



OTÁVIO HENRIQUE MELO DE SOUZA

**DESENVOLVIMENTO DE APLICAÇÕES PARA AUTOMAÇÃO
COMERCIAL NA EMPRESA CRIARE TECNOLOGIA**

LAVRAS – MG

2023

OTÁVIO HENRIQUE MELO DE SOUZA

**DESENVOLVIMENTO DE APLICAÇÕES PARA AUTOMAÇÃO COMERCIAL NA
EMPRESA CRIARE TECNOLOGIA**

Trabalho de Conclusão de Curso apresentado à
Universidade Federal de Lavras, como parte das
exigências do Curso de Engenharia de Controle
e Automação, para a obtenção do título de
Bacharel.

Prof. Dr. Bruno de Abreu Silva

Orientador

LAVRAS – MG

2023

OTÁVIO HENRIQUE MELO DE SOUZA

**DESENVOLVIMENTO DE APLICAÇÕES PARA AUTOMAÇÃO COMERCIAL NA
EMPRESA CRIARE TECNOLOGIA**

Trabalho de Conclusão de Curso apresentado à
Universidade Federal de Lavras, como parte das
exigências do Curso de Engenharia de Controle
e Automação, para a obtenção do título de
Bacharel.

APROVADA em 16 de março de 2023.

Prof. Dr. Bruno de Abreu Silva	DCC / UFLA
Prof. Dr. Maurício Ronny de Almeida Souza	DCC / UFLA
Prof. Dr. Vinícius Miranda Pacheco	DAT / UFLA

Prof. Dr. Bruno de Abreu Silva
Orientador

**LAVRAS – MG
2023**

Dedico à toda minha família, principalmente minha mãe Viviana Lopes de Melo Souza e meu pai Fábio Lúcio de Souza. Dedico também a todos os meus amigos.

AGRADECIMENTOS

Em primeiro lugar, agradeço imensamente aos meus pais e minha irmã, por serem o pilar em minha vida e por sempre me apoiarem em meus sonhos.

Agradeço também a todos os meus amigos, que sempre estiveram ao meu lado compartilhando bons e maus momentos, e me ajudando a alcançar meus objetivos.

Sou grato ao Prof. Dr. Bruno de Abreu Silva por seus ensinamentos durante o curso e por aceitar me orientar e ajudar no processo de produção deste trabalho.

Não poderia deixar de agradecer à Universidade Federal de Lavras e a todos os seus funcionários, os técnicos-administrativos e os professores que tive a honra de ter como docentes, pois cada um deles contribuiu, de sua maneira única, para o meu desenvolvimento pessoal e profissional.

*“O sucesso é a soma de pequenos esforços repetidos dia após dia.”
(Robert Collier)*

RESUMO

Com o aumento da competição no mercado, a utilização de sistemas automatizados para gerenciamento de negócios tem se tornado cada vez mais importante, com isso o presente trabalho apresenta os principais projetos implementados durante o estágio realizado entre os dias 07/10/2020 e 30/06/2021 na empresa Criare Tecnologia, empresa especializada em desenvolvimento de softwares para automação comercial. O objetivo do trabalho é mostrar o desenvolvimento de aplicações focadas em automação comercial, discutir os desafios encontrados e demonstrar os principais resultados obtidos. Foram realizadas diversas atividades de desenvolvimento de software com a IDE Delphi, linguagem Object Pascal e banco de dados MySQL, tais como o desenvolvimento de dois projetos individuais "Exportador Viva Broker" e "Importador Padaria Lisboa", e também no desenvolvimento do SIA, um sistema ERP e fiscal completo, integrando uma equipe de seis colaboradores. Tecnologias como React JS, TypeScript e Node JS também foram utilizadas para desenvolver uma solução Web completa formando uma dupla de desenvolvedores. Independente do projeto, todos os times juntamente com o setor de testes, seguem os ritos do Scrum, com reuniões diárias logo ao início do dia. Os principais resultados obtidos durante o estágio foram os desenvolvimentos bem-sucedidos de sistemas de automação comercial que atenderam às necessidades dos clientes dentro dos prazos estabelecidos pela empresa. Em conclusão, os projetos desenvolvidos contribuíram positivamente para os clientes da Criare, permitindo que eles automatizassem suas operações comerciais de maneira mais eficiente e produtiva.

Palavras-chave: Automação Comercial. Delphi. Framework Scrum. React JS. TypeScript. Node JS. MySQL.

ABSTRACT

With the increase in competition in the market, the use of automated systems for business management has become increasingly important. Therefore, this paper presents the main projects implemented during the internship carried out between October 7, 2020, and June 30, 2021, at Criare Tecnologia, a company specialized in software development for commercial automation. The objective of the work is to show the development of applications focused on commercial automation, discuss the challenges encountered, and demonstrate the main results obtained. Several software development activities were carried out using the Delphi IDE, Object Pascal language, and MySQL database, such as the development of two individual projects "Viva Broker Exporter" and "Lisboa Bakery Importer", as well as the development of SIA, a complete ERP and fiscal system, integrating a team of six collaborators. Technologies such as React JS, TypeScript, and Node JS were also used to develop a complete web solution, forming a pair of developers. Regardless of the project, all teams, together with the testing department, follow the Scrum rituals, with daily meetings at the beginning of the day. The main results obtained during the internship were the successful development of commercial automation systems that met the clients' needs within the deadlines established by the company. In conclusion, the projects developed positively contributed to Criare's clients, allowing them to automate their commercial operations more efficiently and productively.

Keywords: Comercial Automation. Delphi. Scrum. React JS. TypeScript. Node JS. MySQL.

LISTA DE FIGURAS

Figura 2.1 – Exemplo de POO em Código Java	13
Figura 2.2 – Tabela de Produtos em SQL	16
Figura 2.3 – Exemplo de Código em React	21
Figura 2.4 – Exemplo de Código em TypeScript	22
Figura 2.5 – Demonstração do Scrum	24
Figura 2.6 – Demonstração de uma Dashboard no Jira	25
Figura 2.7 – Exemplo de Relatório em Gráfico no Jira	25
Figura 2.8 – Representação do Quadro Kanban	26
Figura 3.1 – Exemplo de Layout (Sistema 1)	29
Figura 3.2 – Exemplo de Arquivo Exportado	30
Figura 3.3 – Exemplo de Layout (Sistema 2)	30
Figura 3.4 – Interface do Exportador	31
Figura 3.5 – Método "OnNotificaInterface"	32
Figura 3.6 – Demonstração da Thread em Delphi	32
Figura 3.7 – Menu de Seleção	33
Figura 3.8 – SIA - Página Inicial	34
Figura 3.9 – SIA - Página Inicial	35
Figura 3.10 – Opção PixCriare na Forma de Pagamento	36
Figura 3.11 – Tela de Pagamento PixCriare	37
Figura 3.12 – <i>Ticket</i> de Impressão PixCriare	38
Figura 3.13 – Pagamento PixCriare Recebido	39
Figura 3.14 – Detalhes do Caso no Mantis BT	39
Figura 3.15 – <i>Select</i> do Relatório de Boletos Emitidos	40
Figura 3.16 – <i>Select</i> do Relatório de Boletos Emitidos	41
Figura 3.17 – Componente FastReport para Criação de Relatórios	41
Figura 3.18 – Relatório de Boletos Emitidos no Sistema	42
Figura 3.19 – Exemplo Layout Importação de Clientes	43
Figura 3.20 – Interface Principal do Importador	43
Figura 3.21 – Segunda Aba do Importador	44
Figura 3.22 – Requisições do React para a API	47
Figura 3.23 – Dashboard	48

Figura 3.24 – Telas de Listagem	48
Figura 3.25 – Formulários	49
Figura 3.26 – Exemplo de Filtros dos Relatórios do SIA Web	49
Figura 3.27 – Exemplo de Relatório Gerado	50
Figura 3.28 – Controle de Usuários (<i>Admin</i>)	50
Figura 3.29 – Aplicação em Celulares	51
Figura 3.30 – Mensagens em Diálogo	51
Figura 3.31 – Mensagens de Alerta	52

LISTA DE QUADROS

Quadro 2.1 – Exemplo de tabela em um banco de dados	15
---	----

SUMÁRIO

1	INTRODUÇÃO	11
2	REFERENCIAL TEÓRICO	12
2.1	Programação Orientada a Objetos	12
2.1.1	Delphi	14
2.2	Banco de Dados e SQL	14
2.2.1	SQL	15
2.2.2	MySQL	16
2.3	Padrão Arquitetural de Software	17
2.3.1	MVC	17
2.4	Desenvolvimento Web e Mobile	18
2.4.1	Node JS	18
2.4.2	React JS	19
2.4.3	TypeScript	20
2.5	Metodologias Ágeis	21
2.5.1	Framework Scrum	22
2.6	Jira Software	24
3	METODOLOGIA DE DESENVOLVIMENTO E RESULTADOS	27
3.1	Metodologia de Desenvolvimento	27
3.2	Exportador Viva Broker	28
3.3	SIA	33
3.3.1	Recebimento de Contas com PixCriare	34
3.3.2	Relatório de Boletos Emitidos	37
3.4	Importador Padaria Lisboa	42
3.5	SIA Web	44
4	CONCLUSÃO	53
	REFERÊNCIAS	55

1 INTRODUÇÃO

A automação comercial é uma área que utiliza tecnologia para automatizar processos e tarefas relacionadas ao comércio e negócios, tanto internos quanto externos (Alcides Junqueira, 2020). Com o aumento da competição no mercado, a utilização de tecnologias avançadas no gerenciamento de negócios tem se tornado cada vez mais importante para manter a eficiência e produtividade. Além disso, a automação comercial pode resultar em redução de custos, aumento de vendas e melhoria da satisfação dos clientes.

No Brasil, há diversas empresas que oferecem sistemas de automação comercial, como a Avanço Informática¹ em Belo Horizonte (MG), que oferece um sistema ERP² para supermercados, padarias, açougues e outros tipos de comércio, a Intellware Solutions³ em Itajubá (MG), que oferece serviços semelhantes, incluindo sistema para gestão e sistema PDV (Ponto de Venda) para vários segmentos do comércio, e a Criare Tecnologia⁴ que é a empresa utilizada como objeto de estudo neste trabalho, oferecendo também soluções como sistemas ERP para gestão de negócios e sistema PDV para o comércio.

O uso de sistemas automatizados também tem um impacto significativo na sociedade como um todo, permitindo que as empresas forneçam produtos e serviços de maneira mais rápida e eficiente, o que pode levar a um aumento do nível de vida e do bem-estar geral. Além disso, essas aplicações são capazes de coletar e armazenar grandes quantidades de dados, possibilitando aos empresários tomar decisões mais informadas e estratégicas.

Considerando isso, este trabalho tem como objetivo mostrar o desenvolvimento de aplicações focadas em automação comercial na Criare, discutir os desafios encontrados e demonstrar os principais resultados obtidos. O Capítulo 2 apresenta o referencial teórico, incluindo os principais conceitos e componentes técnicos utilizados no desenvolvimento dos projetos, enquanto o Capítulo 3 descreve como foram desenvolvidas as aplicações, incluindo imagens e explicações de funcionalidades. Por fim, o Capítulo 4 apresenta as conclusões e considerações finais sobre os resultados e dificuldades dos projetos.

¹ <https://avancoinfo.com.br/>

² ERP significa "Enterprise Resource Planning" ou "Planejamento de Recursos Empresariais", e é um sistema integrado de gestão que permite a automação e integração de processos empresariais (ORACLE, 2023).

³ <https://www.iws.com.br/>

⁴ <https://www3.criareinformatica.com.br/>

2 REFERENCIAL TEÓRICO

Neste Capítulo, são apresentados os conceitos, tecnologias, ferramentas e *frameworks* que foram usados para basear a metodologia.

2.1 Programação Orientada a Objetos

A Programação Orientada a Objetos (POO) é uma abordagem de programação que se baseia no conceito de objetos, que são entidades que possuem atributos e comportamentos. De acordo com Deitel e Deitel (2011), a POO permite a modelagem de sistemas complexos como se fossem objetos do mundo real, tornando o desenvolvimento de software mais fácil e organizado.

A POO é baseada em três pilares: encapsulamento, herança e polimorfismo. O encapsulamento permite a proteção de dados, pois os atributos são escondidos do acesso externo e só podem ser acessados por meio de métodos específicos. Já a herança permite a criação de relações entre objetos, onde um objeto pode herdar atributos e comportamentos de outro objeto. Por fim, o polimorfismo permite a criação de objetos que possuem a mesma interface, mas comportamentos diferentes.

Esta abordagem de programação é amplamente utilizada em várias linguagens, incluindo Java, Delphi, Python, C++, entre outras. A Figura 2.1 exibe um exemplo de código em Java que ilustra a utilização dos pilares de POO.

Nesse exemplo, a classe *Animal* representa um objeto genérico que possui um nome e um comportamento padrão de emitir som. Essa classe possui encapsulamento, pois o atributo nome é privado e só pode ser acessado por meio do método *getNome*.

A classe *Cachorro* e a classe *Gato* herdam a classe *Animal*. Dessa forma, as classes *Cachorro* e *Gato* herdam o atributo *nome* e o método *emitirSom*, mas elas podem sobrescrever o comportamento padrão do método com um comportamento específico para cada animal. Isso ilustra o conceito de herança e polimorfismo.

No método *main*, é criado um objeto da classe *Animal* e um objeto da classe *Cachorro*. Ao chamar o método *emitirSom* de cada objeto, o comportamento específico de cada animal é impresso na saída. Isso ilustra novamente o conceito de polimorfismo, pois mesmo que o objeto seja do tipo *Animal*, o método *emitirSom* utilizado será aquele sobrescrito na classe *Cachorro*.

Figura 2.1 – Exemplo de POO em Código Java

```
1 public class Animal {
2     private String nome;
3     public Animal(String nome) {
4         this.nome = nome;
5     }
6     public String getNome() {
7         return nome;
8     }
9     public void emitirSom() {
10        System.out.println("Animal emitindo som... ");
11    }
12 }
13
14 public class Cachorro extends Animal {
15     public Cachorro(String nome) {
16         super(nome);
17     }
18     @Override
19     public void emitirSom() {
20        System.out.println("Au au!");
21    }
22 }
23
24 public class Main {
25     public static void main(String[] args) {
26        Animal animal = new Animal("Animal");
27        Cachorro cachorro = new Cachorro("Bidu");
28        animal.emitirSom();
29        cachorro.emitirSom();
30    }
31 }
32
```

Fonte: Autor

2.1.1 Delphi

Delphi ¹ é uma IDE (*Integrated Development Environment*), ou seja, uma plataforma de desenvolvimento de software, que utiliza como linguagem de programação a linguagem Object Pascal. A IDE foi originalmente desenvolvida pela Borland e é agora mantida pela Embarcadero Technologies. É conhecida por sua facilidade de uso e alta produtividade, além de suportar várias plataformas, incluindo Windows, macOS, iOS e Android.

Uma das principais características do Delphi é a capacidade de criar interfaces gráficas de usuário (GUI, na sigla em inglês) de maneira rápida e fácil (SMITH, 2021). O Delphi possui componentes visuais prontos para uso, como botões, caixas de texto, listas, entre outros, que podem ser facilmente adicionados a uma aplicação e personalizados de acordo com as necessidades do usuário.

Outra característica importante do Delphi é sua utilização da linguagem de programação Object Pascal. A Object Pascal é uma linguagem de programação baseada na linguagem Pascal, que foi originalmente criada na década de 1970. A Object Pascal adicionou recursos adicionais de orientação a objetos à linguagem Pascal, tornando-a uma linguagem mais poderosa e flexível.

Além das características mencionadas acima, o Delphi também possui outros recursos avançados, incluindo suporte a *multi-threading*, o que permite a execução de várias tarefas ao mesmo tempo, suporte a banco de dados, integração com componentes externos e muito mais.

2.2 Banco de Dados e SQL

Para Elmasri e Navath (2019), um banco de dados é uma coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados. Ele é projetado para armazenar e gerenciar grandes quantidades de informações, permitindo que os usuários acessem e modifiquem esses dados de maneira segura e eficiente. Os bancos de dados podem ser usados em uma ampla variedade de aplicações, incluindo comércio eletrônico, sistemas de gerenciamento de estoques, sistemas de reserva de voos, entre outros. Eles podem ser implementados em várias plataformas e sistemas operacionais, e geralmente são projetados para serem escaláveis, confiáveis e seguros.

Os bancos de dados relacionais são compostos por tabelas, que representam as entidades do sistema, e cada tabela é formada por colunas que caracterizam cada atributo dessas entidades.

¹ <<https://www.embarcadero.com/br/products/delphi>>

Os dados são armazenados nas tabelas como linhas, onde cada linha corresponde a um objeto do sistema. Dessa forma, as informações ficam organizadas e relacionadas entre si, permitindo que as consultas e operações sejam realizadas de maneira eficiente.

Por exemplo, um supermercado que controle informações de vários produtos como código EAN, descrição, preço de venda e custo, contém uma tabela no banco de dados como representado no Quadro 2.1. Esta tabela contém as colunas "Identificador", "EAN", "Descrição", "Preço" e "Custo", e três registros representando três produtos no sistema. Cada um desses produtos tem suas informações representadas pelas colunas, além de ter um identificador único representado na coluna "Identificador".

Quadro 2.1 – Exemplo de tabela em um banco de dados

Produtos				
Identificador	EAN	Descrição	Preço	Custo
1	7891234567890	Arroz Integral 1kg	9,99	6,50
2	7894561231234	Feijão Carioca 1kg	7,99	4,50
3	7897891237895	Óleo de Soja 900ml	4,99	2,70
...

Fonte: Autor

2.2.1 SQL

SQL (*Structured Query Language*) é uma linguagem de programação utilizada para acessar e manipular dados em bancos de dados relacionais. Com ela, é possível realizar diversas operações, como consultar dados, inserir novas informações, atualizar registros existentes e excluir dados. Como descrito por Elmasri e Navath (2019), o SQL é uma linguagem de alto nível que permite aos usuários especificar quais dados desejam recuperar ou manipular, sem se preocupar com os detalhes de como o banco de dados está estruturado fisicamente.

Na Figura 2.2 há um exemplo de um CRUD (*Create, Read, Update, Delete*) de uma tabela de banco de dados feita em SQL. Neste código é primeiramente realizado a criação, com o código "CREATE TABLE", de uma tabela "Produtos" com as colunas "ID", "EAN", "DESCRICAO", "PRECO_VENDA" e "CUSTO", com suas respectivas características. Em seguida é feita a inserção, com o código "INSERT INTO", de três registros na tabela "Produtos", com os atributos de cada produto. É possível também identificar o código "SELECT * FROM" que lista todos os registros de uma tabela, no caso da tabela "Produtos". Em seguida é feita uma atualização, com o código "UPDATE", de um dos registros da tabela "Produtos", através de seu

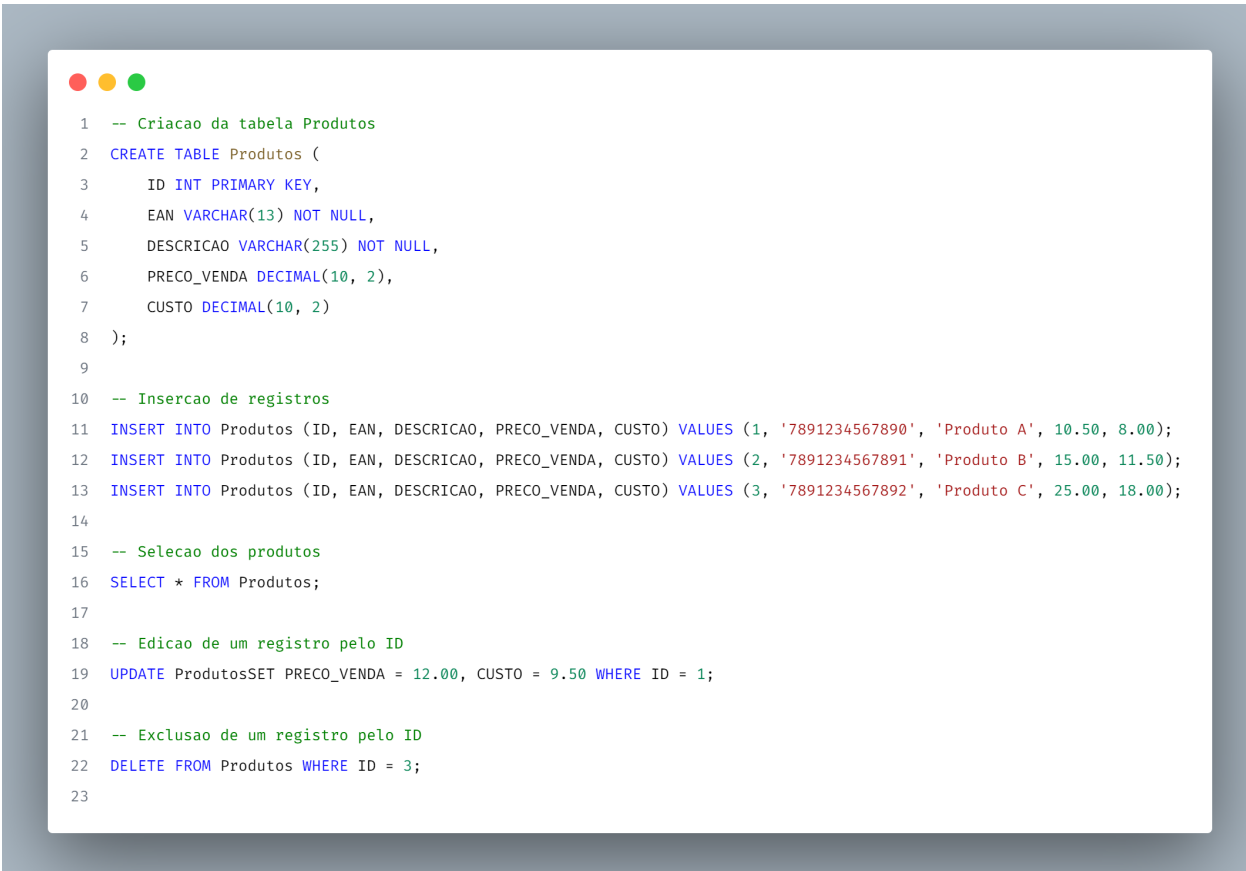
"ID"único. Por fim é excluído um registro desta mesma tabela, através do código "DELETE", utilizando também o "ID"único do produto.

2.2.2 MySQL

MySQL² é um sistema de gerenciamento de banco de dados relacional (RDBMS, na sigla em inglês) de código aberto que foi criado na Suécia em 1995 e é atualmente mantido pela empresa Oracle³. O mesmo é amplamente utilizado em aplicações web e é conhecido por sua escalabilidade, confiabilidade e facilidade de uso.

O MySQL suporta vários tipos de dados, incluindo números inteiros, decimais, strings, datas e horas, além de possuir uma linguagem de consulta SQL. Um dos recursos mais importantes do MySQL é a sua capacidade de lidar com grandes quantidades de dados e usuários simultâneos sem comprometer o desempenho.

Figura 2.2 – Tabela de Produtos em SQL



```
1  -- Criacao da tabela Produtos
2  CREATE TABLE Produtos (
3      ID INT PRIMARY KEY,
4      EAN VARCHAR(13) NOT NULL,
5      DESCRICAO VARCHAR(255) NOT NULL,
6      PRECO_VENDA DECIMAL(10, 2),
7      CUSTO DECIMAL(10, 2)
8  );
9
10 -- Insercao de registros
11 INSERT INTO Produtos (ID, EAN, DESCRICAO, PRECO_VENDA, CUSTO) VALUES (1, '7891234567890', 'Produto A', 10.50, 8.00);
12 INSERT INTO Produtos (ID, EAN, DESCRICAO, PRECO_VENDA, CUSTO) VALUES (2, '7891234567891', 'Produto B', 15.00, 11.50);
13 INSERT INTO Produtos (ID, EAN, DESCRICAO, PRECO_VENDA, CUSTO) VALUES (3, '7891234567892', 'Produto C', 25.00, 18.00);
14
15 -- Selecao dos produtos
16 SELECT * FROM Produtos;
17
18 -- Edicao de um registro pelo ID
19 UPDATE Produtos SET PRECO_VENDA = 12.00, CUSTO = 9.50 WHERE ID = 1;
20
21 -- Exclusao de um registro pelo ID
22 DELETE FROM Produtos WHERE ID = 3;
23
```

Fonte: Autor

² <<https://www.mysql.com/>>

³ <<https://www.oracle.com/>>

2.3 Padrão Arquitetural de Software

Um padrão arquitetural de software é uma solução geral para um problema recorrente de design de software, que fornece um conjunto de regras, diretrizes e práticas para organizar e estruturar o código. Os padrões arquiteturais ajudam a promover a clareza, a flexibilidade, a reusabilidade e a manutenibilidade do software. Eles são geralmente baseados em conceitos abstratos, mas podem ser implementados em várias linguagens de programação e tecnologias de desenvolvimento.

Os padrões arquiteturais são úteis porque ajudam a fornecer uma estrutura para o design e desenvolvimento de sistemas de software. Eles fornecem uma linguagem comum e uma base para a comunicação entre desenvolvedores e arquitetos, além de ajudar a melhorar a qualidade do software, tornando-o mais modular, escalável e fácil de manter (BUSCHMANN; HENNEY; SCHMIDT, 2007). Alguns exemplos de padrões arquiteturais são o MVC (*Model-View-Controller*), o REST (*Representational State Transfer*) e o microsserviço.

2.3.1 MVC

MVC (*Model-View-Controller*) é um padrão arquitetural de software que separa as responsabilidades em três componentes principais (DEV MEDIA, 2013):

- Modelo: representa os dados e a lógica de negócio da aplicação;
- Visão: é a interface com o usuário (geralmente a camada de apresentação);
- Controlador: é responsável por intermediar a comunicação entre a visão e o modelo, recebendo as requisições do usuário e atualizando a visão com os dados fornecidos pelo modelo.

O objetivo do MVC é facilitar o desenvolvimento de software ao separar as preocupações em componentes independentes e bem definidos, permitindo que os desenvolvedores trabalhem em paralelo em diferentes partes da aplicação. Além disso, o padrão MVC torna a aplicação mais fácil de entender e manter, uma vez que cada componente é responsável por uma parte bem definida da lógica do programa (DEV MEDIA, 2013).

2.4 Desenvolvimento Web e Mobile

Nos dias de hoje, o desenvolvimento web e mobile tornou-se cada vez mais relevante e essencial para as empresas que buscam se manter competitivas no mercado. Segundo pesquisa realizada pela Statista, em 2022, mais de 167,54 milhões de pessoas no Brasil acessaram a internet por meio de qualquer tipo de dispositivo móvel. Em 2027, estima-se que esse número chegue a cerca de 188,8 milhões de usuários de internet móvel, representando um aumento de aproximadamente 12,7% em relação a 2022 (STATISTA, 2023).

A internet e os dispositivos móveis se tornaram uma grande fonte de acesso à informação e negócios, o que fez com que muitas empresas migrassem para essas soluções visando atender as necessidades dos usuários. O desenvolvimento web e mobile envolve a criação de aplicativos e sites que possam ser acessados a partir de diferentes dispositivos, como *smartphones*, *tablets* e computadores.

No contexto do desenvolvimento web, a utilização de *frameworks* como React, Angular e Vue tem se tornado cada vez mais comum, permitindo a criação de aplicações web modernas e escaláveis. Já no desenvolvimento mobile, o uso de tecnologias como o React Native, que permite a criação de aplicativos para iOS e Android com uma única base de código, tem ganhado destaque nos últimos anos.

Além dos *frameworks* mencionados, o desenvolvimento web e mobile também depende da parte do servidor para lidar com a lógica de negócios, autenticação, armazenamento de dados, entre outras funcionalidades. Nesse sentido, Node.js tem se destacado como uma das principais tecnologias para a construção de servidores web e APIs (FREEMAN; ROBSON, 2016).

2.4.1 Node JS

Node.js é uma plataforma de desenvolvimento de aplicações em tempo real baseada em JavaScript⁴. Ele é executado no servidor e permite a criação de aplicações escaláveis, com alta performance e capazes de lidar com uma grande quantidade de dados e de conexões simultâneas (NODE.JS FOUNDATION, 2023).

Essa plataforma foi lançada pela primeira vez em 2009 e desde então vem se tornando cada vez mais popular, principalmente no desenvolvimento de aplicações web e mobile.

⁴ JavaScript é uma linguagem de programação interpretada, com tipagem dinâmica, que é utilizada principalmente no desenvolvimento web para tornar as páginas mais interativas. A linguagem é executada no lado do cliente (navegador) e permite a manipulação do conteúdo da página, interação com o usuário e comunicação com servidores web (Mozilla Developer Network, 2023)

A arquitetura do Node.js é baseada em eventos e entrada/saída (I/O) não bloqueante. Isso significa que ele é projetado para lidar com muitas solicitações simultaneamente, sem bloquear o fluxo do programa.

Em uma arquitetura de I/O bloqueante, uma solicitação pode fazer com que o programa inteiro pare de responder até que a solicitação seja concluída. Isso pode levar a problemas de desempenho e escalabilidade, especialmente em aplicativos com muitas solicitações simultâneas.

Já a arquitetura baseada em eventos significa que o Node.js utiliza *callbacks* (funções de retorno) para lidar com eventos assíncronos. Em vez de aguardar a conclusão de uma operação antes de prosseguir, o Node.js simplesmente executa uma função de retorno de chamada quando a operação é concluída. Essa abordagem permite que o servidor continue a executar outras tarefas enquanto aguarda a conclusão da operação assíncrona, aumentando a eficiência e escalabilidade do programa.

2.4.2 React JS

React é uma biblioteca de código aberto para construir interfaces de usuário em aplicações web. Foi criada pelo antigo Facebook, atualmente Meta, em 2011, e é amplamente utilizada na indústria de desenvolvimento de software. Ele permite que você escreva componentes de interface de usuário em uma única linguagem, o JavaScript, que pode ser integrado a outras tecnologias web (REACTJS, 2023b).

O React tem se tornado cada vez mais popular no desenvolvimento web, devido à sua capacidade de criar interfaces de usuário complexas com um código limpo e organizado. Ele também possui uma grande comunidade de desenvolvedores e uma vasta seleção de bibliotecas e ferramentas adicionais, o que torna mais fácil a criação de aplicações web escaláveis e de alta qualidade. Além disso, o React também é utilizado no desenvolvimento de aplicações móveis através do React Native, permitindo que os desenvolvedores criem aplicativos móveis para iOS e Android com uma única base de código.

A Figura 2.3 demonstra um componente simples em React chamado Contador, que utiliza o Hook ⁵ de gerenciamento de estado "useState" para definir uma variável de estado cha-

⁵ Hooks são uma nova adição ao React 16.8 que permitem o uso de estados e outros recursos do React em componentes funcionais, que antes só eram possíveis em componentes de classe.

mada "count". O valor desse estado é exibido em uma tag de parágrafo do HTML e é incrementado quando o usuário clica no botão.

No desenvolvimento com React é comum o uso da extensão de sintaxe JSX que permite a criação de elementos da interface do usuário de forma declarativa. Com ele, é possível escrever o código HTML e JavaScript no mesmo arquivo, o que facilita a leitura e manutenção. O JSX é usado em conjunto com o React para criar componentes personalizados. Apesar de parecer uma mistura de linguagens, o JSX é transformado em JavaScript puro durante a compilação, tornando-o compatível com a linguagem e permitindo sua utilização em qualquer ambiente que suporte JavaScript (REACTJS, 2023a).

Em React, componentes são elementos fundamentais para a criação de interfaces de usuário. Eles são basicamente funções ou classes JavaScript que encapsulam uma parte da interface, permitindo que essa parte seja reutilizada em diferentes locais do sistema. Cada componente é uma peça autônoma e independente, responsável por uma parte específica da lógica de negócios e visualização. Isso permite que as interfaces de usuário em React sejam facilmente escaláveis, mantidas e testadas, já que cada componente pode ser desenvolvido e testado separadamente do restante da aplicação. Além disso, cada componente possui seu próprio estado interno ⁶, ou seja, as alterações em um componente não afetam os demais da página, e quando um estado é atualizado em um componente, apenas esse componente é recarregado, trazendo melhorias de performance para o site.

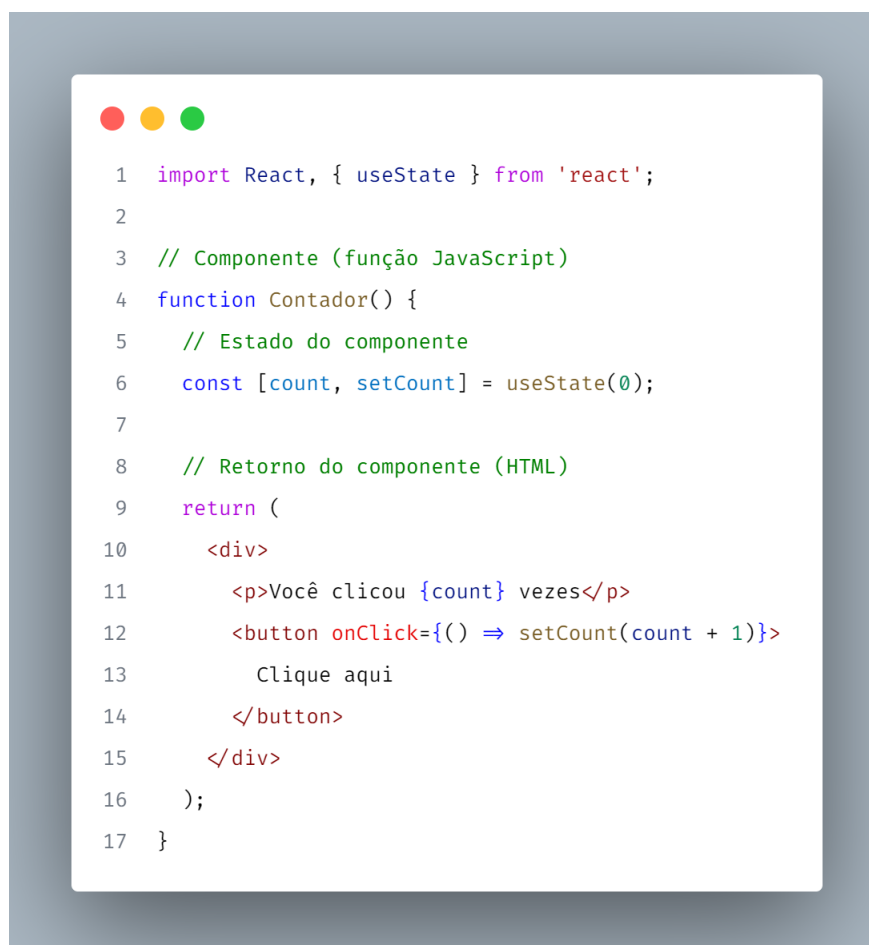
2.4.3 TypeScript

TypeScript é uma linguagem de programação que é baseada em JavaScript, mas inclui recursos adicionais, como tipagem estática e interface. Isso permite a criação de aplicações mais confiáveis e de melhor qualidade, pois ajuda a encontrar erros no desenvolvimento e não somente em tempo de execução (TYPESCRIPT, 2023).

Além disso, o TypeScript é altamente compatível com o JavaScript, o que significa que a maioria do código JavaScript existente pode ser usado sem mudanças. Isso torna a migração para o TypeScript mais fácil e suave. Ele também já pode ser utilizado em projetos front-end com React.

⁶ Estados em React são valores que podem ser atualizados e que determinam como um componente é renderizado em um determinado momento.

Figura 2.3 – Exemplo de Código em React

A screenshot of a code editor window with a white background and a grey border. At the top left, there are three colored circles (red, yellow, green) representing window control buttons. The code is written in a light blue monospace font. It shows the import of React and useState, a function definition for 'Contador', and the use of useState to manage a 'count' state. The return statement includes a div with a paragraph and a button that updates the count when clicked.

```
1 import React, { useState } from 'react';
2
3 // Componente (função JavaScript)
4 function Contador() {
5   // Estado do componente
6   const [count, setCount] = useState(0);
7
8   // Retorno do componente (HTML)
9   return (
10    <div>
11      <p>Você clicou {count} vezes</p>
12      <button onClick={() => setCount(count + 1)}>
13        Clique aqui
14      </button>
15    </div>
16  );
17 }
```

Fonte: Autor

Na Figura 2.4 é demonstrado um exemplo de código utilizando TypeScript. Neste exemplo, é definida uma interface "Pessoa" que especifica a tipagem dos campos "nome", "idade" e um campo opcional "e-mail". Em seguida, é criada a função "imprimirPessoa" que recebe um objeto do tipo Pessoa e imprime os valores de seus campos. Finalmente, são criados dois objetos "pessoa1" e "pessoa2" que são passados como argumentos para a função "imprimirPessoa". Dessa forma, caso o desenvolvedor tentasse acessar algum campo inexistente na interface "Pessoa", um erro já seria exibido em tempo de desenvolvimento.

2.5 Metodologias Ágeis

De acordo com Sommerville (2019), metodologias ágeis são abordagens de desenvolvimento de software que valorizam a colaboração e comunicação contínua entre os membros da equipe, bem como a adaptação constante às mudanças de requisitos e prioridades. Essas metodologias enfatizam a entrega frequente de funcionalidades úteis e a busca pela satisfação

Figura 2.4 – Exemplo de Código em TypeScript

```
1 interface Pessoa {
2   nome: string;
3   idade: number;
4   email?: string;
5 }
6
7 function imprimirPessoa(pessoa: Pessoa) {
8   console.log(`Nome: ${pessoa.nome}`);
9   console.log(`Idade: ${pessoa.idade}`);
10  if (pessoa.email) {
11    console.log(`Email: ${pessoa.email}`);
12  }
13 }
14
15 const pessoa1 = { nome: "João", idade: 25 };
16 imprimirPessoa(pessoa1);
17
18 const pessoa2 = { nome: "Maria", idade: 30, email: "maria@gmail.com" };
19 imprimirPessoa(pessoa2);
```

Fonte: Autor

do cliente, além de encorajar a auto-organização e a autogestão da equipe de desenvolvimento. Diferentemente de abordagens mais tradicionais, as metodologias ágeis buscam minimizar a burocracia e o excesso de documentação, dando mais ênfase à entrega contínua de software e à obtenção de *feedback* dos usuários para guiar o processo de desenvolvimento.

2.5.1 Framework Scrum

Scrum é uma metodologia ágil para gerenciamento de projetos de software que foi desenvolvido na década de 1990 e se tornou uma das metodologias ágeis mais populares no mundo. Ele é baseado no método científico do empirismo visando lidar com as incertezas e imprevisibilidades que surgem durante o processo de desenvolvimento (SCRUM.ORG, 2023). De acordo com Schwaber e Sutherland (2014), os três pilares que sustentam a implementação do controle de processo empírico são:

- **Transparência:** significa que todos os aspectos relevantes do processo devem ser visíveis e compreensíveis para todos os envolvidos, tanto interna quanto externamente ao time Scrum. Isso inclui aspectos como objetivos, resultados de inspeções, planos e progresso do trabalho. A transparência é necessária para garantir que todos os membros do time tenham uma compreensão comum do trabalho que está sendo realizado e do que ainda precisa ser feito.
- **Inspeção:** refere-se ao ato de verificar regularmente os resultados e o progresso do trabalho do time Scrum em relação aos objetivos definidos. Isso inclui inspecionar o produto em si, o plano do produto, o trabalho do time e o processo utilizado para criar o produto. A inspeção é necessária para detectar problemas e oportunidades de melhoria em tempo hábil.
- **Adaptação:** refere-se à capacidade do time Scrum de ajustar e adaptar seu trabalho e processo de acordo com as inspeções realizadas. Isso significa que o time deve ser capaz de identificar problemas e oportunidades de melhoria e, em seguida, implementar mudanças para corrigir os problemas e aproveitar as oportunidades. A adaptação é necessária para garantir que o produto final atenda aos requisitos e expectativas do cliente de forma eficaz e eficiente.

Os ritos do Scrum são eventos pré-definidos que acontecem em intervalos regulares durante um projeto, com o objetivo de criar transparência, inspeção e adaptação ao processo. São eles (SCHWABER; SUTHERLAND, 2014):

- *Sprint*: período de tempo para realizar um conjunto de atividades e entregar um incremento de produto. É a unidade básica de trabalho do Scrum.
- *Planning*: reunião em que a equipe de desenvolvimento estabelece o que pode ser entregue na próxima sprint, criando o plano de trabalho detalhado.
- *Daily Scrum*: reunião diária de no máximo 15 minutos para sincronizar a equipe, discutir o progresso, identificar impedimentos e planejar o trabalho para o dia.
- *Review*: reunião para demonstrar o que foi realizado na sprint, receber feedback e adaptar o backlog do produto.

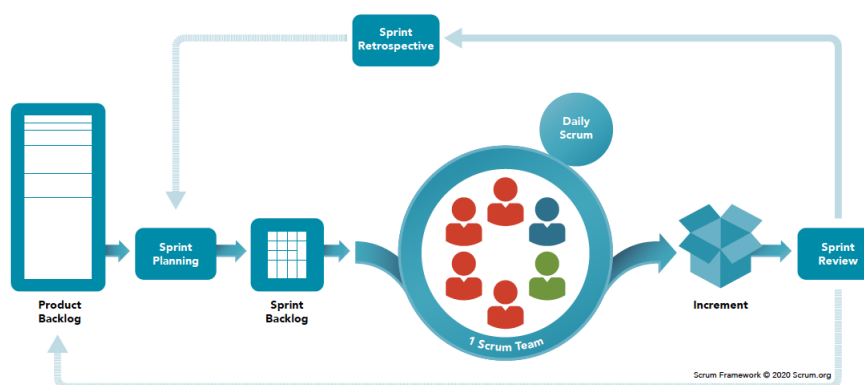
- *Retrospective*: reunião para refletir sobre a sprint que acabou de terminar e identificar oportunidades de melhoria para a próxima sprint.

Os papéis do Scrum são responsabilidades específicas que são atribuídas aos membros da equipe de desenvolvimento, a fim de garantir que o projeto progrida de maneira organizada e objetiva. Existem três papéis principais no Scrum (SCHWABER; SUTHERLAND, 2014):

- *Scrum Master* (SM): é o responsável por garantir que o processo Scrum seja seguido corretamente, removendo obstáculos para a equipe e garantindo que a equipe se comunique efetivamente.
- *Product Owner* (PO): é o responsável por definir as funcionalidades do produto, gerenciar o *backlog* de produto e garantir que o produto final atenda às necessidades do cliente.
- *Time de Desenvolvimento*: é composto por profissionais que são responsáveis por construir o produto, realizando as atividades necessárias para transformar o *backlog* do produto em incrementos do produto a cada *sprint*.

A Figura 2.5 ilustra o ciclo completo de funcionamento de uma *Sprint* no Scrum.

Figura 2.5 – Demonstração do Scrum



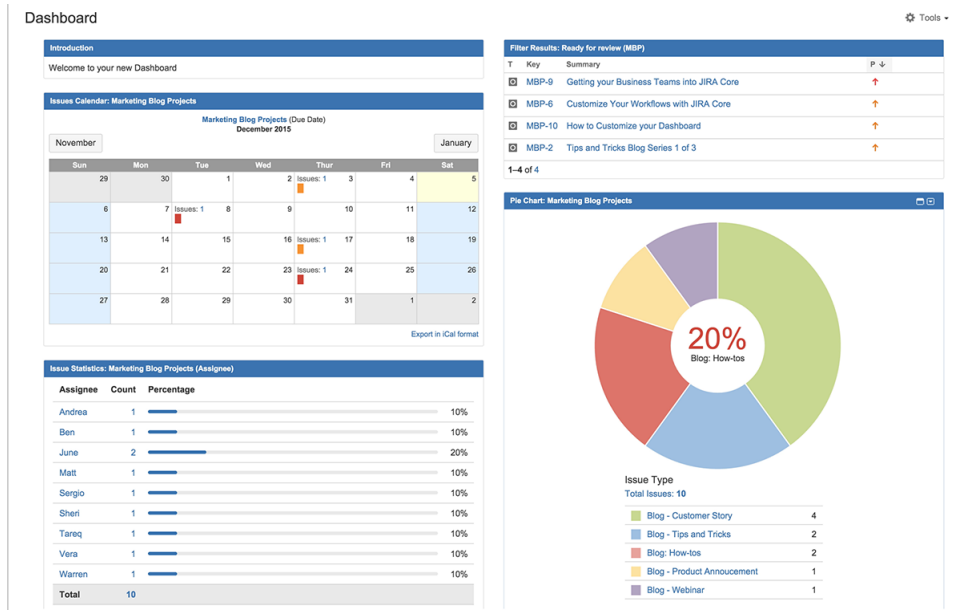
Fonte: Scrum.org (2023)

2.6 Jira Software

Jira Software é uma plataforma de gerenciamento de projetos desenvolvida pela Atlassian. É utilizado por equipes de desenvolvimento de software para acompanhar e gerenciar as atividades relacionadas aos projetos, incluindo *bugs*, tarefas, histórias de usuário e outras demandas. Jira oferece uma ampla gama de recursos, incluindo relatórios, *dashboards*, gestão de

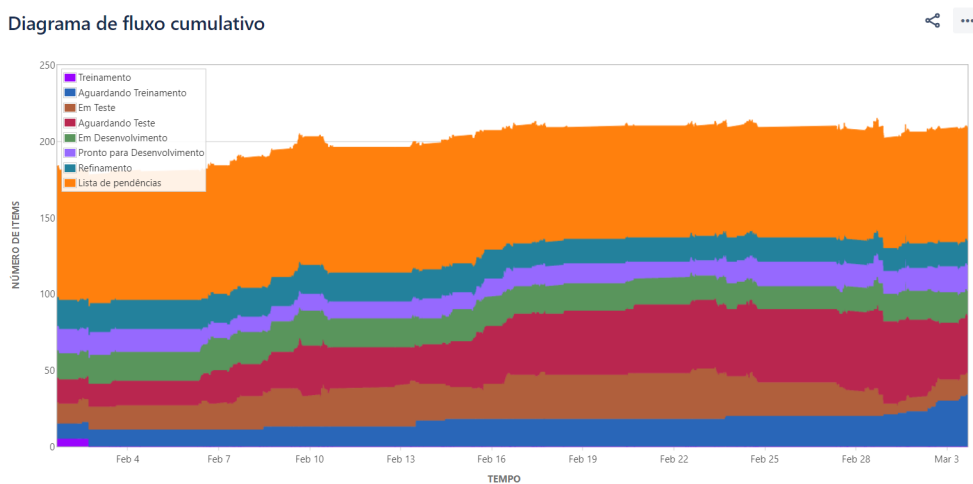
backlogs e integração com outras ferramentas de desenvolvimento de software (ATLASSIAN, 2023c). A Figura 2.6 demonstra uma *dashboard* oferecida pelo Jira, demonstrando um calendário com as datas das atividades, uma tabela de estatísticas mostrando a quantidade de atividades por colaborador e um gráfico dividindo as atividades por projetos. Já a Figura 2.7 exemplifica um relatório em forma de gráfico gerado pela plataforma, mostrando o status dos *cards*.

Figura 2.6 – Demonstração de uma Dashboard no Jira



Fonte: Atlassian (2023a)

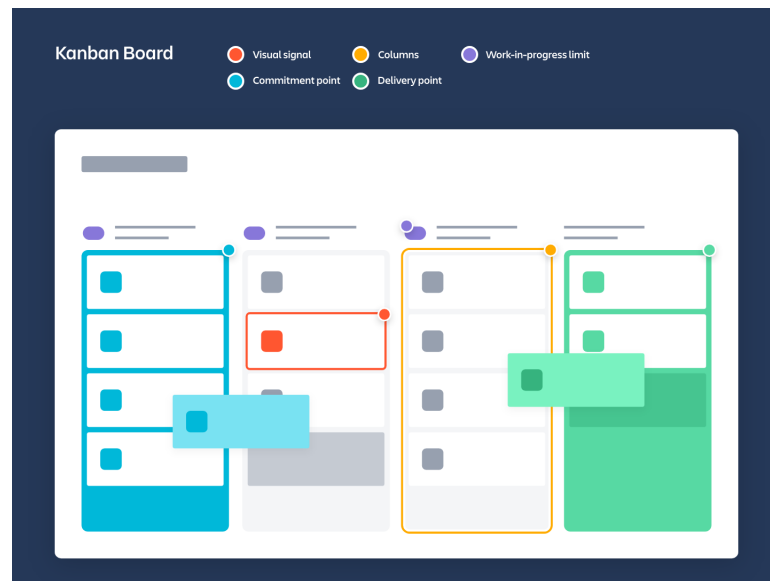
Figura 2.7 – Exemplo de Relatório em Gráfico no Jira



Fonte: Autor

A plataforma Jira é baseada em quadros de atividades, como por exemplo o quadro Kanban ⁷, o que significa que as tarefas são representadas visualmente em colunas e os usuários podem facilmente arrastar e soltar itens de uma coluna para outra para indicar o progresso. Isso torna o processo de gerenciamento de projetos mais simples e ágil, permitindo que as equipes trabalhem de maneira mais eficiente e alinhem suas expectativas. A Figura 2.8 representa visualmente um quadro Kanban.

Figura 2.8 – Representação do Quadro Kanban



Fonte: (ATLASSIAN, 2023b)

⁷ Quadro Kanban é um quadro dividido em colunas que representam as etapas do processo e em cartões que representam as tarefas a serem realizadas

3 METODOLOGIA DE DESENVOLVIMENTO E RESULTADOS

Neste Capítulo, será explorado o processo de desenvolvimento de alguns projetos implementados, suas motivações e os principais resultados obtidos.

3.1 Metodologia de Desenvolvimento

As aplicações foram desenvolvidas na Criare Tecnologia, que é uma empresa especializada em soluções de tecnologia para a automação comercial, atendendo clientes em todo o país. Para atender às necessidades dos clientes, a empresa desenvolve sistemas completos para o atacado, varejo, supermercados e redes de materiais de construção, incluindo o Sistema Integrado de Administração (SIA), o SIA PDV (Ponto de Venda) e atualmente o novo projeto SIA WEB.

O trabalho foi desenvolvido durante o período de estágio na empresa Criare Tecnologia, e foi possível participar do processo de desenvolvimento de alguns projetos implementados pela empresa, seguindo a metodologia Scrum com divisão de tarefas em *backlogs* para ciclos de trabalho quinzenais ou mensais. Esses projetos foram desenvolvidos em equipes compostas por colaboradores dos setores de desenvolvimento e testes.

Inicialmente foi necessário um período de treinamento, dividido em duas partes: treinamento técnico para estudar as tecnologias necessárias para o desenvolvimento das aplicações e treinamento interno para entender a usabilidade do sistema (SIA), as necessidades dos clientes e as demais soluções oferecidas pela empresa. Essa etapa foi essencial para integração com a equipe e entendimento da dinâmica de trabalho da empresa.

Conforme mencionado anteriormente, embora a empresa possua sistemas próprios, o primeiro projeto foi para desenvolver um software personalizado para um cliente específico, como um projeto individual. Esse projeto foi desenvolvido em paralelo com o período de treinamento, permitindo a aplicação prática dos conceitos aprendidos.

Após concluir o projeto para o cliente específico e finalizar os treinamentos, o estagiário passou a integrar o time de desenvolvimento do SIA, o principal software da empresa, composto por seis desenvolvedores liderados pelo Diretor de Produtos que atua como *Product Owner* (PO) e *Scrum Master* (SM). Toda a equipe adota os princípios do Scrum e são feitas reuniões diárias para alinhar as atividades planejadas e realizadas, garantindo a transparência e agilidade do processo de desenvolvimento.

Durante o período nessa equipe, por se tratar de um sistema extenso, foram realizadas diversas implementações em diferentes módulos da aplicação, incluindo melhorias, correções

de *bugs* e desenvolvimento de novas soluções para o sistema. A seguir serão descritas algumas dessas implementações, assim como seus principais resultados.

3.2 Exportador Viva Broker

O primeiro projeto desenvolvido, como citado anteriormente, envolveu a implementação de uma aplicação à parte, que foi uma solicitação exclusiva do cliente Viva Broker. Este cliente tinha a necessidade de exportar dados do sistema em questão para outros dois sistemas terceiros, o que anteriormente era feito manualmente. A implementação de um sistema exportador automatizado foi solicitada com o objetivo de facilitar e acelerar o trabalho do cliente.

A exigência do cliente era que os dados exportados estivessem no formato de arquivo '.txt' e que seguissem os layouts específicos fornecidos pelos sistemas de destino. Esses layouts estabeleciam a estrutura do arquivo, incluindo a nomenclatura, o espaçamento, o número de caracteres, entre outros aspectos importantes.

A Figura 3.1 demonstra um exemplo de informações fornecidas por um dos documentos de layout.

Um exemplo de como deveria ficar o arquivo exportado é demonstrado na Figura 3.2, fornecido pelo mesmo documento do layout.

O layout do segundo sistema foi disponibilizado via planilha, com informações semelhantes porém solicitando padrão e estrutura diferentes do primeiro sistema, como mostra a Figura 3.3.

Esse projeto foi desenvolvido de forma independente e individual, seguindo as metodologias ágeis do Scrum. Durante esse período, foi importante a participação ativa das reuniões diárias juntamente com os demais desenvolvedores, a fim de obter auxílios necessários mediante aos impedimentos que fossem surgindo.

Ao decorrer deste projeto, foi importante também manter um contato direto com o cliente com o objetivo de esclarecer quaisquer dúvidas e garantir o atendimento das expectativas. Além disso, também existiu uma comunicação constante com colegas mais experientes para solucionar eventuais questões técnicas e garantir a entrega do projeto dentro dos prazos estabelecidos.

O projeto foi desenvolvido em Delphi e utilizou o sistema de gerenciamento de banco de dados MySQL. O sistema apresenta uma interface para o usuário selecionar algumas informações importantes como mostrado na figura 3.4, e detalhadas a seguir:

Figura 3.1 – Exemplo de Layout (Sistema 1)

Arquivo de Clientes (Pontos de venda)

Nomenclatura:

Prefixo = "CLIENTES";

A sigla do fornecedor é informada no processo de implantação;

Data no formato Dia + Mês + Ano + Hora + Minuto + Segundo + Milésimo de segundo;

Extensão do nome do arquivo = "TXT";

Exemplo: Arquivo de estoque gerado no dia "30/06/2020" às 15:30:05.130 para um fornecedor com sigla igual a "ESL" teremos:

CLIENTESESL**30062020153005130**.txt;

Estrutura do arquivo:

Registro **cabeçalho** do arquivo de clientes:

Cardinalidade: Deve ser a primeira linha do arquivo e só pode existir um registro deste tipo no arquivo.

Campo	Tipo	Tamanho	Inicial	Formatação	Conteúdo Esperado	Obrigatório	Validação
Tipo do Registro	A	001	001	X(1)	"H"	Sim	Não
Identificador	A	007	002	X(7)	"PDV10"	Sim	Não
CNPJ Fornecedor	A	014	009	9(14)	CNPJ sem Máscara	Sim	Sim

Registro de **clientes**:

Cardinalidade: Repete-se para todos os clientes cadastrados para um determinado fornecedor.

Campo	Tipo	Tamanho	Inicial	Formatação	Conteúdo Esperado	Obrigatório	Validação
Tipo do Registro	A	001	001	X(1)	"D"	Sim	Não
CNPJ Agente Distribuição	A	014	002	9(14)	CNPJ sem máscara	Sim	Sim
Identificação do cliente	A	018	016	X(18)	CNPJ ou CPF, sem máscara	Sim	Sim
Razão social do cliente	A	040	034	X(40)	Razão social do cliente	Sim	Não

Fonte: Autor

- Empresa: Obrigatório. Indica de qual empresa os dados devem ser selecionados;
- Fornecedor: Indica, caso selecionado, qual o fornecedor desejado;
- Categoria: Indica, caso selecionado, qual a categoria dos produtos;
- Família: Indica, caso selecionado, qual a família dos produtos;
- Período de vendas: Indica o intervalo de datas para os dados de vendas;

Figura 3.4 – Interface do Exportador

Exportador Nestle

Selecione a empresa*

Gerar Arquivos

Clientes Produtos Força de Vendas
 Estoques Vendas

Cód. Fornecedor (em branco = sem filtro) ... Nome: CNPJ:

Selecione a categoria (em branco = sem filtro)

Selecione a família (em branco = sem filtro)

Período de vendas

Data Inicial 12/02/2023 Data Final 12/02/2023

Local de Salvamento

Selecionar C:\Users\otavi\Desktop\

Progresso do arquivo

Progresso geral

EXPORTAR

Fonte: Autor

thread com a *thread* principal da aplicação, permitindo que o método "OnNotificaInterface" seja executado nela.

Por sua vez, o método "OnNotificaInterface", demonstrado na Figura 3.5, realiza as alterações visuais na barra de progresso e é repassado para o *data module*, assim o progresso vai acompanhando o *loop* dos registros no banco de dados, onde o parâmetro "pMax" recebe o total de registros e o parâmetro "pPosicao" recebe a posição do registro atual.

Essa abordagem é especialmente útil em processos que envolvem operações de entrada e saída, como leitura e escrita em bancos de dados ou arquivos, que podem ser bastante demorados. Dessa forma, é garantido que a aplicação permaneça responsiva e não trave enquanto o processo é executado em segundo plano.

Figura 3.5 – Método "OnNotificaInterface"

```

1 procedure TFrmExportadorAryzta.OnNotificaInterface(pMax, pPosicao: Integer);
2 begin
3   pbIndividual.Max := pMax;
4   pbIndividual.Position := pPosicao;
5   pbIndividual.Step := 1;
6 end;

```

Fonte: Autor

Figura 3.6 – Demonstração da Thread em Delphi

```

1 procedure TFrmExportadorAryzta.GerarVendedores;
2 var
3   codEmpresa: string;
4 begin
5   codEmpresa := cxlccbEmpresas.EditValue;
6   lblPbIndividual.Caption := 'Progresso do arquivo de VENDEDORES ... ';
7   TThread.CreateAnonymousThread(
8     procedure
9     var
10      DMVendedores: TDMVendedores;
11    begin
12      DMVendedores := TDMVendedores.Create(nil);
13      try
14        // Executa a exportacao em uma thread anonima
15        DMVendedores.ExportarVendedores(edtCaminho.Text, codEmpresa);
16
17        // Sincroniza com a Thread Principal para incrementar a barra de progresso
18        TThread.Synchronize(nil,
19          procedure
20          begin
21            DMVendedores.NotificaInterface := OnNotificaInterface;
22          end);
23        finally
24          DMVendedores.Free;
25        end;
26      end).Start;
27 end;

```

Fonte: Autor

Como foi dito, foram desenvolvidos dois exportadores com diferentes layouts, porém a interface do exportador e os principais métodos foram reaproveitados via herança e polimorfismo, seguindo os conceitos de POO. Para indicar qual dos dois exportadores será utilizado, ao iniciar a aplicação há a possibilidade de escolha para o usuário, como mostra a Figura 3.7.

Figura 3.7 – Menu de Seleção



Fonte: Autor

O projeto Exportador Viva Broker foi finalizado e entregue ao cliente, sem apresentar nenhum erro durante o seu uso. A solução desenvolvida foi capaz de atender aos requisitos, permitindo a exportação dos dados necessários para outras plataformas. Em suma, o Exportador Viva Broker foi bem-sucedido em cumprir o seu propósito e atender às expectativas do cliente.

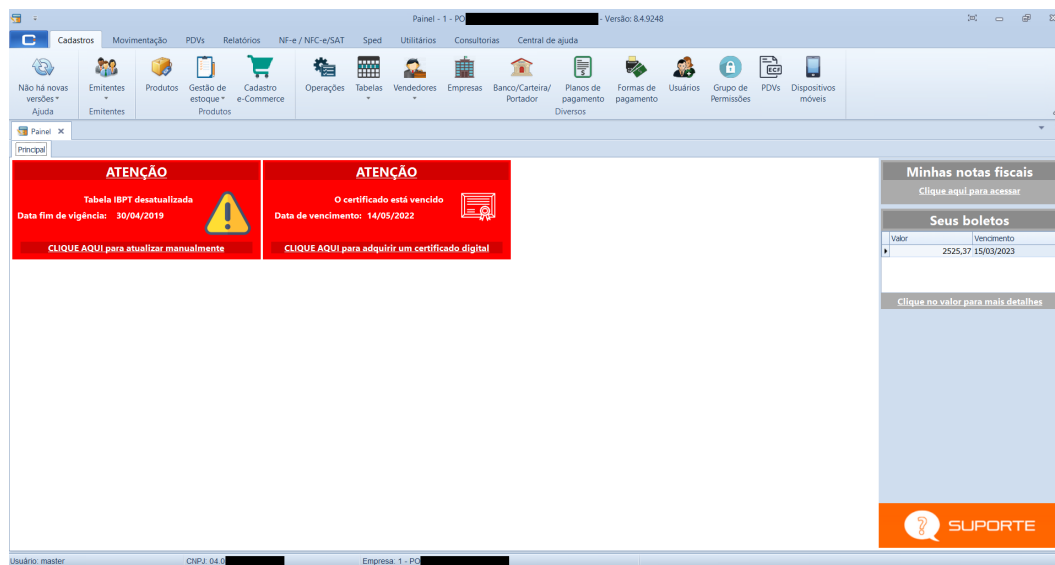
3.3 SIA

Essa seção descreve algumas implementações realizadas no sistema SIA, principal aplicação ofertada pela empresa. O SIA é um sistema ERP completo que automatiza e controla todos os processos de uma empresa, incluindo o controle fiscal, financeiro, estoque, vendas, compras e diversos outros módulos. Além disso, o sistema possui funcionalidades como a geração automática de notas fiscais, integração com sistemas de pagamento e envio de e-mails automáticos. Outra funcionalidade importante é o módulo de *Business Intelligence* (BI)³, que permite aos usuários criar e analisar relatórios personalizados com facilidade.

A Figura 3.8 mostra a tela inicial do sistema. Nela é possível identificar alguns menus de cadastro na barra superior e alguns avisos importantes no painel central.

³ BI é uma área da tecnologia que utiliza a análise de dados para ajudar empresas e organizações a tomarem decisões estratégicas mais informadas.

Figura 3.8 – SIA - Página Inicial



Fonte: Autor

A Figura 3.9 demonstra o funcionamento de um dos módulos do sistema que é o cálculo automático de DRE. E na imagem também é possível observar alguns dos menus de movimentação disponíveis na barra superior.

3.3.1 Recebimento de Contas com PixCriare

Com o surgimento do PIX, se fez necessário o desenvolvimento de um novo recurso para o sistema SIA, que consistiu na adição de uma nova opção de forma de pagamento, PixCriare, no módulo de Recebimento de Contas. PixCriare é um serviço de pagamentos via PIX da Criare em parceria com a plataforma PenseBank⁴. Para criar este recurso, primeiramente foi adicionado a nova opção no combo de Forma de pagamento na tela Recebimento de contas, como mostra a Figura 3.10.

Ao selecionar essa opção e clicar no botão "F2-Receber", uma nova tela é exibida, mostrada na Figura 3.11, contendo informações do cliente, valor cobrado e um *QR Code*⁵ para realizar o pagamento via celular. Além disso, há a outras opções para o usuário realizar o pagamento, sendo elas:

⁴ <https://www.pensebank.com.br/>

⁵ QR Code é um código de barras bidimensional que pode ser escaneado por um dispositivo móvel para acessar informações, como um link para um site ou um texto.

Figura 3.9 – SIA - Página Inicial

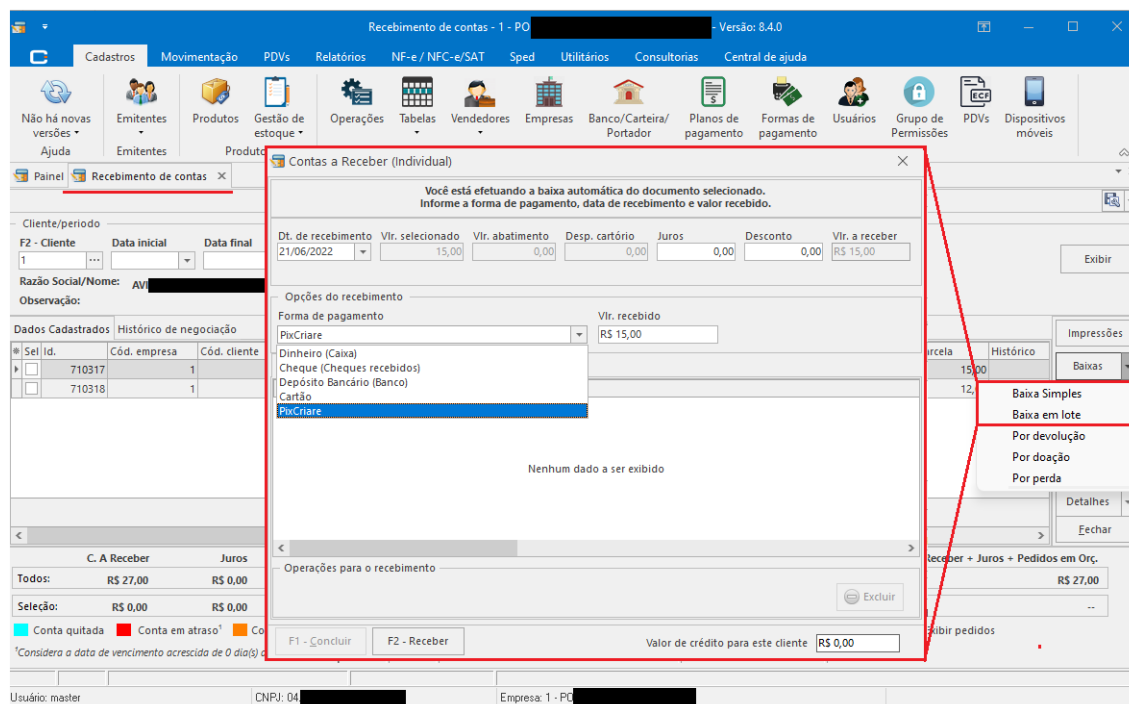
Descrição	01/01/2020 à 01/01/2023	%
RECEITA TOTAL	58.631.397,82	
(+) RECEITAS	58.631.397,82	
RECEITA POR GRUPO	58.546.186,02	99,85
OUTRAS RECEITAS	0,00	0,00
JUROS RECEBIDOS	85.211,80	0,15
(-) CANCELADOS / DEVOLVIDOS	4.537.406,67	7,74
CANCELADOS	2.685.467,13	4,58
DEVOLUÇÕES	1.851.939,54	3,16
(=) RECEITA BRUTA	54.093.991,15	100,00
(-) DEDUÇÕES DA RECEITA BRUTA	0,00	0,00
IMPOSTOS	0,00	0,00
TAXA ADMINISTRADORA DE CARTÕES	0,00	0,00
(=) RECEITA LÍQUIDA	54.093.991,15	100,00
(-) CUSTO DOS PRODUTOS VENDIDOS E SERVIÇOS PRI	48.656.138,82	89,95
C.M.V POR GRUPO	48.656.138,82	89,95
(=) RESULTADO BRUTO	5.437.852,33	10,05
(-) DESPESAS OPERACIONAIS	440.662,66	0,81
DESPESAS PESSOAL OP	0,00	0,00
MATERIAL DE USO E CONSUMO OPERACIONAL	0,00	0,00
TAXAS/MULTAS/TRIBUTAÇÕES OPERACIONAIS	0,00	0,00
INADIMPLENTES	0,00	0,00
CHEQUES DEVOLVIDOS	0,00	0,00
DESCONTOS CONCEDIDOS (RECEBIMENTOS)	0,00	0,00
QUEBRAS / PERDAS	405.578,12	0,75
JUROS PAGOS	35.084,54	0,06
(=) RESULTADO OPERACIONAL	4.997.189,67	9,24
DESPESAS OU RECEITAS NÃO OPERACIONAIS	0,00	0,00
DESPESAS - INVESTIMENTOS	0,00	0,00
RECEITA - INVESTIMENTOS	0,00	0,00
RECEITA - FINANCIAMENTOS	0,00	0,00
= RESULTADO ANTES DO IMPOSTO DE RENDA	4.997.189,67	9,24
(-) PROVISÃO PARA O IMPOSTO DE RENDA E CONTRIB	0,00	0,00
IR E CSLL	0,00	0,00
= RESULTADO LÍQUIDO DO EXERCÍCIO	4.997.189,67	9,24

Fonte: Autor

- Copiar Código: ao clicar nessa opção o código de pagamento é automaticamente copiado para a área de transferência do dispositivo, sendo possível realizar o pagamento via Pix Copia e Cola no próprio computador;
- Enviar para WhatsApp: para utilizar esta opção, basta informar o número de destino no campo indicado e clicar no botão para enviar. O sistema tentará abrir o aplicativo do WhatsApp no computador do usuário e, caso não esteja instalado, abrirá o WhatsApp Web ⁶ com o chat do número informado aberto e o código do Pix já colado na mensagem, bastando ao usuário apenas clicar para enviar.
- Imprimir: a opção de impressão tem como objetivo gerar um *ticket* com o *QR Code* do Pix para facilitar o pagamento do cliente em situações em que não é possível acessar o *QR*

⁶ <https://web.whatsapp.com/>

Figura 3.10 – Opção PixCriare na Forma de Pagamento



Fonte: Autor

Code diretamente da tela do computador. Uma pré visualização deste ticket é mostrada na Figura 3.12.

Nessa tela também há uma opção de cancelar o pagamento clicando no botão "Cancelar Pagamento", com isso o código do Pix é invalidado. O mesmo também ocorre automaticamente quando o tempo limite de 10 minutos é ultrapassado antes do usuário realizar o pagamento. A contagem do tempo é exibida na parte inferior da tela, em segundos.

Essa implementação também utiliza uma abordagem de *threads* semelhante à demonstrada na Seção 3.2, em que todo o processo de validação, comunicação com o servidor e aguardo da resposta de pagamento é realizado em uma *thread* anônima secundária, enquanto todas as alterações de tela, como a atualização da contagem e as interações do usuário, são sincronizadas e realizadas na *thread* principal. Dessa forma, o usuário pode interagir com a tela e escolher outra opção de pagamento ou até mesmo cancelar a operação, sem que a tela fique travada.

Ao realizar o pagamento com sucesso, registros são criados automaticamente nas tabelas necessárias do banco de dados, incluindo no registro de *logs* do sistema. A tela de pagamentos também é fechada automaticamente e o registro do novo pagamento é exibido no *grid* da tela de Contas a Receber, como mostrado na Figura 3.13.

Figura 3.11 – Tela de Pagamento PixCriare



Fonte: Autor

Ao clicar em "Excluir", o pagamento PixCriare é automaticamente devolvido por meio de uma devolução Pix, o que elimina a necessidade de o usuário realizar a devolução manualmente pelo aplicativo do banco, tornando a operação mais fácil e simplificada.

A implementação do PixCriare como uma nova forma de pagamento trouxe uma evolução para o sistema, oferecendo uma solução moderna e eficiente para seus clientes. Além disso, a integração do PIX com o sistema interno da empresa simplificou a gestão financeira, reduzindo custos e tempo gasto com transações bancárias, além de novos ganhos para empresa com uma pequena taxa que é recebida em cada utilização. A implementação do PixCriare foi concluída com sucesso, atendendo as necessidades dos usuários e certamente trará muitos benefícios para a empresa e seus clientes no futuro.

3.3.2 Relatório de Boletos Emitidos

Outro tipo de tarefa que se faz necessária no sistema é a criação de novos relatórios mediante solicitações dos clientes. A Figura 3.14 mostra o caso criado na plataforma MantisBT. Esta plataforma era utilizada pela empresa para gerenciar e detalhar o *backlog* dos projetos. Mantis BT (*Mantis Bug Tracker*) é uma plataforma de rastreamento de problemas e gerenci-

Figura 3.12 – Ticket de Impressão PixCriare



Fonte: Autor

amento de projetos de software de código aberto. Ela é projetada para ajudar as equipes de desenvolvimento a monitorar bugs, gerenciar solicitações de recursos e coordenar o trabalho em equipe (MANTIS BT, 2023).

Como é possível ver na descrição do caso, o cliente solicitou um relatório de boletos emitidos, especificando os campos e filtros necessários. Para atender a essa demanda, foi necessário criar um código SQL para realizar o *select* dos dados, considerando os filtros. A Figura 3.15 exibe a consulta implementada.

Em SQL, o caractere de dois pontos (:) é usado para indicar um parâmetro ou variável. Ele é usado em conjunto com uma cláusula como o "*WHERE*" ou "*SET*" para indicar que um valor será fornecido posteriormente. Por exemplo, na Figura 3.15 é possível verificar o código "*CR.DTPAGTO BETWEEN (:DATAI) AND (:DATAF)*" na linha 15, a expressão "(:DATAI)" é um parâmetro que tem seu valor atribuído posteriormente de acordo com a data inicial selecionada no filtro da tela. A Figura 3.16 mostra a atribuição destes parâmetros através do código do formulário Delphi implementado para este relatório. Neste mesmo formulário foi criada a tela com os filtros do relatório.

Figura 3.13 – Pagamento PixCriare Recebido

Contas a Receber (Individual) X

**Você está efetuando a baixa automática do documento selecionado.
Informe a forma de pagamento, data de recebimento e valor recebido.**

Dt. de recebimento	Vir. selecionado	Vir. abatimento	Desp. cartório	Juros	Desconto	Vir. a receber
21/06/2022	15,00	0,00	0,00	0,00	0,00	R\$ 15,00

Opções do recebimento

Forma de pagamento: Vir. recebido: R\$ 0,00

Histórico de Recebimento: Cheque Depósito Cartão

Forma de pagamento	Valor	Histórico
PixCriare	15,00	RECEBIMENTO DE CONTAS REF. AO TÍTULO [REDACTED] ID. PAGAM

Operações para o recebimento

F1 - Concluir F2 - Receber Valor de crédito para este cliente: R\$ 0,00

Fonte: Autor

Figura 3.14 – Detalhes do Caso no Mantis BT

Ver Detalhes do Caso

Ir para as Anotações Enviar um lembrete Histórico do Caso << >>

Núm	Projeto	Categoria	Visibilidade	Data de Envio	Última Atualização
0011058	Projetos Criare	[Todos os Projetos] Projeto SIA	público	2021-04-14 16:54	2021-04-15 17:34
Relator	Valquiria				
Atribuído a	Otávio				
Prioridade	normal	Gravidade	recurso novo	Freqüência	sempre
Estado	<input checked="" type="checkbox"/> fechado	Resolução	corrigido		
Versão do Produto					
Previsto para a Versão		Corrigido na Versão			
Resumo	0011058: Inclusão de Relatório de Boletos Emitidos nos relatórios padrões do Sia				
Descrição	Cliente solicitou relatório de boletos emitidos no mês, contendo juros, taxas e tarifas pagas. Filtrando por emissão, data pagamento ou data vencimento. Filtrando também por boletos recebidos e não recebidos.				

Fonte: Autor

Outro ponto importante neste tipo de implementação é a criação do layout do relatório em si, e isso é realizado com o auxílio do componente FastReport. O FastReport é um componente de relatórios para o Delphi, que permite criar e gerar relatórios de forma rápida e fácil

Figura 3.15 – *Select* do Relatório de Boletos Emitidos

```

1  SELECT CR.CODIGO_N, CL.NOMECLI, CR.EMISSAO, CR.VENCTO_PARCELA,
2  DATE_FORMAT(IFNULL(CR.DTPAGTO, '00/00/0000'), '%d/%m/%Y') AS DTPAGTO, CR.VLPAGTO,
3  CR.VALOR_PARCELA, CR.NOSSONUMERO, CR.ID_BOLETO, CR.FATURADO, CR.BAIXADO,
4  CR.SINTETICO, BC.TAXAJUROSAOMES, BC.PERCENTUALMULTA,
5  SUM(DRC.VALORTAXACOBANCA) AS VALORTAXACOBANCA, DRC.VALORJUROPAGO, DRC.VALORMULTAPAGA
6  FROM creceber CR
7  INNER JOIN clientes CL ON (CR.CODIGO_N=CL.CODIGOCLI)
8  INNER JOIN bancos BC ON CR.ID_BANCO = BC.CODIGO
9  LEFT JOIN detalhe_retorno_cobranca DRC ON CR.CONTADOR = DRC.ID_CRECEBER
10 WHERE CR.DC = 'D'
11 AND IF(:situacao = 0, 1 = 1,
12 IF(:situacao = 1, CR.BAIXADO = 'S', CR.BAIXADO = 'N'))
13 AND IF(:TipoData = 0, CR.EMISSAO BETWEEN (:DATAI) AND (:DATAF),
14 IF(:TipoData = 1, CR.VENCTO_PARCELA BETWEEN (:DATAI) AND (:DATAF),
15 CR.DTPAGTO BETWEEN (:DATAI) AND (:DATAF)))
16 AND CR.FATURADO = 'N'
17 AND COALESCE(CR.NOSSONUMERO, '') <> ''
18 AND CR.EMPRESA IN (:empresa)
19 GROUP BY CR.CONTADOR
20 ORDER BY CL.NOMECLI
21

```

Fonte: Autor

em diferentes formatos, incluindo PDF, Excel, HTML e muitos outros. Ele oferece uma interface gráfica intuitiva e recursos avançados, como criação de sub-relatórios, gráficos e códigos personalizados (FAST REPORTS, 2023).

A Figura 3.17 exibe a interface do FastReport que foi utilizada para criar o novo relatório de boletos emitidos. Observe que nessa figura há um componente chamado "frxDBDataset1", que recebe os dados do *dataset* "cl_reports" mostrado na Figura 3.16. Essa integração possibilita o acesso a todos os campos que foram carregados no *select*, como a terceira coluna que recebe os valores de "[frxDBDataset1."NOMECLI"]", que representa o nome do cliente. Além disso, nesta interface foram ajustados os alinhamentos das colunas, o cabeçalho e o rodapé, bem como realizados os cálculos dos totais.

Por fim, a Figura 3.18 mostra o formulário implementado já integrado ao sistema SIA, com os filtros de data de entrada e saída. O filtro de período permite indicar se as datas são referentes ao período de vencimento, pagamento ou emissão do boleto, além do filtro de situação. Ao lado está a saída do relatório gerado com os filtros indicados.

Figura 3.16 – Select do Relatório de Boletos Emitidos

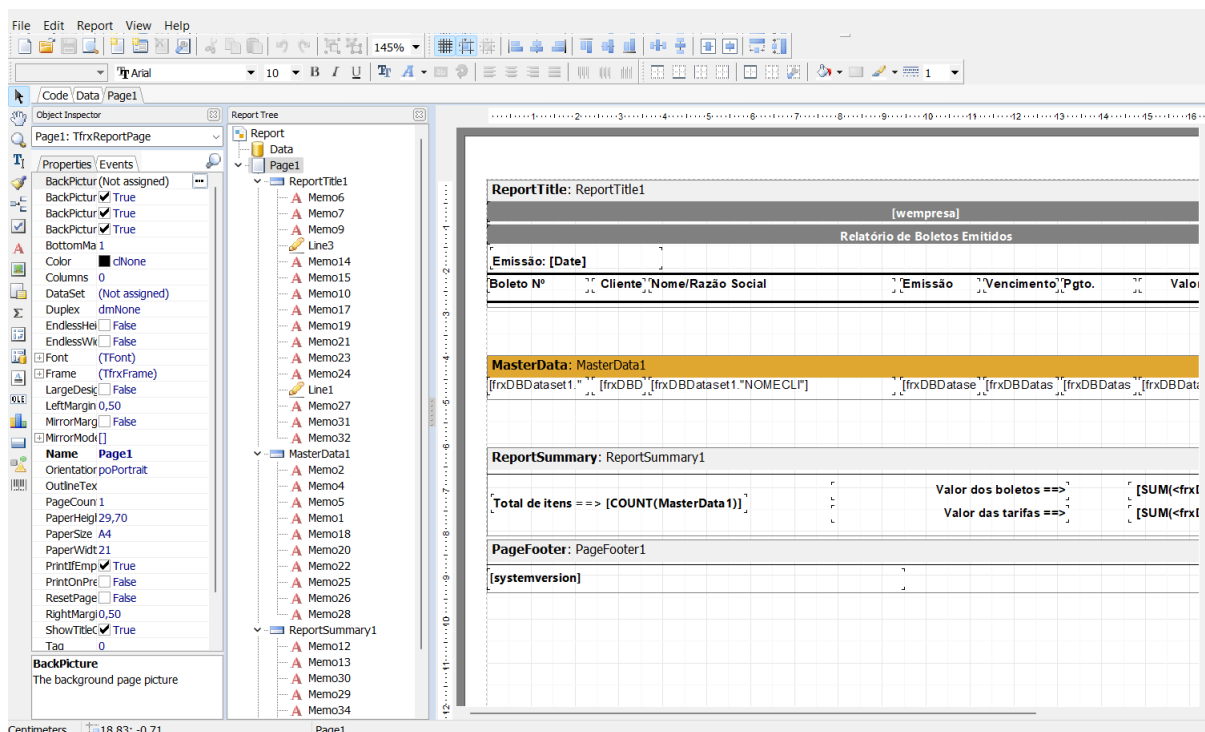
```

1 // Fecha o dataset
2 cl_reports.Close;
3
4 // wsq é a string do select montado
5 cl_reports.CommandText := wsq;
6
7 // Parametros / Filtros (data inicial, data final, tipo de periodo, situacao)
8 cl_reports.Params.ParamByName('DATAI').AsDate := dtInicial.Date;
9 cl_reports.Params.ParamByName('DATAF').AsDate := dtFinal.Date;
10 cl_reports.Params.ParamByName('TipoData').AsInteger := cboTipoPeriodo.ItemIndex;
11 cl_reports.Params.ParamByName('situacao').AsInteger := rgSituacao.ItemIndex;
12 cl_reports.Params.ParamByName('empresa').AsString := cboEmpresa.Text;
13
14 // Abre o dataset
15 cl_reports.Open;

```

Fonte: Autor

Figura 3.17 – Componente FastReport para Criação de Relatórios



Fonte: Autor

A implementação do novo relatório atendeu às demandas do cliente em relação aos campos e filtros necessários, proporcionando uma visualização mais clara e objetiva dos dados requeridos.

Figura 3.18 – Relatório de Boletos Emitidos no Sistema

Boleto Nº	Cod. Cliente	Nome/Razão Social	Emissão	Data Venc.	Data pgto.	Valor	Juros	Multa	Tarifa
09638932	15102	A. C. BA	05/05/2021	03/06/2021		1.853,90	0,00	0,00	0,00
09638925	15102	A. C. BA	05/05/2021	03/06/2021		3.011,30	0,00	0,00	0,00
09638918	15102	A. C. BA	05/05/2021	03/06/2021		1.846,54	0,00	0,00	0,00
09638781	8558	A. C. BA	05/05/2021	03/06/2021		2.557,90	0,00	0,00	0,00
09638799	6777	A. C. BA	05/05/2021	03/06/2021		1.727,15	0,00	0,00	0,00
09643140	6777	A. C. BA	11/05/2021	08/06/2021		3.848,40	0,00	0,00	0,00
09636788	12852	ADAILM	05/05/2021	02/06/2021		201,41	0,00	0,00	0,00
09635990	8270	ADELIA	05/05/2021	02/06/2021		397,00	0,00	0,00	0,00
09650209	10602	ADELM	18/05/2021	01/06/2021		332,50	0,00	0,00	0,00
09650223	10602	ADELM	18/05/2021	15/06/2021		331,00	0,00	0,00	0,00
09633782	10602	ADELM	04/05/2021	01/06/2021		346,00	0,00	0,00	0,00
09650216	10602	ADELM	18/05/2021	08/06/2021		331,00	0,00	0,00	0,00
09653092	464	ADEMA	19/05/2021	09/06/2021		743,20	0,00	0,00	0,00
09645223	1579	ADEMP	12/05/2021	09/06/2021		231,00	0,00	0,00	0,00
09645216	1579	ADEMP	12/05/2021	02/06/2021		231,47	0,00	0,00	0,00
09636638	1579	ADEMP	05/05/2021	02/06/2021		450,00	0,00	0,00	0,00
09643821	15415	ADEMP	12/05/2021	02/06/2021		1.395,00	0,00	0,00	0,00

Fonte: Autor

3.4 Importador Padaria Lisboa

Assim como os primeiros exportadores mencionados neste relatório, este projeto também foi desenvolvido de forma independentemente. Ele foi solicitado pelo cliente Padaria Lisboa, que necessitava importar dados de um sistema externo para a base de dados da empresa e utilizá-los em seus sistemas internos da Criare.

Para isso foi criado um documento de layout próprio que seria utilizado na implementação do sistema, o importador recebe um arquivo '.txt' e automaticamente converte este arquivo inserindo os dados no banco realizando todas as validações necessárias. O layout desenvolvido segue a seguinte estrutura:

Esse sistema apresenta uma interface simples para o usuário selecionar a empresa, o arquivo de texto, um botão para realizar a importação como mostrado na Figura 3.20. A Figura 3.21 mostra que existe também uma segunda aba onde apresenta um *grid* que lista todos os dados importados, dessa forma é possível que o usuário realize a validação antes de aplicar as alterações, pois a aplicação além de inserir novos registros, também atualiza registros caso já existam na base de dados.

Figura 3.19 – Exemplo Layout Importação de Clientes

Layout p/ importação de Clientes:

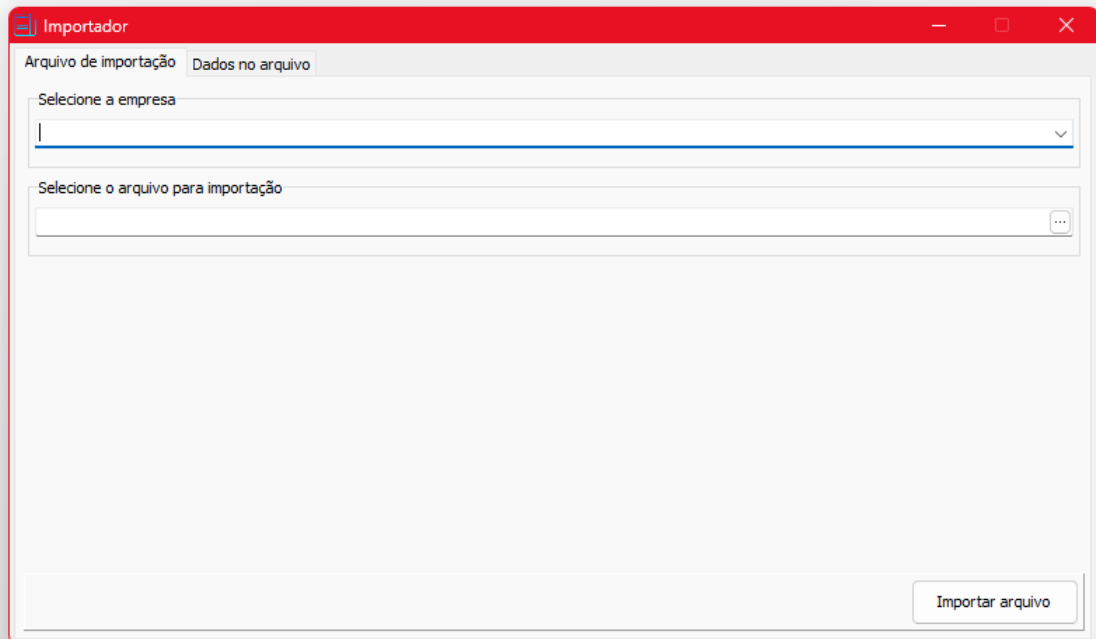
CAMPO	TAM	INICIO	FIM	TIPO	OBSERVACOES
CODIGO	7	1	7	NUMERICO	SEM FORMATAÇÃO(./-) ALINHADOS A DIREITA, COM ZEROS A ESQUERDA
CPF/CNPJ	14	8	21	NUMERICO	SEM FORMATAÇÃO(./-) ALINHADOS A DIREITA, COM ZEROS A ESQUERDA
INSCRICAO/IDENT.	14	22	35	NUMERICO	SEM FORMATAÇÃO(./-) ALINHADOS A DIREITA, COM ZEROS A ESQUERDA
NOME/RAZAO SOCIAL	35	36	70	ALFANUMERICO	ALINHADOS A ESQUERDA COM ESPACOS A DIREITA
ENDERECO	35	71	105	ALFANUMERICO	ALINHADOS A ESQUERDA COM ESPACOS A DIREITA
BAIRRO	20	106	125	ALFANUMERICO	ALINHADOS A ESQUERDA COM ESPACOS A DIREITA
CEP	8	126	133	NUMERICO	SEM FORMATAÇÃO(./-) ALINHADOS A DIREITA, COM ZEROS A ESQUERDA
CIDADE	20	134	153	ALFANUMERICO	ALINHADOS A ESQUERDA COM ESPACOS A DIREITA
ESTADO	2	154	155	ALFANUMERICO	ALINHADOS A ESQUERDA COM ESPACOS A DIREITA
BRANCOS	2	156	157	ESPACOS	DEIXAR EM BRANCO (ESPAÇOS)
TELEFONE	10	158	167	NUMERICO	SEM FORMATAÇÃO(./-) ALINHADOS A DIREITA, COM ZEROS A ESQUERDA
BRANCOS	30	168	197	ALFANUMERICO	ESPAÇOS
BLOQUEADO (*)	1	198	198	ALFANUMERICO	S/N - SIM OU NÃO
SALDO LIMITE (**)	10	199	208	DECIMAL	VALOR ALINHADO A ESQUERDA COM DUAS CASAS DECIMAIS S/SEPARACAO
VALOR LIMITE (***)	10	209	218	DECIMAL	VALOR ALINHADO A ESQUERDA COM DUAS CASAS DECIMAIS S/SEPARACAO
ID CIDADE	8	219	226	NUMERICO	CODIGO DA CIDADE CONFORME TABELA IBGE

OBS:

- (*) - CAMPO "BLOQUEADO":
Informa ao Programa Sia Frente de Caixa que este Cliente está bloqueado independente da situação das contas (ou seja, se não tiver conta em atraso ou cheque Devolvido).
- (**) -CAMPO "SALDO LIMITE":
É o valor do limite que o cliente já atingiu
- (***)-CAMPO "VALOR DO LIMITE":
É o valor do limite liberado para o Cliente.
(O Programa trabalha com o valor do limite que ainda resta, se atingido este limite o cliente será bloqueado no Frente de Caixa).

Fonte: Autor

Figura 3.20 – Interface Principal do Importador



Fonte: Autor

Figura 3.21 – Segunda Aba do Importador

Cód. empres	Cód. produt	Cód. barra	Desc. abreviada	Unitário	Custo	Quantidade	Uni	Promoção	Início promo	Validade pr
1	8222	0000000008222	EMBALAGEM P17 G17 C10	0,1	0,01	0	UN	0		
1	8221	7898080640222	CREME DE LEITE ITALAC 20	45	0,01	0	UN	0		
1	8220	0000000008220	BISCOITO SALGADINHOS QUE	449	0,01	0	KG	0		
1	8219	0000000008219	BISCOITO SALGADINHOS BAC	449	0,01	0	KG	0		
1	8218	0000000008218	BISCOITO SALGADINHOS PIM	0,1	0,01	0	KG	0		
1	8217	0000000008217	BISCOITO AMANTEIGADOS IN	619	0,01	0	KG	0		
1	8216	0000000008216	BISCOITO AMANTEIGADOS A'	619	0,01	0	KG	0		
1	8215	0000000008215	MILKA WHOLE HAZELNUTS 27	0,1	0,01	0	UN	0		
1	8214	7622210702623	MILKA TRIOLADE 280G	0,1	0,01	0	UN	0		
1	8213	7622210999658	MILKA OREO SANDWICH 92G	0,1	0,01	0	UN	0		
1	8212	7622210999368	MILKA STRAWBERRY 100G	0,1	0,01	0	UN	0		
1	8211	7622210194046	MILKA OREO 100G	0,1	0,01	0	UN	0		
1	8210	7622210690845	MILKA NOISETTE 100G	0,1	0,01	0	UN	0		
1	8209	7622210726773	MILKA HAZELNUT 100G	0,1	0,01	0	UN	0		
1	8208	7622300200237	MILKA ALPINE MILK 100G	0,1	0,01	0	UN	0		
1	8207	0000000008207	BAI A DE COCO BRIGADEIRO	400	40	0	KG	0		

Fonte: Autor

3.5 SIA Web

Este último projeto que foi desenvolvido consiste em uma versão web das principais funcionalidades do SIA. Trata-se do primeiro projeto web da empresa, e sua implementação era necessária para trazer melhorias e facilidades para os clientes.

A construção completa do *front-end* foi feita em React JS juntamente com Typescript, e o *back-end* da aplicação foi desenvolvido em Node JS. O sistema necessitava apresentar toda uma estrutura de autenticação, cadastros de vários módulos, diversos tipos de movimentações e transações, além de diversos relatórios.

Durante o desenvolvimento do projeto, foi implementada uma API (*Application Programming Interface*)⁷ em Node.js no *back-end*, seguindo a arquitetura MVC (*Model-View-Controller*). Para isso, foram definidas as rotas de CRUD (*Create, Read, Update, Delete*) da aplicação, permitindo assim o gerenciamento dos dados armazenados no banco de dados. No projeto, o banco de dados foi implementado utilizando o sistema de gerenciamento de banco de

⁷ API é uma interface que permite que diferentes sistemas se comuniquem e compartilhem informações de forma padronizada e segura.

dados MySQL, que oferece uma estrutura robusta e confiável para armazenamento e recuperação de dados.

Uma abordagem adotada para garantir a segurança nas rotas foi a utilização do JWT (*JSON Web Token*) para autenticação. O JWT é um padrão aberto para a criação de *tokens* de acesso que são usados para transmitir informações seguras entre as partes de uma forma compacta e verificável (JWT.IO, 2023). Na aplicação, quando um usuário realiza o login, é gerado um *token* JWT contendo as informações do usuário autenticado, que é enviado para o cliente e armazenado. A partir desse momento, em cada requisição, o cliente deve enviar o *token* JWT no cabeçalho da requisição para acessar as rotas protegidas, que verificam se o *token* é válido e se o usuário tem permissão para acessar a rota solicitada.

Outro controle de acesso que foi implementado na aplicação consiste na atribuição de diferentes níveis de acesso aos usuários, sendo eles:

- *Comum*: para usuários básicos da empresa, que possuem acesso limitado a informações sensíveis da empresa e a controle de outros usuários;
- *Master*: o mais alto nível de acesso, geralmente atribuído a diretores e gerentes, que têm acesso a todas as informações da empresa e possuem o poder de criar e gerenciar outros usuários da própria empresa;
- *Admin*: este nível de acesso é atribuído a desenvolvedores e gerentes da Criare, permitindo o gerenciamento de todas as empresas dos clientes e usuários que utilizam o sistema, incluindo bloqueio de acessos e edição de informações.

O *front-end* da aplicação foi completamente desenvolvido utilizando a tecnologia React em conjunto com TypeScript. O desenvolvimento em React permitiu a criação de uma interface de usuário (UI) moderna e responsiva, utilizando componentes reutilizáveis e otimizados para performance. Além disso, a utilização do TypeScript trouxe mais segurança e confiabilidade ao código, reduzindo erros e garantindo maior manutenibilidade no longo prazo.

A integração com a API Node.js foi realizada utilizando o protocolo HTTP (*Hypertext Transfer Protocol*) para comunicação entre o *front-end* e o *back-end* da aplicação. As rotas foram definidas na API para as operações de CRUD, e o React consumiu essas rotas através

de requisições utilizando a biblioteca Axios ⁸. As respostas das requisições são retornadas em formato JSON ⁹, que é facilmente manipulável em JavaScript.

No React, foram criados serviços para centralizar as requisições à API. A Figura 3.22 mostra os métodos *getByID* e *create* do serviço de "Grupos" do sistema, nos quais são utilizados os métodos *get* e *post* para chamar as rotas correspondentes da API. Adicionalmente, as interfaces do TypeScript são criadas e usadas para garantir o tipo correto de dados nas requisições.

No exemplo apresentado, as funções estão marcadas com *async*, indicando que elas fazem chamadas assíncronas à API e retornam promessas. O uso do *await* nas chamadas indica que o código deve esperar pela resposta da API antes de continuar a execução.

A página inicial do sistema, apresentada parcialmente na Figura 3.23, exibe diversas informações relevantes para a análise de desempenho da empresa, tais como o resumo diário das contas a receber e a pagar, gráficos que relacionam compras e vendas mensais, além de um ranking dos produtos mais vendidos e com maior estoque, entre outras informações. Tudo isso é essencial para o acompanhamento dos negócios da empresa de forma eficiente e prática.

Cada módulo de cadastro e movimentação conta com uma tela de listagem correspondente, como ilustrado na Figura 3.24. Nessas telas, é apresentado um *grid* com todos os registros, permitindo filtros de busca quando necessário, bem como botões para edição ou remoção de registros existentes. Além disso, há um botão para inserção de um novo registro, facilitando o processo de inclusão de novas informações.

Cada módulo do sistema também possui uma tela de detalhes, como exemplificado na Figura 3.25. Essas telas são compostas por formulários com campos de diversos tipos, utilizados para inserir ou editar os dados de um registro específico.

Na Figura 3.26 é mostrado um exemplo de uma tela de filtros para gerar um relatório. Além disso é possível verificar a variedade de relatórios disponíveis no menu lateral, dentro do grupo 'Relatórios'.

Após o usuário selecionar os filtros desejados e clicar em "Gerar Relatório", é realizada uma requisição para a API, que busca os dados, com os dados retornados é utilizada a biblioteca React-pdf para montar os relatórios de acordo com o layout programado. O relatório é gerado

⁸ <https://axios-http.com/ptbr/docs/intro>

⁹ JSON é uma linguagem de formato de dados que utiliza uma sintaxe simples para armazenar e transmitir informações em um formato legível por humanos e por máquinas. É amplamente utilizado em aplicações web e móveis para a transferência de dados entre sistemas.

Figura 3.22 – Requisições do React para a API

```
1  const getById = async (id: string): Promise<IGroupDetails | Error> => {
2    try {
3      const { data } = await Api.get(`/grupo/${id}`);
4
5      if (data) return data[0]; // vem como um array de 1 posicao pelo padrao da API
6
7      return new Error("Erro ao consultar o registro");
8    } catch (error) {
9      console.error(error);
10     return new Error(
11       (error as { message: string }).message || "Erro ao consultar o registro",
12     );
13   }
14 };
15
16 export interface ICreateGroupProps {
17   DESCRICAO: string;
18 }
19 const create = async (dados: ICreateGroupProps) => {
20   try {
21     const { data } = await Api.post("/grupo/empresa", dados);
22
23     if (!data.error)
24       return (data as { message: { ID: string; DESCRICAO: string } }).message;
25
26     return new Error(
27       (data as { message: string }).message || "Erro ao criar o registro",
28     );
29   } catch (error) {
30     console.error(error);
31     return new Error(
32       (error as { message: string }).message || "Erro ao criar o registro",
33     );
34   }
35 };
```

Fonte: Autor

Figura 3.23 – Dashboard



Fonte: Autor

Figura 3.24 – Telas de Listagem

The screenshot shows the "Clientes e Fornecedores" list page. The page includes a search bar, a "Novo +" button, and a table with columns for "Ações", "Cód. Emiten...", "Nome Emitente", "Email", and "Telefone".

Ações	Cód. Emiten...	Nome Emitente	Email	Telefone
	8.000	Kalina Beevis	kbeevisr@netscape.com	(31) 9565-6175
	7.999	Daune Carress	dcarressrq@cbc.ca	640-407-3028
	7.998	Emmerich Lalor	elalorpp@4shared.com	140-950-0612
	7.997	Faustine Shldrake	fshldrakero@upenn.edu	515-191-0988
	7.996	Cly Eric	cericrn@google.pl	228-934-1690
	7.995	Billie Dafydd	bdafyddrm@list-manage.com	128-706-6676

Fonte: Autor

em PDF e é aberto diretamente no navegador do usuário. O resultado é algo semelhante ao demonstrado na Figura 3.27.

Devido ao controle dinâmico dos menus, é possível exibir ou ocultar menus dependendo das permissões do usuário que está acessando o sistema. Por exemplo, usuários comuns terão um conjunto diferente de menus em comparação aos usuários *admin*, como pode ser visto na Figura 3.28 que apresenta apenas os menus "Controle de Usuários" e "Controle de Empresas".

Figura 3.25 – Formulários

The screenshot displays the 'Nova venda' form in the SIA Web system. The interface is divided into several sections:

- Header:** 'Nova venda:'
- Geral:** Contains dropdown menus for 'Operação' (Selezione uma operação), 'Emitente' (8000 - Kalina Beevis), and 'Vendedor' (2 - Otávio Melo). A date field 'Data de Saída/Entrada' is set to 12/02/2023.
- Totais:** A summary section with fields for 'Número de Itens' (0), 'Desconto Total' (R\$ 0), 'Acréscimos/Despesas' (R\$ 0), 'Total Itens' (R\$ 0), 'Total ICMS ST', 'Total FCP ST', 'Total IPI', and 'Total do Documento'.
- Items da Movimentação:** A section for adding items with fields for 'Produto' (Selezione um item), 'Quantidade' (10), and 'Valor Unitário' (R\$ 0).
- Navigation:** A dark sidebar on the left provides access to 'Início', 'Empresa', 'Cadastro', 'Movimentações' (with sub-items like 'Vendas', 'Entrada Devolução', etc.), 'Transações', and 'Relatórios'.

Fonte: Autor

Figura 3.26 – Exemplo de Filtros dos Relatórios do SIA Web

The screenshot displays the 'Relatório entrada de mercadorias' form in the SIA Web system. The interface includes the following elements:

- Header:** 'Relatório entrada de mercadorias'
- Filtros:** A section for filtering reports with dropdown menus for 'Emitente', 'Produto', 'Grupo', 'Familia', and 'Categoria'.
- Período de Emissão:** Date selection fields for 'Data Inicial' (21/02/2023) and 'Data Final' (21/02/2023).
- Action:** A prominent orange button labeled 'GERAR RELATÓRIO'.
- Navigation:** A dark sidebar on the left shows the 'Relatórios' menu with various sub-options like 'Produtos', 'Emitentes', 'Vendedores', etc., with 'Entrada Mercadorias' selected.

Fonte: Autor

Neste projeto, também foi adotada a tecnologia de comunicação WebSocket (WS)¹⁰. Com essa tecnologia, foi possível implementar um controle de usuários, facilitando o envio de notificações instantâneas, sem a necessidade de atualização da página pelo usuário.

Um exemplo deste uso é visto na Figura 3.28, em que é possível identificar quantos usuários estão conectados e quais deles que estão online no momento, isso é indicado pela coluna "Estado". Graças à integração do WS, também foi possível implementar uma garantia de login único para os usuários do sistema. Com essa funcionalidade, caso um usuário esteja

¹⁰ WebSocket é uma tecnologia que permite a comunicação bidirecional entre um navegador da web (cliente) e um servidor da web em tempo real, sem a necessidade de constantes requisições do cliente.

Figura 3.27 – Exemplo de Relatório Gerado

#Código	Cod. Barras	Descrição	QTD Estoque	QTD Reservado	Preço de custo	Margem (%)	Preço de venda	NCM	CEST
1	0070330129627	CANETA BIC CRISTAL AZUL	101	6	R\$ 0,680	150,00	R\$ 1,70	3926.10.00	19.0040.0
2	7896342449514	CORRETIVO	-4	13	R\$ 0,960	200,00	R\$ 2,88	3926.10.00	19.0040.0
3	7897424080762	PINCEL ATOMICO PRETO REF/70414 PILOT	13	0	R\$ 3,030	149,83	R\$ 7,57	3926.10.00	19.0040.0
4	7897424081387	PINCEL QUADRO BRANCO WBM 7 AZUL PILOT	14	0	R\$ 7,200	120,00	R\$ 15,84	3926.10.00	19.0040.0
5	7896066130125	LABRA COLORIDOS NATUREZA 12 CORES	5	0	R\$ 2,630	165,02	R\$ 6,97	3926.10.00	19.0040.0
6	7897424082124	PILOT LUMI COLOR 200-SL MARCA TEXTO AMARELO 1 UNIDADE	-7	0	R\$ 2,190	130,14	R\$ 5,04	3926.10.00	19.0040.0
7	7897424080786	CANETA PILOT ATOMICO 1100-P VERMELHO	2	0	R\$ 3,030	120,13	R\$ 6,67	3926.10.00	19.0040.0
8	7891360577485	CANETINHA FABER-CASTELL 12 CORES	8	0	R\$ 10,830	135,00	R\$ 25,45	3926.10.00	19.0040.0
9	0070330129634	CANETA BIC CRISTAL VERMELHA C/50	66	0	R\$ 0,690	149,28	R\$ 1,72	3926.10.00	19.0040.0
10	0070330132665	CANETA BIC COM 4 CORES: AZUL PRETA VERMELHO VERDE	13	0	R\$ 0,790	150,63	R\$ 1,98	3926.10.00	19.0040.0
11	7896572010065	HRIDROCOR C/12 COORES NED PAN	10	0	R\$ 9,960	119,98	R\$ 21,91	3926.10.00	19.0040.0
12	7896572000370	CANETA ESFER. 0,7MM AZUL	27	0	R\$ 0,683	165,01	R\$ 1,81	3926.10.00	19.0040.0
13	7896572011291	CANETA HIDROGR NEOPEN GIGANTE 6COR 5X1	10	0	R\$ 9,880	150,00	R\$ 24,70	3926.10.00	19.0040.0
14	7897424081943	CANETA CD PRETA 2 0 PILOT	30	0	R\$ 4,620	164,94	R\$ 12,24	3926.10.00	19.0040.0
15	7897424080854	PILOT RETRO PROJETER 2 0 MARCADOR PERMANENTE AZUL 1 UNIDADE	20	0	R\$ 3,490	165,04	R\$ 9,25	3926.10.00	19.0040.0
16	7896644715768	PASTA ABA ELASTICO SOFT FUME UNIDADE POLIBRAS	36	0	R\$ 0,960	250,00	R\$ 3,36	3926.10.00	19.0040.0
18	0070330318731	ME CANETA BIC METAL POINT AZUL	13	0	R\$ 7,200	250,00	R\$ 25,20	3926.10.00	19.0040.0
20	7898324237409	Lampada(teste cod 1)	1	0	R\$ 2,222	25,11	R\$ 2,78	3926.10.00	19.0040.0
21	7891022100143	Sapãoáceo cremoso Bom bril	16	0	R\$ 7,990	35,04	R\$ 10,79	3405.40.00	28.0630.00
22	7896094906815	desodorante	30	0	R\$ 4,090	44,99	R\$ 5,93	0402.99.02	17.0210.0
23		- algo sem ean	3	0	R\$ 5,650	44,96	R\$ 8,19	0402.99.00	17.0210.0

Fonte: Autor

Figura 3.28 – Controle de Usuários (Admin)

Ações	Status	Nome	Master	Admin	Empresa	Estado
	Ativo	Desenvolvimento - Criare TI	✓	✓	Ederson de Souza Selvati & Cia Ltda	Online
	Ativo	Conta Administrativa - Criare TI	✓	✓	Ederson de Souza Selvati & Cia Ltda	Offline
	Ativo	Tester - Criare TI	✓	×	EDERSON DE SOUZA SELVATI & CIA LTDA	Offline
	Ativo	Novo dasdasd	×	×	EDERSON DE SOUZA SELVATI & CIA LTDA	Offline
	Ativo	teste	✓	×	teste	Offline

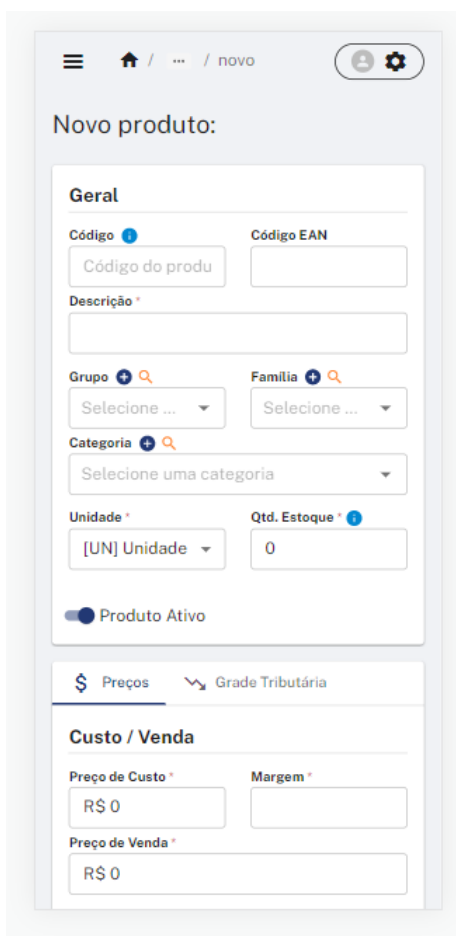
Fonte: Autor

logado em uma máquina e tente fazer login em outra, ele será automaticamente bloqueado, evitando possíveis problemas de segurança ou perda de dados.

A aplicação conta com outras funcionalidades importantes, como o cadastro automático de novos usuários, que inclui o envio de e-mails automáticos para validação da conta. Além disso, há também a opção de recuperação de senha, que também utiliza a validação via e-mail automático com *tokens* temporários para garantir a segurança do acesso às contas dos usuários.

Além disso, a aplicação foi projetada para ser completamente responsiva, permitindo que os usuários acessem e usem a plataforma em seus dispositivos móveis, como pode ser visto na Figura 3.29. Também, para aprimorar ainda mais a experiência do usuário, foram adicionados componentes de mensagem em diálogo e de alerta personalizados, que podem ser visualizados nas Figuras 3.30 e 3.31, respectivamente.

Figura 3.29 – Aplicação em Celulares



Novo produto:

Geral

Código ⓘ Código EAN

Código do produ

Descrição *

Grupo ⓘ Família ⓘ

Selecione ... Seleccione ...

Categoria ⓘ

Selecione uma categoria

Unidade * Qtd. Estoque * ⓘ

[UN] Unidade 0

Produto Ativo

Preços Grade Tributária

Custo / Venda

Preço de Custo * Margem *

R\$ 0

Preço de Venda *

R\$ 0

Fonte: Autor

Figura 3.30 – Mensagens em Diálogo



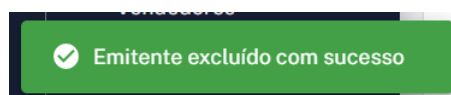
⚠ Excluir Emitente ✕

Deseja realmente excluir esse emitente?

Não Excluir

Fonte: Autor

Figura 3.31 – Mensagens de Alerta



Fonte: Autor

Por fim, o projeto SIA Web, um sistema completo, foi desenvolvido com sucesso, e apesar de não ter sido testado e disponibilizado no ambiente de produção antes do fim do período de estágio, o projeto demonstrou grande potencial e capacidade de atender às necessidades dos clientes da empresa.

4 CONCLUSÃO

Durante o período na empresa, foram desenvolvidos vários projetos, e que foram implementados com sucesso. Com exceção do projeto SIA Web, todos os projetos foram testados e disponibilizados no ambiente de produção, e já estavam em uso pelos clientes finais, funcionando corretamente e atendendo as expectativas.

Com o desenvolvimento dos projetos foi possível obter uma valiosa experiência profissional, principalmente em relação ao entendimento do ambiente de trabalho e das demandas dos clientes. Foi necessário lidar com mudanças constantes no escopo das atividades e encontrar soluções para os desafios enfrentados no desenvolvimento das aplicações. Essa experiência permitiu uma maior compreensão da importância da comunicação efetiva e do trabalho em equipe na resolução de problemas e na entrega de resultados de qualidade.

No início, foi observado que a empresa apresentava certa falta de organização em relação à documentação. A ausência de informações claras e padronizadas em relação à identidade visual, à padronização de código e aos módulos dos sistemas oferecidos dificultou a adaptação ao ambiente de trabalho e ao entendimento dos sistemas existentes. Por essa questão, muitas vezes foram implementando recursos que já haviam sido criados anteriormente, causando uma replicação desnecessária de código, por exemplo.

É notável a relevância dos conhecimentos adquiridos durante o curso de Engenharia de Controle e Automação na UFLA. Ao longo dos períodos, as atividades propostas pelos professores foram essenciais para o desenvolvimento das habilidades de trabalho em equipe, aprimorando a capacidade de colaboração e cooperação, e estimulando a busca constante por soluções para os desafios, seja de forma individual ou coletiva.

É importante ressaltar as disciplinas do curso de graduação que tiveram impacto direto na realização e implementação dos projetos, em especial Algoritmos e Estrutura de Dados I e II e Programação Orientada a Objetos. Essas disciplinas juntamente com o período de trabalho na empresa Júnior Emakers Jr. da UFLA, proporcionaram um sólido embasamento teórico e prático para a compreensão dos conceitos e técnicas utilizados no desenvolvimento de software, contribuindo para a realização das atividades propostas com eficiência e qualidade.

Considerando que a automação é uma área ampla e diversificada, com aplicações em vários setores além da indústria, seria benéfico para os estudantes de engenharia de controle e automação terem a oportunidade de aprender mais sobre metodologias ágeis de trabalho e

organização, como o Scrum, ao longo de sua formação. Essa habilidade pode ser aplicada não apenas em projetos durante a graduação, mas também em suas carreiras profissionais futuras.

REFERÊNCIAS

Alcides Junqueira. **O que é automação comercial**. 2020. <<https://www.infovarejo.com.br/o-que-e-automacao-comercial/00>>. Acessado em 07 de Janeiro de 2023.

ATLASSIAN. **Jira Core dashboard: your project status at a glance**. 2023. Acessado em 05 de Janeiro de 2023. Disponível em: <<https://www.atlassian.com/blog/jira-core/jira-core-dashboard-project-status-glance>>.

ATLASSIAN. **What is a kanban board?** 2023. Acessado em 05 de Janeiro de 2023. Disponível em: <<https://www.atlassian.com/agile/kanban/boards>>.

ATLASSIAN. **What is Jira Software?** 2023. Acessado em 05 de Janeiro de 2023. Disponível em: <<https://www.atlassian.com/software/jira/guides/getting-started/introduction#what-is-jira-software>>.

BUSCHMANN, F.; HENNEY, K.; SCHMIDT, D. C. **Pattern-Oriented Software Architecture, Volume 5: On Patterns and Pattern Languages**. [S.l.]: John Wiley Sons, 2007.

DEITEL, H. M.; DEITEL, P. J. **Java: How to Program**. [S.l.]: Pearson Education, 2011.

DEV MEDIA. **Introdução ao Padrão MVC**. 2013. Acessado em 08 de Janeiro de 2023. Disponível em: <<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>>.

ELMASRI, R.; NAVATH, S. B. **Sistemas de banco de dados - 7ª edição**. [S.l.]: Pearson Education do Brasil, 2019. ISBN 9788543025001.

FAST REPORTS. **FastReport VCL**. 2023. Acessado em 11 de Janeiro de 2023. Disponível em: <<https://www.fast-report.com/pt/product/fast-report-vcl/>>.

FREEMAN, A.; ROBSON, E. **Learning Node.js: A Hands-On Guide to Building Web Applications in JavaScript**. 2nd. ed. [S.l.]: Addison-Wesley Professional, 2016. ISBN 9780134663409.

JWT.IO. **Introduction to JSON Web Tokens**. 2023. Acessado em 10 de Janeiro de 2023. Disponível em: <<https://jwt.io/introduction>>.

MANTIS BT. **MantisBT makes collaboration with team members clients easy, fast, and professional**. 2023. Acessado em 11 de Janeiro de 2023. Disponível em: <<https://www.mantisbt.org/>>.

Mozilla Developer Network. **JavaScript**. 2023. <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Acessado em 09 de Janeiro de 2023.

NODE.JS FOUNDATION. **About Node.js**. 2023. Acessado em 09 de Janeiro de 2023. Disponível em: <<https://nodejs.org/en/about/>>.

ORACLE. **What is ERP?** 2023. Acessado em 10 de Janeiro de 2023. Disponível em: <<https://www.oracle.com/erp/what-is-erp/>>.

REACTJS. **Introduzindo JSX**. 2023. Disponível em: <<https://pt-br.reactjs.org/docs/introducing-jsx.html>>.

REACTJS. **React - Uma biblioteca JavaScript para criar interfaces de usuário.** 2023. Disponível em: <<https://pt-br.reactjs.org/>>.

SCHWABER, K.; SUTHERLAND, J. **Scrum - A arte de fazer o dobro do trabalho na metade do tempo.** [S.l.]: Alta Books, 2014.

SCRUM.ORG. **WHAT IS SCRUM?** 2023. Acessado em 15 de Janeiro de 2023. Disponível em: <<https://www.scrum.org/resources/what-is-scrum>>.

SMITH, J. The power and flexibility of object pascal in delphi. **Journal of Software Development**, Springer, v. 12, n. 3, p. 123–145, 2021.

SOMMERVILLE, I. **Engenharia de software.** 10. ed. [S.l.]: Pearson, 2019.

STATISTA. **Number of mobile internet users in Brazil from 2018 to 2027.** 2023. Acessado em 02 de Janeiro de 2023. Disponível em: <<https://www.statista.com/statistics/259749/mobile-phone-internet-users-in-brazil/>>.

TYPESCRIPT. **TypeScript is JavaScript with syntax for types.** 2023. Disponível em: <<https://www.typescriptlang.org/>>.