**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**

**Faculdade de Tecnologia de Jundiaí – “Deputado Ary Fossen”**

**Curso Superior de Tecnologia em (Gestão de Tecnologia da Informação)**

**Nicolas Briz de Siqueira**

**Reginaldo Benedicto de Oliveira**

**CADASTRO DE CLIENTES E AGENDAMENTO: INTEGRAÇÃO COM SISTEMA DE SOFTWARE COM BANCO DE DADOS**

**Jundiaí**

**2024**

**Nicolas Briz de Siqueira**

**Reginaldo Benedicto de Oliveira**

**CADASTRO DE CLIENTES E AGENDAMENTO: INTEGRAÇÃO COM SISTEMA DE SOFTWARE COM BANCO DE DADOS**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Jundiaí - “Deputado Ary Fossen” trabalho apresentado para conclusão da Disciplina de Projeto Interdisciplinar II e III, sob a orientação do(a) Professor Wagner Luiz de Andrade Júnior.

Este trabalho é

dedicado aos professores e

alunos da Fatec Jundiaí.

**RESUMO**

Este trabalho tem como princípio a criação de um sistema de cadastro de clientes para agendamentos, como nota em nosso projeto interdisciplinar, contendo conteúdos e explicações das matérias de Engenharia de Software e Banco de Dados. Foi feita uma pesquisa para a construção deste sistema, utilizando o github para controle e acesso para todos os integrantes; foi explicado neste trabalho as funções relacionais e não relacionais envolvidas no nosso banco de dados, a utilização de phython como também a do VScode integrando assim com o terminal de banco de dados escolhido por nossa parte, podendo ser tanto o “MySql” como “MSql”. Vai ser explicado também a análise de requisitos do nosso software e seu ciclo de vida, como nós separamos nosso tempo para realizar este trabalho, e como gerenciamos nosso projeto. Além disso, aplicando testes constantes para validar sua funcionalidade e garantir a segurança dos dados ali presentes mantendo a integridade dos dados. Também foram feitos ajustes ao longo do caminho para que o sistema funcionasse de maneira eficiente.

Palavras-Chave: Cadastro de Clientes, agendamentos**abstract**

This work has as its principle the creation of a customer registration system for appointments, as a note in our interdisciplinary project, containing content and explanations of Software Engineering and Database subjects. Research was carried out to build this system, using github for control and access for all members; This work explained the relational and non-relational functions involved in our database, the use of phython as well as VScode, thus integrating with the database terminal chosen by us, which can be either “MySql” or “MSSql Server”. The requirements analysis of our software and its life cycle will also be explained, how we allocate our time to carry out this work, and how we manage our project. Furthermore, applying constant tests to validate its functionality and guarantee the security of the data present there while maintaining data integrity. Adjustments were also made along the way to make the system work efficiently.

Key words: Customer registration, appointments.

# Lista de imagens

Imagem 1 = Comentários colocados no arquivo README dentro da Branch principal do Git.

Imagem 2 = Diagrama de tabelas e como elas se conectam.

Imagem 3 = Teste padrão cucumber para tabela de clientes.

Imagem 4 = Teste padrão cucumber para tabela de profissionais.

Imagem 5 = Teste padrão cucumber para tabela de agendamentos.

Imagem 6 = Implementação de tabela com a biblioteca prettytable.

Imagem 7 = Demonstração da tabela apresentado em tela.

Imagem 8 = Implementação para coletar OS e limpar o terminal corretamente.

Imagem 9 = Implementação que une data e hora separados, compondo uma variável de tipo *datetime*

# Sumário

[Lista de imagens 5](#_Toc183037184)

[Sumário 6](#_Toc183037185)

[Introdução 7](#_Toc183037186)

[1. Conceitos essenciais sobre Engenharia de software 8](#_Toc183037187)

[1.1Análise e especificação de requisitos 8](#_Toc183037188)

[1.2 Compatibilidade: 10](#_Toc183037189)

[1.3 IMPLEMENTAÇÃO 11](#_Toc183037190)

[2 ciclo de vida do projeto 12](#_Toc183037191)

[3.CONCEITOS ESSENCIAIS REFERENTES À BANCO DE DADOS: 13](#_Toc183037192)

[3.1 implementações e bibliotecas 16](#_Toc183037193)

[3.2 SGDBs 18](#_Toc183037194)

[4.CONCLUSÃO 20](#_Toc183037195)

[5. Revisão Bibliográfica (ou título apropriado) 21](#_Toc183037196)

[6. Referências 22](#_Toc183037197)

# Introdução

Este trabalho tem como objetivo apresentar a aplicação do software e a integração realizada no banco de dados com a criação de agendamentos de cliente e o seu acesso. O trabalho foi desenvolvido com base nos aprendizados das matérias de Engenharia de Software e Banco de dados com intuito de aplicar as teórias e técnicas aprendidas.

A escolha do tema se justifica pelo propósito encontrado ao incialmente apresentado o escopo deste projeto, fazendo assim um cadastro de agendamento, para futuramente poder ser capaz de apresentar esta ideia para pequenas empresas que tenham dificuldade em fluxo de cadastros, garantindo a segurança dos dados de quem entra e de fácil modificação e suporte caso necessário.

Utilizamos a linguagem de programação Phython juntamente com um acesso ao banco de dados na web local Microsoft SQL server, devido a uma dificuldade encontrada ao tentar utilizar o servidor de nome “XAMPP” em que ele não iniciava em nenhuma máquina, impossibilitanto seu uso para integrar as aplicações com os bancos de dados.

Este trabalho está separado em partes, primeiramente explicando os conceitos utilizados de engenharia de software, explicado separadamente, em seguida está apresentado conceitos utilizados como aplicador de banco de dados.

# Conceitos essenciais sobre Engenharia de software

## 1.1Análise e especificação de requisitos

Na etapa de Análise de Requisitos, o foco é entender e documentar o que o sistema precisa fazer. Para isso, realizamos entrevistas com usuários finais e gerentes, buscando captar as necessidades reais do projeto. Nesse processo, desenvolvemos histórias de usuário que abordam tanto o cadastro de clientes quanto o agendamento de serviços.

Os requisitos funcionais que surgem incluem, por exemplo, a possibilidade de cadastrar informações dos clientes, como nome, CPF, telefone e e-mail. Também é essencial que o sistema permita agendar serviços, com opções para selecionar data, hora e tipo de serviço. Além disso, garantir que o usuário possa consultar e gerar relatórios sobre os agendamentos é fundamental para um bom gerenciamento. Essa análise serve como base para as próximas etapas do desenvolvimento, garantindo que tudo esteja alinhado com as expectativas dos usuários.

Para os requisitos funcionais entre eles temos:

O cadastro de clientes, onde o sistema deve permitir que os usuários cadastrem novos clientes incluindo informações igual previamente ditas. Edição ou exclusão destes mesmos clientes.

A forma como vai agendar os horários, escolhendo a data e a hora, como também visualizar todos os agendamentos, podendo utilizar vários tipos de filtros e classificações.

Ser notificado quando um agendamento ocorre é de extrema importância, para manter controle e sempre acompanhar a atualização em primeira mão do banco de dados.

Para a parte dos requisitos não funcionais, aqueles que explicam o comportamento do sistema, como se diz, “Requisitos não funcionais são essenciais para garantir que um sistema não apenas faça o que é necessário, mas que também o faça de maneira eficiente, segura e agradável."- SOMMERVILLE, Ian. *Software Engineering*. 9. ed. Boston: Addison-Wesley, 2011, temos:

Desempenho: Deve-se ser capaz de processar várias entradas de usuários e agendamentos dentro de um período de dois segundos.

Sua interface deve ser intuitiva e fácil de navegar, sua usabilidade do sistema deve ser possível até para aqueles com pouca experiência, não limitando as pessoas que seriam capazes de utilizar.

“O conceito de usabilidade busca identificar maneiras de projetar uma interface intuitiva, permitindo que o usuário mais leigo tenha uma curva de aprendizado bem pequena ao utilizar o sistema.

Deve ser capaz de proteger as informações dos usuários e clientes, utilizando criptografia e autenticação de usuários.” DRANKA, Marcos Andrei. Sistema auxiliar para comparação estrutural de diferentes bancos de dados no processo de integraçãode banco de dados. Florianópolis, SC, 2023. Trabalho de Conclusão de Curso (TCC) — [UFSC] p.17

Escalabilidade, refere-se à capacidade do sistema de suportar um aumento de número de usuários e dados sem comprometer sua performance.

## 1.2 Compatibilidade:

Com relação a um último requisito, temos a questão da compatibilidade. É necessário que seja capaz de poder acessar o sistema de vários navegadores, seja no Chrome, Firefox, Edge e ser bastante responsivo em todos, como também estar adaptado a diversos dispositivos como desktop, tablets ou smartphones. Isso significa que a interface deve se ajustar, redimensionando e reorganizando as informações para tornar a navegação mais simples e prática para cada usuário.

Além disso, é fundamental que o sistema siga padrões web e boas práticas de desenvolvimento, garantindo que todas as funcionalidades estejam acessíveis em qualquer plataforma. Com essa abordagem, não só deixamos melhor a usabilidade, mas também proporcionamos uma experiência mais agradável, aumentando a satisfação dos usuários fazendo com que eles adotem o sistema.

## 1.3 IMPLEMENTAÇÃO

Na parte de implementação do livro "Integrating Software and Database Systems", Simon Graham enfatiza a importância de um planejamento cuidadoso antes de começar a codificação. Ele sugere que é fundamental "definir claramente os requisitos e escolher as tecnologias certas", além de estabelecer um cronograma que seja realista. Graham defende uma abordagem incremental para a implementação, onde as etapas são realizadas uma a uma. Isso permite testar e validar cada parte do sistema antes de passar para a próxima, o que, segundo ele, "reduz os riscos de falhas".

O autor destaca a importância de documentar todo o processo de implementação, desde as decisões tomadas até as configurações usadas. Isso não só ajuda na manutenção futura, mas também facilita a integração de novos desenvolvedores na equipe. Graham enfatiza que "a coleta contínua de feedback dos usuários é essencial para aprimorar a implementação", permitindo que ajustes e melhorias sejam feitos ao longo do tempo. Assim, sua abordagem fornece uma visão clara e prática sobre como tornar a implementação de sistemas integrados mais eficaz e menos propensa a erros.

Teste:

Se fosse necessária uma longa explicação antes do teste, isso já é um indício de que a usabilidade deixa a desejar. No próprio sistema há um card com algumas explicações básicas sobre o uso, para que qualquer usuário possa entender rapidamente o funcionamento. DRANKA, Marcos Andrei. *Sistema auxiliar para comparação estrutural de diferentes bancos de dados no processo de integração de banco de dados*. Florianópolis, SC, 2023. Trabalho de Conclusão de Curso (TCC) — [UFSC] p.39

# 2 ciclo de vida do projeto

O ciclo de vida do projeto abrange todas as etapas, desde o levantamento inicial de requisitos até a manutenção contínua do sistema. Cada uma dessas fases é essencial para garantir que o projeto resulte em uma solução eficaz e robusta, que integre de forma fluida o sistema de agendamento com o banco de dados. Seguir um ciclo de vida bem definido é crucial para assegurar que todos os requisitos do sistema sejam atendidos e que ele seja capaz de oferecer uma experiência de uso segura, eficiente e escalável, atendendo às necessidades e expectativas dos usuários finais.

Para a realização deste projeto, foi feito um repositório na plataforma Github, para poder ter o acompanhamento do progresso de todos os participantes.

Foi realizado 2 “branchs” ou galhos no repositório para ter controle das ações feitas.

É localizado no galho principal a versão finalizada do código, aquela para demonstração, depois de todos os testes e versões, aqui vai ser onde o final das interações reside. Por outro lado, o outro galho define a zona de desenvolvimento dos códigos, onde é construído, mudado, atualizado, commitado todas as mudanças realizadas.

Para alguns pontos de entendimento do código, foi utilizado os comentários no arquivo README.md, pois apresenta uma melhor qualidade de acompanhamento visto aos objetivos e definições do sistema como um todo.

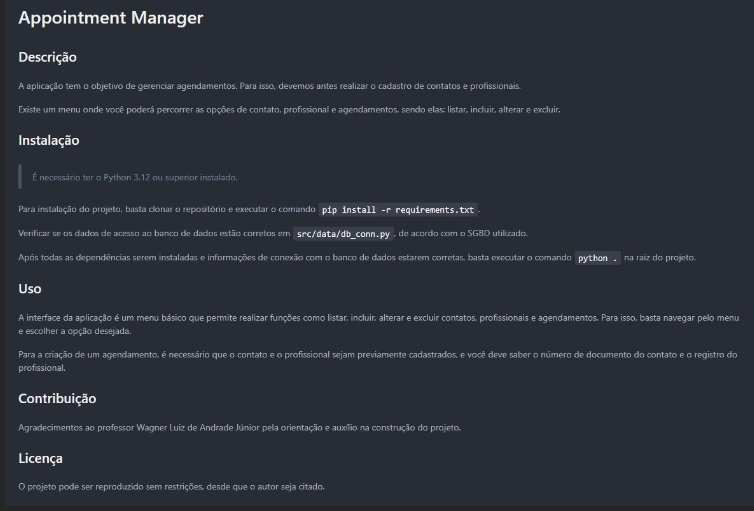


Imagem 1

# 3.CONCEITOS ESSENCIAIS REFERENTES À BANCO DE DADOS:

Para a explicação das relações entre as tabelas do banco de dados, pode-se visualizar a tabela a seguir, onde a conexão entre as tabelas é onde as informações são passadas e traduzidas para os acessos e acompanhamento dos dados.

Com o dado criado na tabela (contato) “ID”, criamos uma normalização de dados a serem puxados para relacionar a outra tabela, deixando as informações sem o perigo de possuírem duplicatas ou até mesmo nomes que confunda o código a rodar.

Lendo este “ID” do contato, que seria o usuário assim que ele adicionou todas suas informações, na tabela de agendamentos separa a data escolhida, hora escolhida, o profissional que vai atender o usuário (que também é definido por um ID próprio para normalizar na hora de puxar os dados), e deixar tudo de fácil entendimento na hora do acesso dos profissionais na hora de registrar e constar no banco os agendamentos realizados.

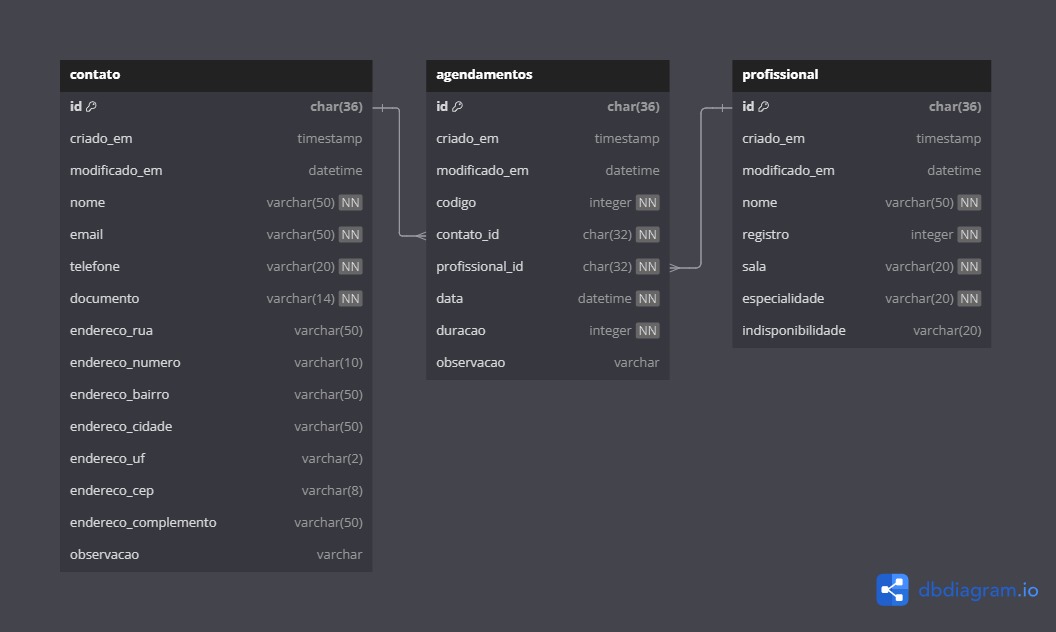


Imagem 2

Com os testes (escritos em Cucumber, padronização para automação), verifica-se se o software se comporta como o esperado. Foram criados conjuntos de cenários de teste para cada funcionalidade, descrevendo os pré-requisitos, ações, e comportamentos esperados.

O primeiro conjunto de testes as os comportamentos esperados do gerenciador de contatos.

Texto

Descrição gerada automaticamente

Imagem 3

Verifica que este conjunto de cenários descreve o que usuário pode esperar do sistema assim que este realizar as ações no sistema.

Este tipo de padronização serve para mostrar cada tipo de cenário dentro do código, é uma forma de descrever os testes.

Em segundo ponto, testa-se o funcionamento da tabela de agendamentos: Texto

Descrição gerada automaticamente  
Imagem 4

Texto

Descrição gerada automaticamente  
Imagem 5

E aqui é o último e teste final com a tabela de profissionais

## 3.1 implementações e bibliotecas

Para o desenvolvimento da aplicação, foi necessária a utilização de algumas bibliotecas externas de outros autores, pois elas nos trazem facilidade na realização de algumas rotinas que pode vezes são impossíveis desenvolver manualmente, ou outras quais tornam o trabalho mais rápido, o que nos faz ter tempo para ter atenção em outros pontos mais importantes do sistema.

**Bibliotecas externas** precisaram ser baixadas utilizando o comando ‘*pip install*’, pois não fazem parte da instalação do Python:

* mysql-connector-python: Driver para realização de conexão e execução de operações CRUD no banco de dados em um SGBD MySQL;
* pyodbc: Driver para realização de conexão e execução de operações CRUD no banco de dados em um SGBD Microsoft SQL Server;
* prettytable: Biblioteca para construção de tabelas dinâmicas, estas que exibem os dados de lista de forma mais apresentável para o usuário.

Abaixo, segue a implementação de uma tabela usando a biblioteca prettytable, e em seguida, como ela é apresentada em tela

Texto

Descrição gerada automaticamente Imagem 6

Texto

Descrição gerada automaticamente  
Imagem 7

As bibliotecas externas utilizadas estão descritas na raiz do projeto, no arquivo *requirements.txt*, o qual auxilia na instalação da aplicação, permitindo que sejam instaladas todas as bibliotecas necessárias para funcionamento da aplicação com o comando ‘*pip install -r requirements.txt*’.

**Bibliotecas internas** também foram necessárias, estas não precisaram ser instaladas da internet, mas fazem parte da instalação do Python:

* os: Biblioteca que contém um método (system) que informa o sistema operacional qual está sendo utilizado, pois é necessário para saber qual comando utilizar no momento de limpar a tela para montagem de um novo menu.
* datetime: Biblioteca que contém um método (strptime) que converte data e hora em formatos de texto, em uma data em formato datetime, apta a ser armazenada no banco de dados.



Imagem 8



Imagem 9

## 3.2 SGDBs

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é um conjunto de programas que tem como principal função gerenciar, armazenar e organizar os dados de um banco de dados. Ele permite que os usuários realizem operações como criação, leitura, atualização e exclusão de informações de forma eficiente e segura. O SGBD funciona como um intermediário entre o usuário e o banco de dados, facilitando o acesso e a manipulação dos dados, sem que o usuário precise se preocupar com os aspectos técnicos de como os dados são armazenados.

Os SGBDs são fundamentais para garantir que os dados sejam mantidos íntegros, seguros e consistentes. Eles organizam as informações de maneira lógica, geralmente utilizando modelos como o relacional, que as distribui em tabelas. Além disso, esses sistemas oferecem funcionalidades importantes, como controle de acesso, backups, recuperação de dados e otimização de consultas. Isso tudo ajuda a manter o funcionamento do sistema eficiente, mesmo quando há grandes volumes de dados sendo processados.

# 4.CONCLUSÃO

Para a realização deste trabalho, foram adquiridos conhecimento sobre vários

aspectos de engenharia de software e aplicações, como a definição dos requisitos para termos clareza do que precisamos que o sistema faça, a forma de planejarmos o andamentos do desenvolvimento lidando com agendamento de implementações e correções, colaboratividade, controle e rastreamento de alterações (Git), estratégias de abordagem do código-fonte e identificação de pontos importantes a serem testados, tudo isso considerando que o sistema não apenas deve ser criado e existir, mas precisa ser mantido em pleno funcionamento durante seu ciclo de vida, e com uma boa documentação para que não apenas os usuários saibam do que ele é capaz, mas que os próximos mantenedores do código possam compreender o que o sistema faz, e ter êxito na compreensão para uma bem sucedida implementação.

Não apenas de estratégias de codificação e bom gerenciamento existem as aplicações, deve-se pensar na estrutura e infraestrutura de dados que serão manipulados com esta aplicação, afinal, nada mais importante em uma aplicação que os dados que são manipulados. O fluxo dos dados deve ser bem definido, e para isso contamos com ferramentas que nos auxiliam nessa modelagem, agregando clareza a quem precisa entender de onde o dado vem, como ficará disposto, e através de onde serão processados ou consultados. Para completar, estes dados devem ser tratados com muito cuidado, devendo ser agregadas estratégias de desempenho, pois o usuário precisa da informação para ontem, e de segurança, pois as informações não podem ser acessadas por qualquer um, e deve haver um ótimo sistema de backup, pra possíveis problemas sistêmicos.

A aplicação desenvolvida contou com várias técnicas de ambas as engenharias (sistemas e dados), não foi possível aplicar alguns aspectos pois o tempo para o projeto não era longo, e a quantidade de pessoas por vezes não justificaria – por exemplo – a adoção de metodologias ágeis. Foi tempo de aprender e aprofundar conhecimentos como o trabalho com banches no Git, uso de novas linguagens para desenvolvimento ou escritura de testes (Python, Cucumber, Markdown), criação de diagrama de dados com uma ferramenta que já provê o código para criação das tabelas.

O software foi entregue, ainda com um grande backlog de recursos a serem implementados, mas que com certeza irá auxiliar os funcionários do consultório a terem uma melhor organização em seu dia a dia de trabalho.

# Revisão Bibliográfica (ou título apropriado)

* **Graham S.** (2010). *Integrating Software and Database Systems*. New York: Springer.
* DRANKA, Marcos Andrei. *Sistema auxiliar para comparação estrutural de diferentes bancos de dados no processo de integração de banco de dados*. Florianópolis, SC, 2023. Trabalho de Conclusão de Curso (TCC)
* O artigo de David Moises B. Santos e outros com tema: Aplicando Project-based learning no estudo integrado de Engenharia de Software, Análise e Projeto de sistemas e banco de dados.

- SOMMERVILLE, Ian. *Software Engineering*. 9. ed. Boston: Addison-Wesley, 2011.

# 6. Referências

* <http://www.abenge.org.br/cobenge/legado/arquivos/12/artigos/441-Hugo%20Saba%20Pereira%20Cardoso.pdf>’
* DRANKA, Marcos Andrei. *Sistema auxiliar para comparação estrutural de diferentes bancos de dados no processo de integração de banco de dados*. Florianópolis, SC, 2023. Trabalho de Conclusão de Curso (TCC)
* **Graham, S.** (2010). *Integrating Software and Database Systems*. New York: Springer.