a utilização deste sistema, problemas gerados por atraso de paciente, perda de prontuários é algo que afeta muitas clínicas e por consequência seus clientes.

Através do agendamento gerenciado pelo sistema e toda análise que foi feita para trazer melhor qualidade para os clientes do consultório, é possível que esses problemas enfrentados diariamente possam ser diminuídos drasticamente, fazendo com que de fato o sistema seja a solução que o mercado precisa.

O resultado que obtivemos durante as pesquisas e construção do sistema foram positivos, pois é uma lacuna a ser preenchida pela área de gestão e conseguimos contribuir de fato para preenchê-la.

REFERÊNCIAS

ADEGEO, *et al.* **Guia da área de trabalho (Windows Forms .NET)**. 2023
Disponível em: <https://learn.microsoft.com/pt-br/dotnet/desktop/winforms/overview/?view=netdesktop-8.0>. Acesso em: jun. 2024.

AJCVICKERS, *et al.* **Entity Framework Core**. Vale do Silício: Microsoft, 2023.
Disponível em: <https://learn.microsoft.com/pt-br/ef/core/>. Acesso em: jun. 2024.

ALMEIDA, Siderly do Carmo Dahle de; SOARES, Tania Aparecida. **Os impactos da Lei Geral de Proteção de Dados - LGPD no cenário digital**. Perspectiva em Ciência da Informação. Dezembro de 2022.

ASSOCIAÇÃO BRASILEIRA DE NORMAS E TÉCNICAS. **NBR ISO 9000**. São Paulo: ABNT, 2015.

BERG, Ernesto. **Por que as Empresas perdem Clientes?** Revista e Portal Meio Filtrante. Edição Nº 66 - Janeiro/Fevereiro de 2014 - Ano 12. (<https://www.meiofiltrante.com.br/Artigo/1005/por-que-as-empresas-perdem-clientes>. Acesso em 20/06/2024).

BILLWAGNER. **Visão geral:** um tour pela linguagem C#. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/overview>. Acesso em: jun. 2024.

BONATO, Vera Lúcia. **Gestão de qualidade em saúde:** melhorando a assistência ao cliente. O Mundo da Saúde, São Paulo: 2011. (https://bvsmms.saude.gov.br/bvs/artigos/gestao_qualidade_saude_melhorando_assistencia_cliente.pdf. Acesso em 20/06/2024).

BRASIL. **Lei Geral de Proteção de Dados Pessoais (LGPD)**. Brasília: Ministério dos Esportes, 2024.

CASTELLS, M. **A sociedade em rede:** a era da informação 6. ed. São Paulo: Paz e Terra, 2002. v. 1.

CHIAVENATO, Idalberto. **Empreendedorismo:** dando asas ao espírito empreendedor. São Paulo: Saraiva, 2007.

CLOUD GOOGLE. O que é um banco de dados relacional? Vale do Silício: Google, 2024. Disponível em: <https://cloud.google.com/learn/what-is-a-relational-database?hl=pt-br>. Acesso em: jun. 2024.

CORTÉS, Mariela Inés. **Informática:** fundamentos de engenharia de software. Fortaleza: Secretaria de Apoio às Tecnologias Educacionais, 2013. Disponível em: <https://www.uece.br/cct/wp-content/uploads/sites/28/2021/07/Fundamentos-de-Engenharia-de-Software.pdf>. Acesso em: jun. 2024.

CRUZ, F. A., OLIVEIRA G. R., & SANTOS, H. M.. **Segurança da Informação em Clínicas Médicas:** Aspectos e Benefícios da Digitalização e Automação de Processos. Jornal Brasileiro de Informática em Saúde, 2018.

DAVID, ORTIZ DE ARCANJO ANTONIO **Modularização de Software**. 2023. Disponível em: <https://dev.to/ortizdavid/modularizacao-de-software-508>. Acesso em: jun. 2024

DAVIDBRITCH, *et al.* **O que é o .NET MAUI?**. Vale do Silício: Microsoft, 2024. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/maui/what-is-maui?view=net-maui-8.0>. Acesso em: jun. 2024.

DIVARDIN, Giulia. **Desenvolvimento de um sistema de gestão da qualidade para uma clínica odontológica**. 2017. 61 f. Trabalho de Conclusão de Curso (Graduação) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2017

GALDÁMEZ, E. V. C.; CARPINETTI, L. C. R., GEROLAMO, M. C. Proposta de um sistema de avaliação do desempenho para arranjos produtivos locais.

Gestão & Produção, v. 16, n. 1, p. 133-151, 2009.
<http://dx.doi.org/10.1590/S0104-530X2009000100013>.

GEWARREN. **.NET (e .NET Core) – introdução e visão geral**. 2024.
Disponível em: <https://learn.microsoft.com/pt-br/dotnet/core/introduction>.
Acesso em: jun. 2024.

GOOGLE. **O que é um banco de dados relacional?**. [s.d.] Disponível em:
<https://cloud.google.com/learn/what-is-a-relational-database?hl=pt-br#:~:text=Um%20banco%20de%20dados%20relacional%20>. Acesso em: jun. 2024.

JANNUZZI, Celeste Aída Sirotheau Corrêa; FALSARELLA, Orandi Mina; SUGAHARA, Cibele Roberta. **Sistema de Informação**: um entendimento conceitual para a sua aplicação nas organizações empresariais. Perspectivas em Ciências da Informação. Janeiro de 2015.
(<https://www.scielo.br/j/pci/a/fKbBSPKSPdN6XbSkfyGMKMK/#>. Acesso em 18/06/2024)

JURAN, J. M.; GRYNAL, F. M. **Controle da qualidade**: handbook - conceitos, políticas e filosofia da qualidade. São Paulo: Makron/McGraw-Hill, 1991.

LAGROSEN, S. Quality management and environment: exploring the connections. *International Journal of Quality & Reliability Management*, v. 24, n. 4, p. 333-346, 2007. <http://dx.doi.org/10.1108/02656710710740527>

LAGROSEN, S.; LAGROSEN, Y. **Quality configurations**: a contingency approach to quality management. *International Journal of Quality & Reliability Management*, v. 20, n. 7, p. 759-773, 2003.
<http://dx.doi.org/10.1108/02656710310491203>

LAGROSEN, Y.; BACKSTRON, I.; LAGROSEN, S. **Quality management and health**: a double connection. *International Journal of Quality & Reliability Management*, v. 24, n. 1, p. 49-61, 2007.
<http://dx.doi.org/10.1108/02656710710720321>

LAGROSEN, Y.; LAGROSEN, S. **The effects of quality management**: a survey of Swedish quality professionals. *International Journal of Operations & Production Management*, v. 25, n. 10, p. 940-952, 2005.

LUCIANO, Josué; ALVES, Wallison Joel Barberá. **Padrão de Arquitetura MVC Model - View – Controller**. Centro Universitário UNIFAFIBE – Bebedouro – SP. Revista EPeQ Fafibe, 3ª. Ed., Vol. 01 – 2011.
(<https://unifafibe.com.br/revistasonline/arquivos/revistaepqfafibe/sumario/20/16112011142249.pdf>). Acesso em 18/06/2024).

MICROSOFT. **O que é .NET framework?** Vale do Silício: Microsoft, 2024. Disponível em: <https://dotnet.microsoft.com/pt-br/learn/dotnet/what-is-dotnet-framework>. Acesso em: jun. 2024.

NETO, Antonio Quinto. **Melhoria na Espera em Sala de Recepção:** Um Tópico de Excelência. Blog: Setor Saúde. (https://setorsaude.com.br/antonioquinto/2015/08/25/melhoria-na-espera-em-sala-de-recepcao-um-topico-de-excelencia/#google_vignette. Acesso em 18/06/2024).

OLIVEIRA, José Augusto de; NADAE, Jeniffer de; OLIVEIRA, Otávio José de; SALGADO, Manoel Henrique. **Um estudo sobre a utilização de sistemas, programas e ferramentas da qualidade em empresas do interior de São Paulo.** UNESP, Bauru, SP. Produção, v. 21, n. 4, p. 708-723, out./dez. 2011. Doi: 10.1590/S0103-65132011005000044. (<https://www.scielo.br/j/prod/a/FZnGc9F8sDG8Bj6rJb9rPmp/?format=pdf&lang=pt>. Acesso em 20/06/2024).

Prazeres, P M. **Dicionário de termos da qualidade.** São Paulo: Atlas; 1996.

REDSPARK. T. **Entendendo o entity framework.** São Paulo: Redspark, 2015. Disponível em: <https://www.redspark.io/entendendo-o-entity-framework>. Acesso em: jun. 2024.

RODRIGUES, C; BLATTMANN, U. **Gestão da informação e a importância do uso de fontes de informação para geração de conhecimento.** Perspectivas em Ciência da Informação, Belo Horizonte, v. 19, n. 3, p.4-29, jul./set. 2014.

ROSSI, Carlos Alberto Vargas; SLONGO, Luiz Antônio. **Pesquisa de satisfação de clientes:** o estado-da-arte e proposição de um método brasileiro. <https://www.scielo.br/j/rac/a/nwpcfNHVMj4LYjbrxbQmCFNS/#>. Publicado em 24 de abril de 2009.

SETOR SAÚDE. **Melhoria na espera em sala de recepção:** um tópico de excelência. Porto Alegre: Setor Saúde, 2015. Disponível em: https://setorsaude.com.br/antonioquinto/2015/08/25/melhoria-na-espera-em-sala-de-recepcao-um-topico-de-excelencia/#google_vignette. Acesso em: jun. 2024.

SILVA, A. B., SOUZA, C. D., & LIMA, E. F. (2020). **Eficácia dos Sistemas Automatizados de Agendamento na Melhoria dos Serviços de Saúde.** Revista de Gestão em Saúde, 2020.

SRDOC, A.; SLUGA, A.; BRATKO, I. A quality management model based on the “deep quality concept”. **International Journal of Quality & Reliability Management**, v. 22, n. 3, p. 278-302, 2005. <http://dx.doi.org/10.1108/02656710510582499>

TURBAN, E.; RANIER JR., R. K.; POTTER, R. E. **Introdução a sistemas de informação**: uma abordagem gerencial. Tradução Daniel Vieira. Rio de Janeiro: Elsevier, 2007.

USANDOPY. **O que é MVC?** entenda arquitetura de padrão MVC. São Paulo: Usando Py, 2023. Disponível em: <https://www.usandopy.com/pt/artigo/o-que-e-mvc-entenda-arquitetura-de-padrao-mvc/>. Acesso em: jun. 2024.