

GABRIEL ALMEIDA RIZZUTTO

**Criação de uma aplicação para automação e otimização de processos internos de uma
empresa em linguagem JAVA**

Guaratinguetá - SP

2022

Gabriel Almeida Rizzutto

Criação de uma aplicação para automação e otimização de processos internos de uma empresa em linguagem JAVA

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

Orientador (a): Prof. Dr. Daniel Julien B. da S. Sampaio

Guaratinguetá - SP

2022

R627c	<p>Rizzutto, Gabriel Almeida</p> <p>Criação de uma aplicação para automação e otimização de processos internos de uma empresa em linguagem JAVA / Gabriel Almeida Rizzutto – Guaratinguetá, 2022.</p> <p>33 f : il.</p> <p>Bibliografia: f. 33</p> <p>Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2022.</p> <p>Orientador: Prof. Dr. Daniel Julien B. da Silva Sampaio</p> <p>1. Software Desenvolvimento. 2. JavaServer pages (Programa de computador). 3. Banco de dados. I. Título.</p> <p>CDU 681.3.06</p>
-------	---

GABRIEL ALMEIDA RIZZUTTO

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE
DO REQUISITO PARA OBTENÇÃO DO DIPLOMA DE
"GRADUADO(A) EM ENGENHARIA ELÉTRICA"

APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Prof.(*) Dr.(*) DANIEL JULIEN B. DA S. SAMPAIO
Coordenador(a)

BANCA EXAMINADORA:


Prof.(*) Dr.(*) DANIEL JULIEN B. DA S. SAMPAIO
Orientador(a)/UNESP-FEG


Prof.(*) M.Sc. CARLOS A. MARCONDES DOS SANTOS
UNESP-FEG


Prof.(*) M.Sc. THIAGO JOSÉ MICHELIN
UNESP-FEG

DADOS CURRICULARES

GABRIEL ALMEIDA RIZZUTTO

NASCIMENTO 16.06.1995 – São Paulo / SP

FILIAÇÃO Reinaldo Rizzutto
Márcia de Almeida Rizzutto

2013/2022 Engenharia Elétrica - Bacharel
Universidade Estadual Paulista – Júlio de Mesquita Filho

Dedico este trabalho de modo especial, à minha
mãe Márcia e meu pai Reinaldo
por todo apoio recebido.

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, fonte da vida e da graça. Agradeço por minha vida, minha família e meus amigos;

ao meu orientador, Prof. Dr. Daniel Julien B. da S. Sampaio que jamais deixou de me incentivar. Sem a sua orientação, dedicação e auxílio, o estudo aqui apresentado seria praticamente impossível;

aos meus pais Reinaldo e Márcia, que apesar das dificuldades enfrentadas, sempre incentivaram meus estudos;

às funcionárias da Biblioteca do Campus de Guaratinguetá pela dedicação, presteza e principalmente pela vontade de ajudar;

aos funcionários da Faculdade de Engenharia do Campos de Guaratinguetá pela dedicação e alegria no atendimento.

“Uma ideia sem ação é como um arco
sem uma flecha.”

Martin Luther King

RESUMO

Este trabalho tem o objetivo de demonstrar a criação de software com a finalidade de automatizar tarefas manuais e otimizar processos internos de uma empresa. Para o desenvolvimento desse trabalho foi utilizado conceitos aprendidos ao longo do curso como a linguagem JAVA e a capacidade de resolução de problemas. A aplicação utiliza também um banco relacional para persistência de dados. O resultado obtido foi satisfatório, atingindo uma diminuição nos erros no processo e uma maior eficiência no atendimento de clientes por parte da empresa.

PALAVRAS-CHAVE: JAVA. Banco Relacional. Aplicação. Automatização. Otimização de Processos. Persistência de Dados.

ABSTRACT

This work aims to demonstrate the creation of a software with the purpose of automating manual tasks and optimizing internal processes of a company. For the development of this work, concepts learned throughout the course were used, such as the JAVA language and the ability to solve problems. The application also uses a relational database for data persistence. The result was satisfactory, achieving a reduction in errors in the process and greater efficiency in customer service by the company.

KEYWORDS: JAVA. Relational Database. Application. Automation. Process Optimization. Data Persistence.

LISTA DE FIGURAS

Figura 1 – Fluxograma da aplicação.....	21
Figura 2 – Página Inicial do programa desenvolvido	22
Figura 3 – Cabine de controle desenvolvida no programa	23
Figura 4 – Tela de abertura de chamados desenvolvido neste programa	25
Figura 5 – Tela de baixa.....	26
Figura 6 – Tela de relatórios	27
Figura 7 – Esquema relacional do banco de dados.....	28

LISTA DE TABELAS

Tabela 1 – Resumo das atividades realizadas	19
---	----

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	OBJETIVO.....	12
1.2	ESTRUTURA DO TRABALHO.....	13
2	ASPECTOS TEÓRICOS.....	14
2.1	SOBRE A LINGUAGEM JAVA.....	14
2.2	SOBRE O MYSQL.....	15
2.3	A EMPRESA ESCOLHIDA PARA O ESTUDO DE CASO.....	16
2.4	NECESSIDADE DO PROJETO.....	16
3	DESENVOLVIMENTO DA SOLUÇÃO.....	18
3.1	METODOLOGIA E ATIVIDADES.....	18
3.2	FLUXOGRAMA DA APLICAÇÃO.....	19
3.3	INTERFACES GRÁFICAS.....	20
3.3.1	Página principal.....	21
3.3.2	Página de Cabine de controle.....	22
3.3.3	Página de Abertura de chamados.....	24
3.3.4	Página das Baixas (ou finalizações).....	25
3.3.5	Relatórios.....	26
3.4	INFRAESTRUTURA.....	27
3.4.1	Clientes.....	28
3.4.2	Chamados.....	29
3.4.3	Baixas.....	29
3.4.4	Técnicos.....	30
3.4.5	Outras tabela.....	30
4	CONSIDERAÇÕES FINAIS.....	31
4.1	TESTES.....	31
4.2	RESULTADOS.....	31
4.3	PONTOS DE MELHORIA.....	31
4.4	CONCLUSÃO.....	31
	REFERÊNCIAS.....	33

1 INTRODUÇÃO

A popularização dos computadores devido ao surgimento e aprimoramento de novas tecnologias, trouxe inúmeros impactos na dinâmica da nossa sociedade, desde a maneira como nos comunicamos, entretemos e principalmente como trabalhamos. Segundo Adriano Pereira (2018), atualmente estamos vivendo em uma época conhecida como Quarta Revolução Industrial (ou Indústria 4.0) onde sistemas físicos se relacionam com o mundo digital, possibilitando uma maior eficiência, velocidade e escalabilidade para processos produtivos.

Por causa do avanço tecnológico, muitas empresas passaram a incorporar sistemas digitais em seu dia a dia com o objetivo de reduzir custos, acelerar tomadas de decisão e automatizar, controlar e otimizar processos internos. Graças a este movimento, surgiram novas posições de trabalho para os profissionais responsáveis por criar sistemas digitais e vendê-los no mercado como um produto.

Por isso tais empresas buscam como atualizar seus processos internos de produção, logística, atendimento, vendas etc., tendo como objetivo maior a redução de custos através da automatização e otimização de tarefas recorrentes aplicando tecnologias novas para que seus produtos continuem sendo atrativos e competitivos no mercado atual.

1.1 OBJETIVO

O objetivo deste trabalho é criar um sistema digital funcional para uma empresa do setor de telecomunicações, responsável pela instalação, manutenção, aluguel e revenda de produtos telefônicos para instituições públicas, visando um maior controle e qualidade das informações referentes aos serviços prestados, automação de tarefas manuais recorrentes e uma maior visibilidade das etapas de processos internos.

Com isso a empresa ganharia escalabilidade, podendo atender mais clientes sem precisar contratar muitos outros funcionários para desempenhar funções manuais. Além disso as informações históricas sobre atendimento serão guardadas de maneira mais estruturada, facilitando futuras consultas tanto para fins de auditoria quanto para fins de análise ou estudos de caso. Outro ganho significativo no emprego de sistemas como o descrito neste trabalho é o de velocidade nos processos internos, facilitando assim um atendimento pela empresa mais preciso e efetivo para a resolução do problema junto ao cliente.

Para a realização deste trabalho foram desenvolvidos softwares em linguagem JAVA em conjunto com a plataforma MySQL para a automatização de algumas tarefas manuais exercidas pelos usuários. Esta plataforma permite gravar algumas informações para seu funcionamento, o que torna o uso de um banco de dados essencial.

1.2 ESTRUTURA DO TRABALHO

No capítulo 1 está descrito a introdução do trabalho, assim como seus objetivos e a estrutura e organização dele.

No capítulo 2 estão apresentadas as descrições da escolha da linguagem utilizada e da plataforma, além de uma breve explicação sobre a empresa, necessidade do projeto e apresentação do problema que a aplicação desenvolvida vai resolver.

No capítulo 3 apresenta-se uma explicação sobre o desenvolvimento da solução, assim como explicações sobre as classes e a infraestrutura criadas para a aplicação.

Finalmente no capítulo 4 são apresentadas as considerações finais, resultados e conclusão.

2 ASPECTOS TEÓRICOS

A escolha de uma linguagem de programação específica vem da necessidade de encontrar dentro desta linguagem parâmetros importantes que consigam suprir as necessidades impostas pelo projeto. Por ter a necessidade de uma interface gráfica onde o usuário possa interagir de maneira intuitiva com o software, o autor escolheu a linguagem JAVA que possibilita o desenvolvedor criar e configurar tal interface de maneira relativamente simples, acelerando assim o processo de desenvolvimento. Outra necessidade do projeto era o armazenamento de dados gerados pela utilização do usuário com o software que futuramente seriam utilizados para análises, controles e relatórios. Por isso o autor escolheu a plataforma MySQL, responsável pelo armazenamento dos dados que é uma plataforma muito utilizada no mercado além de seu uso ser gratuito para este fim.

2.1 SOBRE A LINGUAGEM JAVA

Um programa, aplicação ou software pode ser classificado como um conjunto de instruções que devem ser executados por um computador. Podemos criar esses conjuntos de instruções através de linguagens de programação específica dada as necessidades e características do trabalho a ser realizado. Existem dois tipos de linguagem, a de baixo-nível e a de alto-nível. A primeira consiste em comandos para a máquina executar de maneira explícita, ou seja, definindo a nível de circuito o que deve acontecer sem nenhuma ou quase nenhuma abstração. A segunda permite que o desenvolvedor possa usar termos genéricos de programação que depois serão interpretados e traduzidos, para os comandos que a máquina deverá realizar (FARRELL, 2011).

A escolha sobre qual tipo de linguagem utilizar, baixo-nível ou alto-nível, depende da finalidade do projeto. As grandes vantagens das linguagens de baixo-nível é o tempo de execução ser menor, já para as de alto-nível são a facilidade do seu aprendizado e o fato de não ser necessário customizar os comandos exclusivamente para cada configuração de máquina, o que a torna escalável (FARRELL, 2011).

JAVA é uma linguagem de programação de alto-nível, baseada em classes e orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems, com ampla aplicação no mercado. Foi criada de

maneira que fosse simples o suficiente para sua popularização entre desenvolvedores do mundo todo (FLANAGAN, 2005). Ela possui uma sintaxe similar com a linguagem C, sua precursora, porém, tem o objetivo de ser uma linguagem voltada ao mercado e não a pesquisa (GOSLING, 2000). Além disso outra razão responsável pela sua popularização foi o fato de ela ser independente do sistema operacional que roda, precisando apenas da plataforma JAVA para ser executada. Isso possibilita o desenvolvimento de aplicações para diferentes tipos de dispositivos e computadores, tornando-a mais acessível (FLANAGAN, 2005). Outra razão pelo seu destaque no mundo inteiro, é de que por ser baseada em classes tem a possibilidade de ser organizada de maneira modular. Isso significa dar liberdade ao programador de reaproveitar códigos já utilizados em outros projetos, facilitando assim o desenvolvimento de projetos de maneira colaborativa, o que impulsionou a sua disseminação pela internet (FLANAGAN, 2005).

2.2 SOBRE O MYSQL

Segundo Terry Halpin (2010) um banco de dados é basicamente um conjunto de dados ou informações que se associam e sua aplicação mais usual é a de persistência de dados. Existem diferentes maneiras de se organizar estes conjuntos e o escolhido para este projeto foi o modelo relacional que é amplamente usado no mercado. O modelo relacional é um banco baseado em um modelo de dados relacionais onde a informação é guardada em tabelas, sendo as colunas os campos nomeados e as linhas os dados referente ao objeto em questão. Assim, todos os objetos (linhas) pertencentes a uma tabela possuem as mesmas características (campos nomeados).

Tais bancos de dados historicamente eram muito caros por exigirem uma quantidade substancial de hardware para rodarem com uma boa “performance” e por terem custos altos de venda devido ao software e suporte envolvido (DUBOIS, 2013). Devido o desenvolvimento de softwares comunitários e gratuitos (*open source*) este cenário mudou, o que possibilitou a difusão de bancos de dados relacionais entre as aplicações desenvolvidas (DUBOIS, 2013).

Um dos principais softwares gratuitos responsáveis por tal mudança é o MySQL.

O MySQL é um sistema gerenciador de banco de dados de código aberto usado principalmente para gerir dados relacionais automatizados de algumas tarefas manuais que são exercidas pelos usuários.

Ele é um sistema de gestão de bancos de dados relacional SQL (*Structured Query Language*) (DUBOIS, 2013). O SQL é uma das linguagens de *query* mais utilizadas nas interações com bancos de dados relacionais (HALPIN, 2010), por isso é amplamente divulgada dentro das comunidades de desenvolvedores, o que facilita o seu aprendizado.

O MySQL é um software completo que permite configurar um servidor com passos simples e de maneira rápida, o que facilitou a implementação de um servidor para o projeto.

2.3 A EMPRESA ESCOLHIDA PARA O ESTUDO DE CASO

A empresa fundada em 1985 fornece serviços como locação, instalação e manutenção de sistemas telefônicos principalmente para órgãos públicos. Estes serviços são prestados para produtos como sistemas de PABX e acessórios, de gravação digitais e correio de voz, de tarifação, aparelhos telefônicos, entre outros. É uma empresa já consolidada no mercado e que já teve e tem como parceira outras grandes empresas multinacionais.

A empresa incorpora novos clientes através de licitações feitas pelos órgãos públicos, com as necessidades do projeto definido através de editais. Assim que terminado o processo de licitação, um contrato é assinado especificando quais tipos de serviços a empresa deverá prestar e os seus respectivos prazos.

Por serem editais próprios e necessidades específicas, cada cliente possui um contrato diferenciado com a empresa, tendo assim suas próprias definições de prazos para atendimento, características de produto, instalações específicas, material utilizado, mão de obra aplicada etc. Por isso o programa a ser desenvolvido para atender as diferentes demandas deve ser flexível para suportar todos os possíveis diferentes casos de clientes.

2.4 NECESSIDADE DO PROJETO

Esta empresa particularmente possui todo o seu atendimento para clientes externos é feito através de chamados internos. Estes podem ser feito pelo próprio cliente que entra em contato com uma área específica da empresa, *Help Desk*, para solicitar algum tipo de serviço, ou para a prestação de algum serviço definido em contrato.

Nos dois casos o *Help Desk*, após entender e definir qual a necessidade do cliente deve passar à equipe técnica quais serão os serviços prestados. Todo o processo de definição das necessidades solicitadas era manual e incluía analisar o contrato vigente para verificar se o chamado estava contemplado além de registrar e arquivar o que seria realizado no chamado. Esta dinâmica demandava muito tempo e estava sujeita a falhas de atendimento.

Além disso, não existia um controle oficial de qual funcionário da equipe técnica seria o responsável pela assistência técnica solicitada, ou em qual etapa do atendimento o processo estava. Isso resultava em erros de prestação de serviços não contemplados em contrato ou atrasos na resolução do chamado, por exemplo.

A fim de mitigar tais erros, optou-se pela utilização de um sistema interativo e automático desenvolvido por linguagem de programação e sua aplicação foi desenvolvida contemplando os casos especiais de atendimento e incluindo todas possíveis funcionalidades para seu funcionamento. O software de programação desenvolvido vem a atender as diferentes demandas e objetivos:

- cadastrar os clientes externos para definir qual tipo de serviço pode ser realizado,
- registrar os chamados criados pelo time de *Help Desk*,
- facilitar o acompanhamento de tais chamados de maneira visual, prática e fácil,
- controlar os estados de cada chamado existente,
- organizar a fila de atendimento melhorando assim a utilização dos recursos da empresa,
- monitorar a equipe técnica que está realizando o atendimento ou disponível para este fim.

3 DESENVOLVIMENTO DA SOLUÇÃO

O programa desenvolvido em JAVA foi construído utilizando um ambiente de desenvolvimento integrado (*Integrated Development Environment, IDE*) *open source* chamado NetBeans™. A escolha do ambiente IDE se deve a familiarização com a ferramenta utilizada e por possuir algumas características essenciais que facilitam o desenvolvimento da aplicação. Dentre estas facilidades pode-se citar:

- o editor de código-fonte, capaz de dar sugestões de métodos, autocompletar palavras e trechos de código e indentação automática do código,
- editor de interface gráfica capaz de posicionar elementos gráficos de maneira visual,
- depurador/compilador para testar e executar a aplicação além de outras ferramentas.

A aplicação desenvolvida pode ser dividida em 3 partes para facilitar o entendimento. A primeira corresponde as interfaces gráficas, onde o usuário consegue interagir com a aplicação. A segunda corresponde as classes e as regras de negócio que a aplicação possui, e suas funcionalidades. Já a terceira corresponde a infraestrutura de dados que a aplicação consome para que haja persistência de dados.

3.1 METODOLOGIA E ATIVIDADES

O mapeamento do processo de criação e atendimento de cada chamado consiste em identificar como o fluxo era organizado, todas as etapas e interações que cada funcionário da empresa tinha como responsabilidade e especificar prazos necessários. de maneira visual, depurador/compilador para testar e executar a aplicação além de outras ferramentas.

Para realização deste mapeamento foi construído abaixo um diagrama de atividades macro para facilitar a organização do processo. Na tabela 1 estão descritas as atividades a serem realizadas:

Tabela 1 - Resumo das atividades realizadas

Atividades	Descrição
A1	Mapeamento do processo de criação e atendimento do chamado
A2	Desenho das interfaces gráficas
A3	Implementação das funcionalidades e operações no programa
A4	Desenho e implementação da infraestrutura
A5	Ajustes finais, testes e correções de erros
A6	Treinamento de utilização e acompanhamento de uso

Fonte: Produção do próprio autor.

A atividade A1, mapeamento do processo de criação e atendimento do chamado, consiste em identificar como o fluxo era organizado, todas as etapas e interações que cada funcionário da empresa tinha como responsabilidade e especificar prazos necessários. de maneira visual, depurador/compilador para testar e executar a aplicação além de outras ferramentas.

As atividades A2, A3 e A4 dependem da etapa de mapeamento e consiste na parte de desenvolvimento do código para o programa e implementação de todas as funcionalidades exigidas, e é baseada nos insumos trazidos pela etapa A1 de mapeamento.

A atividade A5 utiliza um protótipo inicial que será refinado até atingir um grau de usabilidade exigido pela empresa. Nesta etapa o processo roda paralelamente de maneira manual e no programa a fim de encontrar possíveis erros, quando comparados os resultados, e resolvê-los.

A atividade A6 foi um treinamento de usabilidade com os usuários, e pequenos ajustes tanto visuais nas interfaces gráficas quanto funcionamento de funcionalidades. Foi passado também instruções para possíveis manutenções no sistema.

3.2 FLUXOGRAMA DA APLICAÇÃO

A figura 1 a seguir mostra um fluxograma simplificado da aplicação mostrando as principais etapas do atendimento.

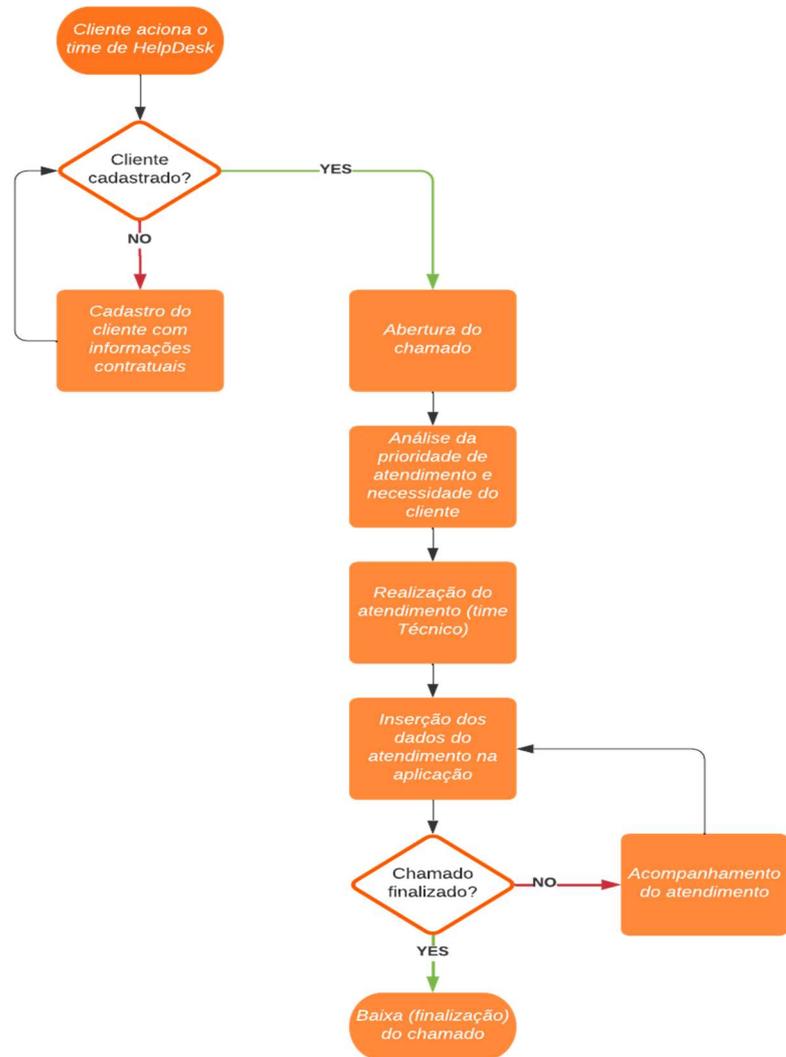
O processo inicia com o contato do cliente com a empresa prestadora de serviço informando algum mau funcionamento de algum equipamento utilizado pelo cliente ou para a instalação de algum outro. Com isso é gerado um chamado de atendimento. Para que o operador possa inserir as informações do chamado, o cliente deve estar cadastrado na aplicação, caso não esteja, deve ser feito o cadastro antes.

Depois da abertura do chamado dentro do sistema, o atendimento é realizado, levando em conta a prioridade e tempo máximo para o atendimento. Com isso, é adicionado os dados do atendimento no chamado dentro do sistema e realizado a baixa (finalização) do chamado.

3.3 INTERFACES GRÁFICAS

Todas as interfaces foram construídas utilizando o editor de interface gráfica da IDE, e foram utilizados botões, menus, containers, *labels* entre outros elementos gráficos. Os próximos tópicos trarão exemplos de parte da aplicação com a finalidade de demonstrar seu layout e o seu funcionamento.

Figura 1 - Fluxograma da aplicação



Fonte: Produção do próprio autor.

3.3.1 Página principal

A página principal, figura 2, exibe os botões que direcionam para a página de *login* no sistema e realizam o *logout* do usuário na aplicação. Além disso, possui também “*labels*” (*identificadores*) que indicam qual usuário está conectado naquela execução da aplicação, um relógio com a data e hora local e se a aplicação conseguiu se conectar ao banco de dados.

Figura 2 - Página Inicial do programa

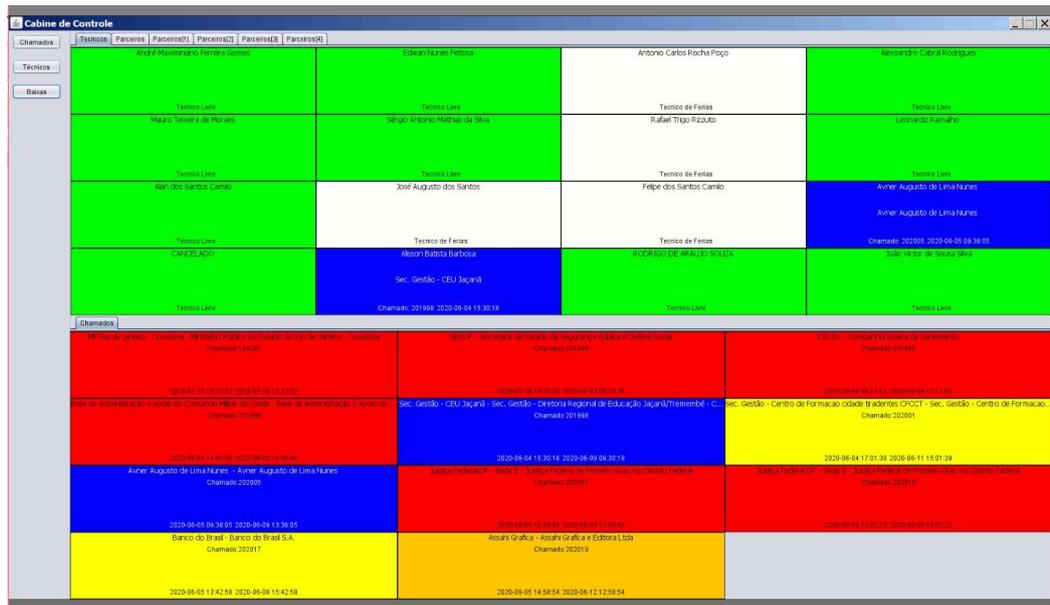


Fonte: Produção do próprio autor.

3.3.2 Página de Cabine de controle

A página de cabine de controle pode ser considerada o elemento de maior funcionalidade do programa desenvolvido. Nela são apresentados os elementos essenciais do sistema exibindo os chamados a serem atendidos ou que estão em processo de atendimento (figura 3). Ela possui três botões, um para abrir novos chamados (Chamados) outro para adicionar um novo técnico ou parceiro (Técnicos) e o último para adicionar uma finalização (baixa) ou realizar uma pesquisa de chamados ou baixas já existentes.

Figura 3 - Página cabine de controle desenvolvida no programa



Fonte: Produção do próprio autor.

Na parte superior desta página pode-se encontrar os técnicos já cadastrados no sistema. Ela é composta de um objeto da classe *JTabbedPane* podendo conter no máximo 16 técnicos ou parceiros distribuídos ao longo das abas do painel. Cada técnico ou parceiro mostrado na tela é um objeto da classe *ModeloTecnico*, criada para servir de modelo para cada elemento inserido no painel, e é uma classe que estende a classe *JPanel*. O objeto modelo é composto por três *labels* contendo o nome do técnico ou parceiro, o nome do cliente ao qual ele está alocado e o seu atual status. Os possíveis status são:

- “Técnico Livre”; caso não esteja realizando atendimento.
- “Técnico Ausente”; caso esteja ausente.
- “Técnico de férias”; caso esteja de férias.
- “Chamado:”; seguido o número do chamado e horário de início da etapa mais recente de atendimento.

A parte inferior da página é análoga a parte superior, porém possui os dados dos chamados abertos. Contendo o nome do cliente, número do chamado horário de abertura e horário de finalização. Outra diferença é que o chamado mostrado é um objeto da classe *ModeloChamado*, que também estende a classe *JPanel*. Um elemento importante neste desenvolvimento e a identificação dos chamados organizados pela sua data de abertura.

Utilizou-se uma codificação de cores nos dois modelos desenvolvidos para facilitar a identificação de seu status de maneira mais visual:

- Vermelho; caso o prazo para o atendimento tenha sido extrapolado, ou caso o chamado já tenha sido atendido, mas não finalizado.
- Laranja; caso esteja dentro do prazo para atendimento e o técnico ou parceiro que realizará o atendimento está em deslocamento ao cliente.
- Amarelo; caso esteja dentro do prazo, mas ainda não houve nenhuma movimentação interna para o atendimento.
- Azul; caso esteja dentro do prazo para atendimento e o técnico ou parceiro está realizando o atendimento

3.3.3 Página de Abertura de chamados

Na parte superior temos uma caixa de texto que permite a edição do número do chamado que foi gerado pelo sistema, caso necessário. Os números do chamado são no formato ano, seguido de quatro dígitos, por exemplo: 201001. Caso o usuário não queira definir um número específico para o chamado, o sistema sugere o próximo número sequencial automaticamente.

As outras caixas de texto ou de seleção permitem a inserção de informações como: fantasias de clientes cadastrados razões sociais, e os dados do cliente cadastrado, utilizado para checagem. Uma caixa de seleção de prioridade, permite o usuário escolher a informação de quantas horas o chamado deve ser atendido, elemento importante para controle do atendimento e de sua eficiência. As opções de escolha que aparecem para o usuário são referentes ao cliente escolhido. O prazo de atendimento é definido contratualmente com o cliente e inserida no sistema ao cadastrar um novo cliente. Temos também o tipo do chamado pode ser marcado nas caixas de combinação com letras P (preventivo), C (corretivo), I (instalação), V (vistoria) e R (remoto), sendo as marcações em opções não excludentes. Também nesta página de abertura de chamado tem-se e uma caixa de texto para guardar um descritivo do defeito apresentado. Finalmente na parte inferior existem dois botões para salvar (guardar as informações inseridas) ou cancelar (descartar as informações).

Figura 4 - Tela de abertura de chamados desenvolvido neste programa

The image shows a software window titled "Abertura de Chamados". It contains the following elements:

- Nº do Chamado:** A text input field with a search icon on the right.
- Fantasia:** A dropdown menu.
- Razão Social:** A dropdown menu.
- Razão Social:** A text input field.
- Contato:** A text input field.
- Endereço:** A text input field.
- Nº, Bairro, Cidade, Estado, CNPJ:** Five separate text input fields.
- Prioridade:** A dropdown menu followed by "--> 0 horas para atendimento."
- Tipo do Chamado:** Five checkboxes labeled P, C, I, V, and R.
- Defeito Apresentado:** A large text area for describing the issue.
- Numero de Caracteres: 0:** A character count indicator at the bottom right of the text area.
- Buttons:** "Salvar" (Save) and "Cancelar" (Cancel) buttons at the bottom.

Fonte: Produção do próprio autor.

3.3.4 Página das Baixas (ou finalizações)

A página de baixas desenvolvida pode ser acessada através da cabine de controle, clicando no botão "Baixas" ou clicando em algum objeto do tipo ModeloChamado. Nela é possível consultar chamados abertos ou finalizados, possui os dados do cliente referente ao chamado, dados sobre o próprio chamado. Nesta página que é feita a inserção das informações sobre o atendimento. Grande parte da gestão do fluxo de atendimento dos chamados é realizada através desta tela.

Figura 5 - Tela de baixa

The screenshot shows a web-based form titled "Baixa" (Closing) with the following fields and controls:

- Nº de Serviço:** A text input field with a search icon.
- Nome do Cliente:** A large text input field.
- Fantasia:** A text input field.
- Contato:** A text input field.
- Email / Numero de Contato:** A text input field.
- Endereco:** A text input field.
- Address Fields:** A row of six input fields labeled "Nº", "Bairro", "Cidade", "Estado", "CEP", and "CNPJ".
- Contract Type:** Radio buttons for "Garant.", "Loc.", and "Man.".
- Emergency Service Time:** A spinner field labeled "Prazo de Atendimento Emergencial" with a value of "0".
- Normal Service Time:** A spinner field labeled "Prazo de Atendimento Normal" with a value of "0".
- Service Type:** Checkboxes for "Remoto" and "Local".
- Timeline Buttons:** A row of four buttons: "Data do Chamado", "Partida do Técnico", "Entrada do Técnico", and "Saída do Atendimento".
- Defects and Summary:** Two buttons: "Defeitos Apresentados" and "Resumo do Contrato".
- Technician Selection:** Three dropdown menus, each labeled "Técnico".
- Observations:** A large text area labeled "Observações".
- Bottom Buttons:** Four buttons: "Salvar", "Finalizar Chamado", "Cancelar Chamado", and "Cancelar".

Fonte: Produção do próprio autor.

3.3.5 Relatórios

Dada a necessidade de gerar relatórios para gerenciamento das informações existentes no banco de dados fez necessário dentro da página de baixas um mecanismo que permita a confecção de relatórios. A visualização ou extração dos dados é feita através da tela de relatórios podendo ser acessada pelo menu da página principal. Existem dois tipos de relatórios que podem ser gerados, um para clientes e outro para técnicos. Assim o operador do sistema pode fazer o acompanhamento de todos os chamados abertos em um determinado período, referente a um cliente ou todos, assim como em relação aos chamados atendidos por um técnico, ou todos. Tais relatórios podem ser extraídos selecionando um dos dois formatos, “.xslm” ou “.pdf”, e clicando no botão “Salvar...”.

Figura 6 - Tela de relatórios

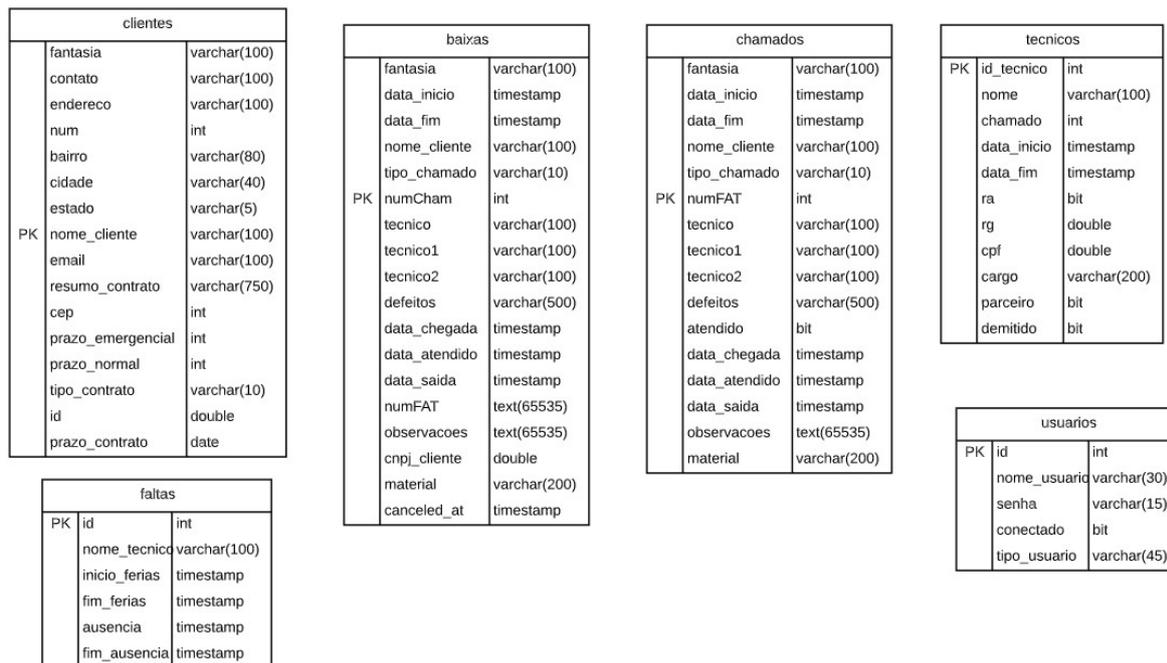
Title 1	Title 2	Title 3	Title 4

Fonte: Produção do próprio autor.

3.4 INFRAESTRUTURA

A aplicação possui o executável, desenvolvido em JAVA, que engloba as funcionalidades e interface do sistema e o banco de dados, responsável pela persistência de dados. O banco de dados corresponde a um servidor local baseado em linguagem de consulta estruturada (Structured Query Language, SQL) utilizando o gerenciador MySQL, desenvolvida pela empresa Oracle. O MySQL foi escolhido por ser gratuito e com alta popularidade em diversos outros sistemas. As tabelas foram criadas com o intuito de armazenar as informações inseridas através da interação do usuário com o programa a fim de evitar com que o usuário tivesse que reinsserir dados repetidos a cada interação. As tabelas relacionais criadas juntos com seus respectivos campos podem ser vistas no esquema do banco de dados na figura 6.

Figura 7 - Esquema relacional do banco de dados



Fonte: Produção do próprio autor.

Cada tabela criada tem como objetivo armazenar dados tanto para uma melhor fluidez no uso do programa quanto como registro histórico de informações sobre os chamados.

3.4.1 Clientes

A tabela Clientes na figura 7 tem a função de guardar dados cadastrais relacionados a cada cliente com contrato com a empresa. Para cada linha da tabela se encontra um cliente específico e suas informações como endereço, nome da empresa, prazos de atendimento (definidos no contrato) e a data em que o contrato se encerra.

A data em que se encerra o contrato é utilizada para impedir que chamados sejam abertos para clientes que estejam com o contrato vencido e como lembrete para os operadores que o contrato está próximo do encerramento para que possam tomar as devidas providências junto ao time comercial. Tal lembrete aparece ao se iniciar o programa 3 meses antes do vencimento do contrato.

3.4.2 Chamados

A tabela Chamados na figura 7 possui todos os chamados que estão com o status aberto, ou seja, todos que ainda devem ser atendidos ou estão no processo de atendimento. Toda vez que um chamado é criado pela tela de abertura de chamados, uma nova linha é inserida nessa tabela, analogamente, toda vez que um chamado é finalizado ou cancelado a linha correspondente a este chamado é apagada dessa tabela. Assim podemos considerar esta tabela como uma tabela com dados temporários onde são criadas linhas e excluídas logo em seguida.

Quando criados, os chamados possuem apenas alguns campos básicos preenchidos, como por exemplo, data de abertura do chamado, prazo de atendimento, nome do cliente, e tipo do chamado. Enquanto o atendimento for acontecendo o operador vai inserir no programa dados como quais são os técnicos ou parceiros que realizarão o atendimento (limite 3), horário de saída do técnico para o atendimento, horário de chegada do técnico no cliente e o horário que termina o atendimento. Assim que o último horário for inserido e o botão de “Salvar” dentro da tela de “Baixas” for clicado, o chamado vai ser considerado como atendido, e permanecerá na tabela até que o número do FAT (identificador utilizado para a empresa prestadora de serviço) for inserido, terminando assim todas as pendências em relação ao atendimento.

Quando finalizado o chamado, é feita uma “baixa” nele, deletando a sua linha na tabela de chamados e inserindo uma nova linha contendo as informações do atendimento na tabela de baixas.

3.4.3 Baixas

A tabela Baixas possui todos os chamados em que o atendimento já foi finalizado, ou o chamado foi cancelado. Esta tabela tem o objetivo de guardar o histórico dos atendimentos para mais tarde ser usado em relatórios ou possíveis análises sobre tais atendimentos. Ela possui todos os campos presentes na tabela Chamados e mais alguns como “canceled_at”, que guarda a data de cancelamento do chamado (aqueles que não são cancelados, o campo está como vazio).

Diversas telas consultam essa tabela como por exemplo a tela de baixas. Essas consultas são feitas através da tela de baixas inserindo o número do chamado (identificador do chamado no sistema), ou através da tela de relatórios, que mostram os chamados em aberto e finalizados referentes a um cliente ou um técnico que realizou o atendimento.

3.4.4 Técnicos

A tabela Técnicos na figura 7 tem o objetivo de guardar dados relacionados aos técnicos ou parceiros responsáveis por realizarem os atendimentos dos chamados. Assim todos os campos como nome, RG, CPF, cargo, são referentes aos técnicos cadastrados. Quando um técnico é desligado da empresa, o campo desligado (binário) é colocado como verdadeiro e sua linha não é apagada da tabela para que ainda seja possível extrair relatórios referentes aos atendimentos que ele realizou.

Apenas os técnicos e parceiros registrados nessa tabela e ativos aparecem como opção para realizar um novo atendimento na tela de baixas. Assim, para designar qualquer técnico para um chamado, deve-se primeiro cadastrar o mesmo através da tela de técnicos.

3.4.5 Outras tabelas

Existem outras tabelas utilizadas como suporte para uma boa execução do sistema, como por exemplo, a tabela de Usuários que guarda os dados dos operadores do sistema. Apenas depois do operador se autenticar no sistema ele consegue fazer alterações em chamados ou extrair relatórios. Existem três tipos de usuários no sistema.

- “ADM”; administradores do sistema com acesso a todos os recursos, como inclusão de novos usuários.
- “TEC”; técnicos com a visibilidade apenas da cabine de controle para acompanhar os chamados que precisam de atendimento.
- “HELP”; operadores do sistema capazes de cadastrar novos clientes, chamados e geri-los.

Outra tabela na figura 7 é a de Faltas que tem como o objetivo guardar as ausências ou férias dos técnicos, para que o sistema possa identificar quais técnicos estão disponíveis para um atendimento e quais não.

CONSIDERAÇÕES FINAIS

4.1 TESTES

A aplicação foi testada durante duas semanas com um atendente e alguns clientes específicos. Esta etapa foi utilizada para encontrar ajustes finos tanto na parte gráfica da aplicação como posicionamento de botões como mudança de alguns textos de descrição de campos, quanto na ajustes em lógica e casos de exceção, como adicionar a possibilidade de mais de um técnico para o atendimento. Todos esses pontos de melhoria só foram percebidos graças aos testes e uso da aplicação, por isso a importância dessa etapa.

4.2 RESULTADOS

Com a implementação dessa nova ferramenta a empresa conseguiu ter um melhor controle sobre os atendimentos que ocorriam evitando assim possíveis desperdícios de materiais em estoque, e um maior controle sobre a atuação do time técnico nos atendimentos aos clientes. Outros resultados positivos importantes foram a capacidade de se gerar relatórios sobre os atendimentos de maneira automática e mais rápida e gestão destes atendimentos que evitam atrasos ou a falta de eficiência nos chamados.

4.3 PONTOS DE MELHORIA

A aplicação criada possui algumas funcionalidades que poderiam ser implementadas para garantir uma maior segurança e qualidade dos dados. Uma dessas funcionalidade seria a criação de redundância de dados, através de cópias recorrentes do banco de dados para evitar possíveis perdas de dados por mal uso.

Outra funcionalidade que poderia ser incluída é o rastreamento de todas as interações que os usuários têm com o programa (*log*). Assim seria possível encontrar os erros de uso incorreto do programa para mitigá-los.

Um terceiro ponto de melhoria seria o treinamento constante sobre a utilização da aplicação, tanto para novos usuários quanto para os antigos.

4.4 CONCLUSÃO

A aplicação conseguiu resolver diversos problemas recorrentes do processo manual que existia anteriormente a aplicação. Isso possibilitou a empresa uma melhor alocação de recursos

para solução dos chamados, otimizando os atendimentos e evitando perdas monetárias significativas por quebras de contrato nos atrasos dos atendimentos.

Este projeto se mostrou aplicável e escalável, podendo ser adotado em outras empresas com o mesmo fim. Porém a implementação desse projeto exige certo conhecimento sobre a área.

Uma das dificuldades encontradas foi a implementação das funcionalidades e organização do processo. O processo de atendimento dos chamados, por ser totalmente manual, era desorganizado o que dificultou o seu mapeamento e acarretou num maior tempo para definir o que o programa deveria contemplar e executar.

Outra dificuldade foi a adaptação para a utilização do programa, por isso o autor decidiu incluir no projeto um tempo de treinamento e acompanhamento, como pode ser visto no cronograma acima.

REFERÊNCIAS

DUBOIS, Paul. **MySQL**. 5. ed. Londres: Pearson Education, 2013.

FARRELL, Joyce. **Java programming**. 6. ed. Boston: Cengage Learning, 2011.

FLANAGAN, David. **Java in a Nutshell**. 5. ed. Sebastopol: O'Reilly Media, 2005.

GOSLING, J. *et al.* **The Java language specification**. 2. ed. Boston: Addison Wesley, 2000.

HALPIN, Terry; MORGAN, Tony. **Information modeling and relational databases**. 2. ed. Amsterdam: Elsevier Science, 2010.

PEREIRA, Adriano; SIMONETTO, E. D. O. Indústria 4.0: conceitos e perspectivas para o Brasil. **Revista da Universidade Vale do Rio Verde**, v. 16, n. 1, p. 1-9, jul.2018.