

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

GUSTAVO KEN HONDA

MICHEL CHIEREGATO GRIETSCHICHKIN

PEDRO LUI NIGRO CHAZANAS

**Sistema de gerenciamento de vendas e controle de estoque usando
Inteligência Artificial**

SÃO PAULO

2018

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

GUSTAVO KEN HONDA

MICHEL CHIEREGATO GRIETSCHICHKIN

PEDRO LUI NIGRO CHAZANAS

**Sistema de gerenciamento de vendas e controle de estoque usando
Inteligência Artificial**

**Trabalho de Conclusão de Curso
apresentado à Escola Politécnica da
Universidade de São Paulo**

SÃO PAULO

2018

GUSTAVO KEN HONDA
MICHEL CHIEREGATO GRIETSCHICHKIN
PEDRO LUI NIGRO CHAZANAS

**Sistema de gerenciamento de vendas e controle de estoque usando
Inteligência Artificial**

**Trabalho de Conclusão de Curso
apresentado à Escola Politécnica da
Universidade de São Paulo**

**Área de Concentração:
Engenharia Elétrica com Ênfase em
Computação**

Orientador: Prof. Dr. Fábio Levy Siqueira

SÃO PAULO

2018

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catálogo-na-publicação

Honda, Gustavo Ken
Sistema de gerenciamento de vendas e controle de estoque usando
Inteligência Artificial / G. K. Honda, M. C. Grietschichkin, P. L. N. Chazanas -
- São Paulo, 2018.
67 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São
Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Engenharia 2.Engenharia Elétrica 3.Engenharia de Computação 4.Curso de
Graduação 5.Ensino e Aprendizagem I.Universidade de São Paulo. Escola
Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.
III.Grietschichkin, Michel Chieregato IV.Chazanas, Pedro Lui Nigro

AGRADECIMENTOS

“Agradeço primeiramente ao nosso orientador Prof. Dr. Fábio Levy Siqueira pela sua dedicação e pela paciência nas nossas discussões durante o andamento do projeto. Agradeço aos meus pais Mônica e Marcos por sempre acreditarem em mim e no meu potencial, pois sem essa crença e investimento eu não estaria aqui. Agradeço aos meus colegas politécnicos que me ajudaram muito academicamente, e também agradeço aos meus amigos tanto politécnicos quanto não politécnicos, sobretudo a Victor Funabashi, meu companheiro que nunca me deixou nas horas mais difíceis. Finalmente, agradeço à minha namorada Mayara Clemente de Souza Leitão por todo o carinho e apoio emocional”.

Gustavo Ken Honda

“Gostaria de agradecer a minha família por todo apoio e amizade: meu pai, Igor e Patrícia, meu irmão Felipe e a minha namorada Camila, que sempre está ao meu lado. Vocês me fazem muito feliz. Agradecer aos meus companheiros politécnicos, principalmente aos membros da atlética, meus melhores amigos, e ao Victor Funabashi, pois sem ele não teria me formado. E um agradecimento especial ao Prof. Fábio Levy Siqueira, por toda paciência nesse projeto e por nos lembrar que ainda existem ótimos professores”.

Michel Chierigato Gretschischkin

“Agradeço fortemente ao nosso orientador Prof. Dr. Fábio Levy Siqueira, por compartilhar seus conhecimentos e experiências ao longo de todo o projeto, e também pelo seu zelo em nos ensinar e nos corrigir quando necessário. Agradeço aos meus pais, João e Silvia, e ao meu irmão Ricardo, pelo carinho, pela compreensão e pela insistência na minha capacidade que me permitiram chegar a este momento. Agradeço aos meus amigos politécnicos, principalmente a Victor Funabashi, pelos seus conselhos essenciais ao projeto. Por fim, agradeço minha namorada Ligia Cavalcante Delgado, que me apoia e me fortalece há anos, tornando tudo isso possível”.

Pedro Lui Nigro Chazanas

RESUMO

O projeto tem como objetivo desenvolver um sistema de gestão empresarial com um serviço de previsão de vendas utilizando técnicas de inteligência artificial, aplicando-o em uma pequena empresa que fabrica e comercializa uniformes escolares. A firma de pequeno porte enfrenta alguns problemas na efficientização das suas atividades operacionais e na falta de instrumentos de análises de dados para auxiliar a administração da empresa na sua tomada de decisão. O presente trabalho teve como diretiva a utilização da metodologia ágil *Scrum* e a técnica de coleta de requisitos funcionais de *Histórias de Usuário*.

Palavras-chave: Sistema de Gestão Empresarial, Vendas, Previsão, Inteligência Artificial, *Scrum*, *Histórias de Usuário*.

ABSTRACT

The project has the main objective of developing an Enterprise Resource Planning system with a sales prediction service using Artificial Intelligence techniques that has a real use application on a small company that manufactures and sells school uniforms. The company faces some efficiency problems in its operational activities and a lack of data analysis tools to help the business' administration in its decision making. The present work has as a directive to use the agile methodology Scrum and the functional requirement gathering technique called User Stories.

Keywords: Enterprise Resource Planning System, Sales, Prediction, Artificial Intelligence, Scrum, User Stories.

LISTA DE FIGURAS

Figura 1: Exemplo de tela do sistema Protheus da empresa TOTVS.	22
Figura 2: Exemplo de tela do sistema de Planejamento de Recursos Empresarias da Mestre Digital	24
Figura 3: Diagrama Entidade-Relacionamento do projeto baseado no esquema já existente.....	36
Figura 4: Tabela que mostra exemplo de dados	38
Figura 5: Índices de vendas de camiseta manga curta em 2017.	39
Figura 6: Índices de vendas de calça de moletom em 2017.	39
Figura 7: Erros em peças de roupas previstas de diferentes formas de previsões para base de validação (ano de 2018).....	41
Figura 8: Diagrama representando a arquitetura do projeto.....	43
Figura 9: Tela mostrada para o vendedor ao se autenticar.....	47
Figura 10: Tela de adição de produtos da funcionalidade de venda.	48
Figura 11: Tela de pagamento da funcionalidade de venda.....	49
Figura 12: Tela de pesquisar venda.	50
Figura 13: Tela apresentada ao administrador após autenticação.....	51
Figura 14: Tela de controle de estoque do administrador.	52
Figura 15: Tela de previsão de vendas.	53
Figura 16: Trecho de relatório de previsão de vendas por produto.	54

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Objetivo	8
1.2	Justificativa	8
1.3	Escopo	9
1.4	Método	9
1.5	Organização do Trabalho	10
2	REVISÃO BIBLIOGRÁFICA	11
2.1	Sistemas de Gestão Empresarial.....	11
2.2	Inteligência Artificial e Aprendizado de Máquina	13
2.2.1	Algoritmos de árvore.....	14
2.3	Desenvolvimento de Software com Scrum.....	15
2.3.1	Scrum	16
2.3.2	Histórias de Usuário	18
2.4	Conclusão.....	19
3	PESQUISA DE MERCADO.....	20
3.1	Visão Geral da Empresa Cliente	20
3.2	Análise de Sistemas de Gestão Empresarial.....	21
3.2.1	TOTVS.....	21
3.2.2	Benner	22
3.2.3	Mestre Digital.....	24
3.3	Conclusão.....	25
4	ESPECIFICAÇÃO DOS REQUISITOS.....	26
4.1	Requisitos Funcionais	26
4.1.1	Sistema de Vendas.....	27

4.1.2	Previsão de Vendas.....	27
4.1.3	Gerência de Estoque.....	28
4.1.4	Relatórios para a Administração.....	29
4.2	Requisitos Não Funcionais	29
4.3	Conclusão.....	30
5	ORGANIZAÇÃO DO DESENVOLVIMENTO DO PROJETO	31
5.1	Definição das Releases	31
5.2	Definição das Sprints.....	32
5.3	Conclusão.....	32
6	ARQUITETURA E TECNOLOGIA	33
6.1	<i>Front-End</i>.....	34
6.2	<i>Back-End</i>.....	35
6.2.1	Componente de lógica de negócio	35
6.2.2	Banco de Dados	35
6.2.3	Inteligência Artificial.....	37
6.3	Integração dos Componentes.....	42
6.4	Conclusão.....	44
7	DESENVOLVIMENTO DO PROJETO	45
7.1	O <i>Scrum</i> Aplicado ao Projeto.....	45
7.2	Sistema	46
7.2.1	Funcionalidades do vendedor.....	47
7.2.2	Funcionalidades do administrador	51
8	TESTES E AVALIAÇÃO	55
8.1	Teste de Garantia de Qualidade.....	55
8.2	Teste de Sistema.....	55
8.3	Teste de Aceitação.....	56

8.4	Avaliação Final	57
9	CONCLUSÕES	58
9.1	Trabalhos Futuros.....	58
9.2	Contribuições do Trabalho.....	59
9.3	Conclusão Final.....	60
	REFERÊNCIAS.....	61
	APÊNDICE A - CRONOGRAMA INICIAL DO PROJETO	63

1 INTRODUÇÃO

Atualmente, grandes empresas possuem recursos suficientes para empregar sistemas de gestão empresarial especializadas para suas necessidades e também técnicas de inteligência artificial somadas com estratégias de negócios para realizar suas próprias análises. Por outro lado, empresas médias e pequenas geralmente não tem recursos para investir nesses estudos, e também não dispõem de seu próprio sistema de estoque e vendas, tornando essas firmas dependentes de serviços de ERP (*Enterprise Resource Planning*) de empresas terceirizadas. Esses, normalmente, possuem uma interface complexa, o que pode resultar numa utilização difícil por parte do cliente, e por causa disso, o usuário não sabe explorar todas as funções do sistema terceirizado (MENDES e ESCRIVÃO FILHO, 2007). Diante deste cenário, este projeto propõe uma solução mais simples, aplicável a empresas de pequeno e médio porte.

Este projeto inicialmente foi desenvolvido para um pequeno comércio de uniformes escolares, o qual possui 4 unidades físicas e atualmente utiliza um sistema ERP com os mesmos problemas descritos para realizar a venda sem utilizar de técnicas de análise.

1.1 Objetivo

O projeto consiste na criação de um sistema de vendas e controle de estoque para pequenos comércios utilizando técnicas de inteligência artificial para prever a demanda futura dos produtos da empresa. Isso permite ao empresário reduzir seus gastos na produção evitando desperdícios de estoque, ou obter tanto uma visão geral quanto uma detalhada do seu negócio, motivando-o a explorar oportunidades de melhoria nas operações atuais e possíveis oportunidades de crescimento.

1.2 Justificativa

Para competirem com as empresas de grande porte, as empresas de pequeno e médio porte precisam do mesmo tipo de auxílio nas suas atividades operacionais

suprido nas empresas grandes pelos sistemas de informação para gestão empresarial. No entanto, algumas empresas de pequeno e médio porte não possuem uma estrutura de tecnologia da informação para criarem seu próprio sistema, sendo necessário recorrer à terceirizar com empresas que oferecem esses serviços.

Pelo que foi estudado no capítulo 3, empresas como a TOTVS ou a Benner possuem opções de soluções personalizadas para seus clientes. No entanto, no caso de empresas pequenas este tipo de sistema está fora do escopo por limitações financeiras. Um sistema que satisfaz esse balizamento é o pertencente à empresa Mestre Digital, pois introduz uma solução geral para todos seus clientes.

1.3 Escopo

O projeto tem seu escopo limitado a uma empresa de uniformes escolares de porte pequeno, tendo em vista suas demandas de negócio específicas. Foi planejado para a empresa cliente um escopo inicial de um sistema de gestão empresarial que realiza as mesmas funções já utilizadas pelos usuários do sistema de informação anterior com um adicional do módulo de previsão de vendas usando inteligência artificial para o controle mais eficiente do estoque. Além disso, alguns recursos adicionais do projeto conseguiram também serem desenvolvidos.

1.4 Método

O projeto foi desenvolvido como um projeto de produto de software, sendo trabalhado primeiramente o problema em conjunto com o cliente, gerando as especificações do projeto. Após este passo, foi criada a solução a partir das especificações e planejamento do projeto.

As informações necessárias para o desenvolvimento do problema foram fornecidas principalmente pela empresa cliente. Já as informações utilizadas na criação da solução foram pesquisadas e estudadas a partir dos conteúdos já existentes tanto em trabalhos científicos quanto na internet.

A avaliação dos resultados do projeto foi feita a partir da validação com o gestor e os funcionários da empresa de uniformes escolares com a utilização do sistema em uma das unidades.

1.5 Organização do Trabalho

No capítulo dois são apresentados os conceitos que foram essenciais para o desenvolvimento do projeto, tanto para a definição do problema como para a definição da solução.

No capítulo três faz-se uma análise sucinta do público alvo e das soluções existentes no mercado.

No capítulo quatro apresenta-se os requisitos funcionais e não funcionais coletados a partir das necessidades da empresa cliente.

No capítulo cinco foi apresentado a definição de entregáveis do projeto e de pacotes de trabalho referenciando o cronograma localizado no apêndice.

No capítulo seis detalha-se e justifica-se a escolha da arquitetura da solução, as tecnologias escolhidas pelo grupo de trabalho.

No capítulo sete é explicitado o processo de desenvolvimento do projeto de forma objetiva, incluindo possíveis disparidades com o planejamento das outras seções.

No capítulo oito apresenta-se os testes realizados para a validação da solução e também são apresentados os resultados dos testes do sistema.

Por fim, no capítulo nove, são apresentadas as contribuições do trabalho, possíveis trabalhos futuros e por fim a conclusão final do trabalho.

2 REVISÃO BIBLIOGRÁFICA

Esta seção apresenta o embasamento teórico que foi necessário para o desenvolvimento da solução.

Primeiramente, são apresentadas as definições dos conceitos de um sistema de informação computacional para negócios, mais especificamente de gestão empresarial.

Em seguida são apresentados os conceitos de inteligência artificial, a definição da ideia, as técnicas existentes e as possíveis aplicações para o projeto.

Finalmente, apresenta-se o conceito de métodos ágeis de desenvolvimento, discutindo-os especificamente na área de desenvolvimento de sistemas computacionais para a aplicação no trabalho.

2.1 Sistemas de Gestão Empresarial

Os sistemas de gestão empresarial, também conhecidos como sistemas ERP (*Enterprise Resource Planning*) são sistemas de informação computacionais cuja finalidade é auxiliar uma empresa na tomada de decisão, seja por meio da visualização do cenário interno e externo do negócio, seja pela criação de análises com esses mesmos dados (MORESI, 2000).

Segundo Reynolds e Stair (2011), esses sistemas empresariais são centrais para a organização e precisam garantir que as informações sejam transparentes para a empresa. Empregando-se de banco de dados operacionais e de planejamento, o sistema empresarial consegue registrar e consultar os dados necessário ao negócio.

Exemplos de funções de sistemas empresariais incluem o processamento de pedidos; a administração do estoque e das compras; o gerenciamento de relacionamento com consumidores; e a manutenção do marketing e dos processos de serviços de atendimento ao consumidor.

É importante ressaltar que, de acordo com Rajan e Baral (2008) cerca de dois terços dos projetos que envolvem a aplicação de um sistema de ERP em uma empresa fracassam. Os autores explicam que grande parte dos problemas destes projetos não são exclusivamente causados pelas limitações técnicas, mas também por vários fatores comportamentais dos seus usuários. As organizações responsáveis por esses projetos precisam entender a perspectiva do usuário do sistema e não somente focalizar nos benefícios que tal ferramenta traria para a empresa. Segundo Davis e Venkatesh (2000) existem duas variáveis cruciais que definem o sucesso da aplicação do sistema ao negócio. São essas a percepção do usuário sobre a utilidade do ERP, ou seja, o ganho que o usuário entende ser possível extrair do sistema, e a percepção do mesmo na facilidade de uso do ERP.

Essas duas percepções podem ser influenciadas por fatores individuais, fatores organizacionais e fatores tecnológicos (RAJAN; BARAL, 2008). Os fatores individuais concernem principalmente na confiança do usuário em manusear um sistema computacional e cumprir seus objetivos adequadamente. Os fatores organizacionais incluem o suporte técnico e o treinamento. Os fatores tecnológicos são a complexidade do sistema de informação e a compatibilidade do produto com sistemas já disponíveis na empresa.

Têm-se como fatores que levam à adoção do sistema ERP pelos usuários o empoderamento do cliente na tomada de decisão, devido ao acesso transparente da informação (PSOINAS, KERN; SMITHSON, 2000) e o desempenho individual, geralmente subjetivo de cada utilizador ao comparar seu desempenho com e sem o sistema de informação (LAW; NGAI, 2007).

Os resultados do uso do ERP são retro-alimentados às variáveis de percepção do usuário, ou seja, também influenciam no sucesso da adoção sistema de informação pelos usuários.

Com isso, apesar dos potenciais benefícios que um sistema de informação empresarial pode trazer ao negócio, deve-se verificar primeiramente os fatores de contexto e, a partir deste estudo, adotar um ERP adequado, levando em conta as demandas dos usuários e os possíveis entraves para a adoção da nova tecnologia.

2.2 Inteligência Artificial e Aprendizado de Máquina

A IA (Inteligência Artificial) pode ser definida como o estudo de computações que conseguem perceber, pensar e agir (WINSTON, 1992). O primeiro trabalho reconhecido como de IA é de 1943, quando Walter Pitts e Warren McCulloch juntaram conhecimentos de neurociências, lógica propositiva e a teoria de Turing da computação (RUSSEL, 2009). Apesar desse estudo datar de tanto tempo atrás, a aplicação desse conhecimento levou muito tempo para acontecer, principalmente pela limitação de *hardware* dos computadores primitivos (RUSSEL, 2009).

O campo de Aprendizado de Máquina é uma parte da IA que se dedica a construir programas de computador que se aprimorem automaticamente com experiência (MITCHELL, 1997). Projetos que aplicam conhecimentos dessa área são diversos, como programas que sugerem tratamentos médicos a partir de casos passados, sistemas bancários que preveem a probabilidade de pagamento de uma dívida dado o histórico do devedor, ou até mesmo aplicativos que deduzem as próximas compras dos clientes em uma loja. Nas últimas décadas, o crescimento do Aprendizado de Máquina, tanto no mercado, quanto na academia, tem sido cada vez maior (MITCHELL, 1997), tornando assim esse estudo extremamente importante.

De acordo com Russel (2009), tem-se três tipos de *feedback* que determinam três tipos de aprendizado. O aprendizado não supervisionado reconhece padrões nos dados mesmo quando não há um *feedback* explícito do mesmo. O maior exemplo deste tipo de aprendizado é a clusterização, quando se determina diversos grupos nos dados de entrada, porém sem que estes sejam previamente rotulados para efeito de comparação.

O segundo tipo de aprendizado segundo Russel (2009) é o reforçado, quando o agente muda seu comportamento depois de uma série de eventos chamados prêmios e punições. De uma maneira geral, o programa continua com determinado comportamento se seu resultado se aproxima do esperado, ou muda sua abordagem em caso contrário.

O último tipo é o supervisionado, quando o programa desenvolve, segundo (RUSSEL, 2009), uma função gerada a partir de um conjunto de entradas e saídas

de dados. Russel (2009) também explica que se deve dividir as informações disponíveis em uma parte de treino, sob a qual será desenvolvido o método que encontra uma dada saída para sua respectiva entrada a partir de um certo algoritmo; e uma parte de teste, onde o modelo desenvolvido será testado, comparando as respostas geradas com as esperadas e medindo as diferenças por algum método conveniente (acurácia, diferença quadrática média, área sob a curva ROC, etc.).

O aprendizado supervisionado pode resolver tanto problemas de classificação quanto de regressão (RUSSEL, 2009). O primeiro tipo ocorre quando o número de saídas diferentes possíveis por entrada é limitado, o que ocorre normalmente quando se deseja separar os dados em grupos (RUSSEL, 2009). O último tipo ocorre quando se deseja prever um número real associado a cada dado, ou seja, ao invés de separar as entradas em grupos, se estima um valor de uma variável relacionada a cada uma delas (RUSSEL, 2009).

Existem diversas maneiras de validar a qualidade de um modelo de *machine learning*, e uma das mais comuns é a validação cruzada (RUSSEL, 2009). Este método consiste na divisão aleatória dos dados de em diversos subgrupos, e dentre destes se tem uma nova divisão entre conjunto de teste e treino. O modelo é treinado com essa pequena parte das informações, aplicado na seção de testes e a métrica gerada é armazenada. No final de todas as iterações, é calculado uma média de todas as métricas geradas (RUSSEL, 2009). É um método bom para dados distribuídos aleatoriamente, e tem a vantagem de garantir que a solução dada funciona para uma certa variedade de dados (RUSSEL, 2009).

2.2.1 Algoritmos de árvore

O algoritmo da Árvore de Decisão é um dos métodos mais comuns no universo de *Machine Learning* para prever cenários (MITCHELL, 1997). Deste, muitos outros algoritmos foram gerados (MITCHELL, 1997), e dentre eles, pode-se citar *Random Forrest* e *XGBoost* como os mais notórios.

Mitchell (1997) descreve o funcionamento do algoritmo da Árvore de Decisão. Inicialmente, o algoritmo organiza as variáveis da entrada em um formato de árvore. Com isso, se escolhe uma variável e a partir dela se divide os dados, resultando em

subgrupos homogêneos na incógnita em questão (por exemplo, divisão de altura de edifícios entre menores e maiores de um dado valor). Se os dados de saída do subgrupo forem muito próximos (baixo desvio padrão), se declara o fim do ramo. Se não, se subdivide novamente o grupo de dados a partir de uma nova variável. O processo se repete até que todos os subgrupos tenham baixo desvio padrão, ou quando a altura da árvore máxima for atingida, ou quando as incógnitas se esgotarem. A ordem de variáveis de divisão de grupos se dá por aquelas que geram a maior redução de desvio padrão resultante. Para a dedução de valor para a saída, se faz uma regressão linear com os resultados do subgrupo ao qual pertence à entrada.

A *Random Forest* é um algoritmo baseado na Árvore de Decisão, porém costuma apresentar um desempenho melhor (BREIMAN, 2001). Segundo Breiman (2001), o classificador *random forest* consiste em uma coleção de árvores de decisão geradas por vetores aleatórios identicamente distribuídos, em que cada árvore sugere em um valor de classificação, sendo o mais comum dentre todas as árvores é escolhido pela floresta. No caso de regressões, o mesmo se aplica, porém, as árvores são regressoras e o valor escolhido é uma média (BREIMAN, 2001). Esse algoritmo costuma gerar resultados melhores que a árvore porque ele é mais genérico, ou seja, leva em consideração mais cenários, permitindo que o programa aprenda uma maior variedade de entradas e evitando assim o chamado *overfitting*, quando um previsor é perfeito para um conjunto limitado de dados, porém é muito impreciso para outras entradas, por ser muito específico (BREIMAN, 2001).

Os algoritmos de *boosting* aplicados a *Random Forest* fazem sucessivas regressões lineares com os valores das árvores visando reduzir a função de erro. Uma das implementações de melhores resultados é a *XGBoost* (CHEN, 2016).

2.3 Desenvolvimento de Software com Scrum

Segundo o SWEBOK (IEEE, 2014), uma metodologia de engenharia de software tem como objetivo tratar o desenvolvimento como uma atividade sistemática, repetitiva e, assim, orientada ao sucesso. Tem-se, portanto, uma

abordagem regrada para a especificação, design, construção e teste do software e seus produtos associados.

Para o processo de desenvolvimento desse projeto, optou-se por utilizar o *Scrum*, um framework ágil. Além disso, foi utilizado a técnica de histórias de usuário para descrever as funcionalidades do sistema pela visão dos clientes.

2.3.1 Scrum

Métodos ágeis são métodos de desenvolvimento de software que possuem características como ciclos iterativos de desenvolvimento, times auto organizados, design simples, constante refatoração de código, desenvolvimento guiado por testes, alto envolvimento do cliente e com ênfase em apresentar um produto funcionando a cada ciclo de desenvolvimento. (IEEE, 2014).

O *Scrum* é um *framework* ágil utilizado na área de desenvolvimento de software com o intuito de agilizar o processo de desenvolvimento. Sua aplicação em projeto inicia-se com a criação de um *Product Backlog*, que, segundo Rubin (2013), é uma lista de características e funcionalidades desejadas que um determinado produto deve conter, baseada na visão do cliente e no que ele quer criar.

Essa lista prioriza as características mais importantes, deve também ser inteligível, além de ter como objetivo a fácil comunicação entre o mundo do negócio e o mundo técnico. Assim, a equipe sempre trabalhará na ordem de maior prioridade para menor prioridade. Os elementos da lista são tratados de forma contínua ao longo do projeto, com isso, o *Product Backlog* pode e deve ser constantemente alterado.

Para alinhar a organização do *Product Backlog* com o desenvolvimento do produto e entrega de resultados, é necessária uma organização interna da equipe, denominada de *Time Scrum*. De acordo com Schwaber e Sutherland (2013), esse time é dividido em três papéis: *Product Owner*, *Time de Desenvolvimento* e o *Scrum Master*.

O *Product Owner* é o responsável por maximizar o valor do produto e do trabalho do *Time de Desenvolvimento*. Ele deve gerenciar o *Product Backlog*, sendo responsável por expressar claramente os itens do *Backlog* e ordená-los, garantindo que todo o *Time de Desenvolvimento* os entenda com clareza (SCHWABER; SUTHERLAND, 2013).

O *Time de Desenvolvimento* consiste nos profissionais que realizam o trabalho de entregar um incremento do produto a cada nova iteração. É um *time* auto organizado e multifuncional, possuindo todas as habilidades necessárias para o incremento do produto (SCHWABER; SUTHERLAND, 2013).

O *Scrum Master* é responsável por garantir que o *Scrum* seja entendido e aplicado. Ele auxilia o *Product Owner* a encontrar técnicas para o gerenciar o *Product Backlog*, ao passo que o ensina a criar os itens de forma clara e concisa. Exerce, também, serviço para o *Time de Desenvolvimento*, treinando-o para seu autogerenciamento e interdisciplinaridade, levando-os a completa adoção do *Scrum*. Por fim, é responsável por gerenciar os *Eventos Scrum* (SCHWABER; SUTHERLAND, 2013).

No *framework Scrum*, o ciclo de desenvolvimento do produto, que é iterado múltiplas vezes ao longo do projeto, é chamado de *Sprint*. As *Sprints* têm um tempo fixo – que tende a ser o mesmo durante todo o projeto – e geralmente pode ter sua duração fixa definida de uma semana ou até um mês. Esse limite fixo de tempo é essencial, pois obriga a equipe a priorizar tarefas, sempre demonstrar progresso, evitar perfeccionismos desnecessários e ter *feedbacks* constantes (RUBIN, 2013).

As *Sprints* são compostas por uma reunião de planejamento da *Sprint*, reuniões diárias acompanhando o progresso, uma revisão da *Sprint* e a retrospectiva da mesma (SCHWABER; SUTHERLAND, 2013).

A reunião de planejamento engloba todo o *Time Scrum*. Nela, deve-se definir o que deve ser entregue como incremento no final da *Sprint*. Isso deve ser feito com base nos itens definidos no *Product Backlog* e nas suas prioridades. Assim, o *Time de Desenvolvimento* avalia o que pode ser completado ao longo da *Sprint*.

Posteriormente, o *Time de Desenvolvimento* avalia como devem ser realizadas as histórias de usuário, normalmente quebrando-as em tarefas menores. Dá-se, assim, origem ao *Sprint Backlog*. Por fim, define-se o objetivo dessa *Sprint*, encerrando a reunião de planejamento.

Todos os dias, são feitas reuniões diárias, que devem ser curtas e dinâmicas. A Reunião Diária do *Scrum* é um evento curto de 15 minutos, para que o *Time de Desenvolvimento* possa sincronizar as atividades e criar um plano para as próximas 24 horas (SCHWABER; SUTHERLAND, 2013).

Segundo Aniea Cole (2015), durante a reunião, devem ser esclarecidos os seguintes pontos:

1. O que eu fiz ontem para ajudar o time a atender a meta da *Sprint*?
2. O que eu farei hoje para ajudar a atender a meta da *Sprint*?
3. Estou com algum obstáculo que me impeça de atender a meta da *Sprint*?

Ao final da *Sprint* são feitas, ainda, mais duas reuniões, que podem ser realizadas no mesmo dia: a revisão e a retrospectiva da *Sprint*. A primeira é feita com o intuito de se inspecionar o incremento e adaptar o *Backlog* do Produto. Isso garante que o *Backlog* seja sempre atualizado, e o produto seja constantemente ajustado para atender novas oportunidades.

Já na retrospectiva da *Sprint*, o *Time Scrum* inspeciona a si próprio e cria um plano para melhorias a serem aplicadas no próximo processo iterativo (SCHWABER; SUTHERLAND, 2013). Isso garante que o Time sempre se avalie e possa implementar melhorias na sua forma de trabalho e de aplicação do framework *Scrum*.

2.3.2 Histórias de Usuário

No SWEBOK (IEEE, 2014) se observa que as histórias de usuário são uma maneira curta de se ter descrições de alto nível das funcionalidades requeridas pelo

produto nos termos do cliente e, por isso, são amplamente utilizadas em metodologias adaptativas, como os métodos ágeis.

Ainda segundo o SWEBOK (IEEE, 2014), a história de usuário deve conter apenas informação suficientes para que os desenvolvedores consigam produzir adequadamente uma estimativa do esforço necessário para se implementar uma determinada característica do produto. Visa também evitar dispendir uma grande quantidade de tempo em longos requisitos de maneira precoce, que podem se tornar inválidos assim que o desenvolvimento começar.

Uma forma sugerida por Rubin (2013) para escrever uma história é especificar o papel do usuário no negócio, seja ele um vendedor ou administrador, qual o objetivo dele com esse requisito e como ele será beneficiado com isso. Assim, pode-se escrever a história do usuário da seguinte maneira (RUBIN, 2013).

COMO <papel do usuário>, **QUERO** <objetivo> **PARA** <benefício>.

Segundo Rubin (2013), as histórias de usuários são comumente usadas no Scrum para representar os itens do *Product Backlog*, sendo elas feitas e priorizadas pelo *Product Owner*, e posteriormente estimadas por pontos, isso é, uma estimativa de duração relativa delas, ou seja, em quanto tempo a equipe espera conseguir projetar e implementar aquele requisito.

2.4 Conclusão

Com os conceitos de sistema de gestão empresarial, inteligência artificial e o processo de desenvolvimento de software *Scrum* com a técnica de especificação de requisitos de histórias de usuário, o próximo capítulo apresenta um aprofundamento do contexto de negócio em que o projeto atua, tanto na empresa cliente que tem o projeto implementado quanto em soluções já existentes do mercado.

3 PESQUISA DE MERCADO

Nesta seção será apresentado a empresa cliente que utilizará o sistema desenvolvido no projeto e algumas soluções existentes do mercado.

3.1 Visão Geral da Empresa Cliente

O projeto foi aplicado em uma empresa que denominaremos de *empresa cliente*. Com esse cliente coletamos os requisitos funcionais e não funcionais e também o retorno das iterações funcionais do projeto para futuras melhorias.

A empresa em questão é um comércio e fabricação de uniformes escolares que realiza suas operações em quatro unidades físicas dentro das próprias unidades da escola, é composta por poucos funcionários e um gestor.

Devido ao fato de utilizar o nome e o espaço da escola, a empresa precisa compartilhar parte de sua receita, sendo necessário um estudo de preços considerando o custo de fabricação para garantir um fluxo de receitas acima do prejuízo, ficando à cargo do gestor tomar as decisões necessárias para o futuro do negócio.

Atualmente a empresa cliente utiliza um sistema de informação empresarial da empresa Mestre Digital, sistema que será descrito com mais detalhes na próxima seção. Segundo os próprios funcionários da loja, apesar de o sistema possuir todas as funções necessárias para o funcionamento da loja, por causa da sua dificuldade de uso os funcionários se limitam somente às funções de venda e de estoque. Além disso, como o sistema não possui um manual de instruções para referência, o suporte técnico da Mestre Digital é responsável por atender às dificuldades de uso da empresa cliente. Por fim, o software não possui qualquer ferramenta de previsão de volume de vendas, dependendo de análises feitas pelo próprio gestor.

3.2 Análise de Sistemas de Gestão Empresarial

Estudou-se os sistemas de gestão empresarial existentes no mercado. Foram analisadas algumas empresas que oferecem um sistema de informação computacional para esse fim, com foco no que o produto pode prover aos seus clientes.

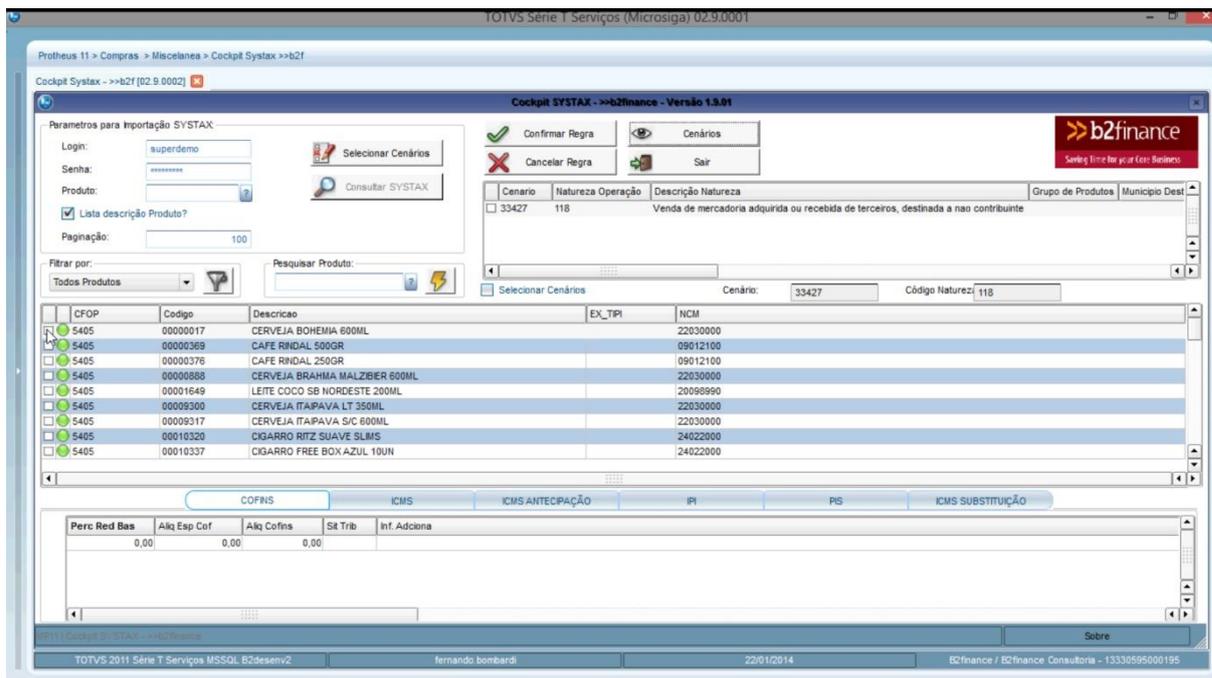
3.2.1 TOTVS

Segundo a TOTVS (2018), a empresa oferece vários produtos relacionados ao auxílio do negócio. Além do produto de ERP, existem produtos de Business Intelligence, e-Commerce, auxílio para o Recursos Humanos, entre outros.

Especificamente sobre o Protheus, produto ERP da TOTVS, a proposta principal do sistema é de possibilitar o controle total de todas as atividades operacionais, administrativas e financeiras. Além do auxílio nos processos, o produto também propõe utilizar os dados coletados nas atividades operacionais para realizar análises e gerar relatórios.

De acordo com a própria empresa, o produto oferece uma solução flexível e completa, ou seja, seu sistema tem como objetivo alcançar uma gama abrangente de clientes de diversos tamanhos e áreas de atuação, e adaptar seu produto às necessidades específicas de cada cliente. Pode-se verificar a partir da Figura 1 que a interface do sistema permite ser utilizado para comércios de áreas distintas. A figura denota uma tela de criação de regras de venda para um comércio, no caso, uma regra para uma bebida alcoólica.

Figura 1: Exemplo de tela do sistema Protheus da empresa TOTVS.



Fonte: Dantas (2015).

Para uma empresa cliente utilizar esse sistema, é necessário um estudo do cliente por parte da própria TOTVS, a partir de entrevistas e pesquisas aprofundadas do contexto do negócio. Posteriormente à pesquisa, é realizado um planejamento da personalização da solução e os possíveis tipos de contratação que o cliente pode optar. Assim que o contrato é realizado, é começado um projeto de implantação do sistema. Após a implantação, é necessário constante acompanhamento das operações do projeto finalizado.

A empresa TOTVS também oferece serviços de inteligência artificial, usados principalmente para previsões e análises não intuitivas dos dados. Contudo, essas aplicações não estão inclusas no sistema ERP. Para serem integradas ao sistema existentes a empresa oferece uma contratação dos serviços a um custo adicional.

3.2.2 Benner

Seguindo o que foi informado pelo Grupo Benner (2018), a empresa oferece dois tipos de sistemas ERP, cada um focando em um escopo diferente.

O produto chamado somente de ERP é visado para empresas de maior porte e infraestrutura tecnológica avançada. O processo de implantação do produto nesses clientes é mais demorado e demanda maior planejamento e desenvolvimento, visto que é necessária maior robustez para suas operações e para a alta demanda das mesmas. A solução oferecida por esse produto é extremamente personalizável, variando muito de cliente para cliente.

O outro produto, o ERP360, diferente do anterior, possui como escopo empresas de menor porte com pouca ou nenhuma infraestrutura tecnológica. O maior atrativo deste produto é a fácil implantação utilizando tecnologia em nuvem com o objetivo de economizar a infraestrutura tecnológica da empresa cliente. Essa solução é mais geral e automatizada para os clientes finais, por consequência é também um serviço mais acessível financeiramente.

Ambos os produtos possuem quatro módulos de funcionalidades. Existe o módulo das ferramentas administrativas, o qual realiza o controle de compras, contratos, importação, produção, qualidade e outros aspectos fundamentais para o negócio, ou seja, atividades essenciais do sistema ERP. Também existem módulos como o portal de clientes e representantes oferecendo soluções para o relacionamento com clientes da empresa, o controle contábil e de obrigações tributárias e por último o módulo financeiro que controla orçamentos, cobranças e fluxos. A contratação de cada módulo fica a cargo do cliente, e a Benner se responsabiliza para prover a solução sempre seguindo as especificações.

Atualmente o grupo Benner (2018) oferece serviços de inteligência artificial exclusivamente para auxílios jurídicos de seus clientes. O produto em questão se chama BIA (*Benner Intelligent Assistant*) e é um *chatbot*, um robô que responde à interações humanas por meio de uma conversa de texto. Ou seja, atualmente a empresa Benner não oferece a funcionalidade de previsão de vendas para comércios.

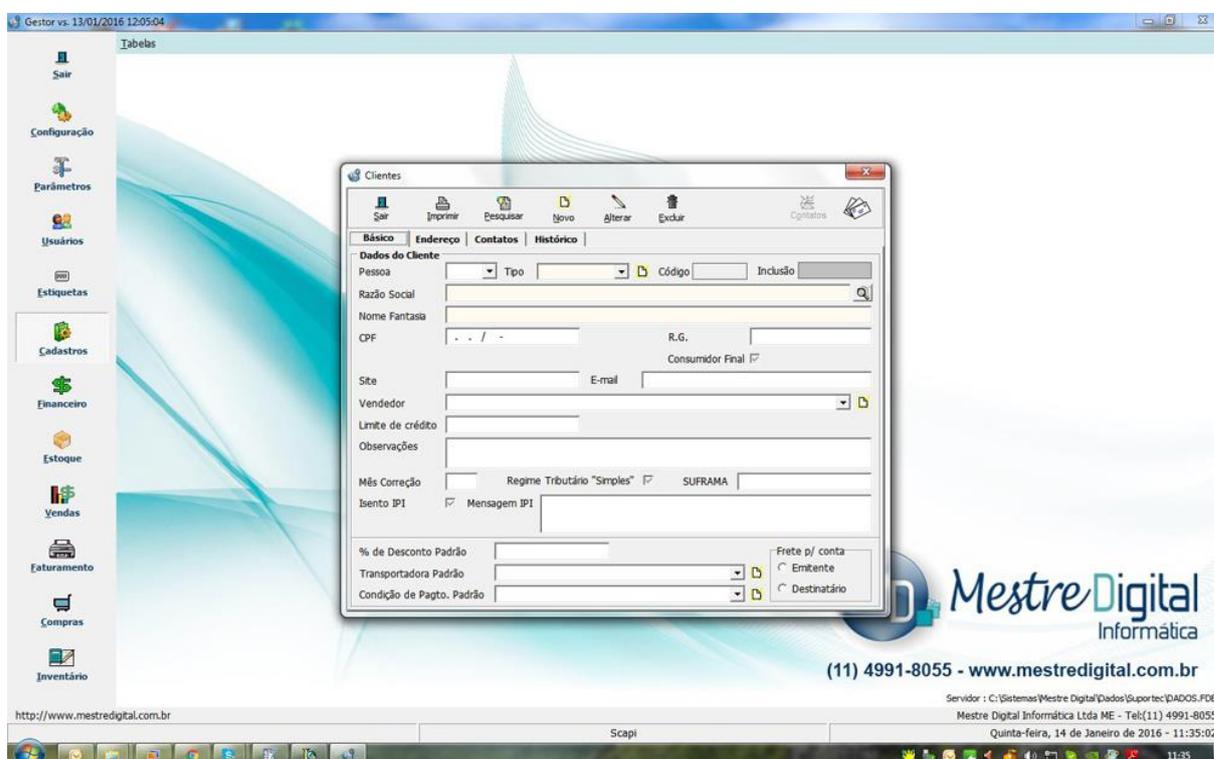
3.2.3 Mestre Digital

De acordo com a Mestre Digital (2018), a empresa tem como seu maior foco a disponibilização de tecnologia de computação para pequenos empresários que não conseguem optar por soluções mais sofisticadas por questões financeiras.

A proposta de produto ERP desta empresa é a integração entre as funcionalidades de negócio em um único módulo, evitando repetições de operações. O sistema possui vários serviços para auxiliar as operações de uma empresa e também serviços de armazenamento e visualização dos dados do negócio.

A Mestre Digital propõe o auxílio em diversas operações de seus usuários, operações estas como o registro de compras de fornecedores, vendas para clientes, cadastro de clientes como mostra a Figura 2, contas a pagar, contas a receber, controle de estoque, faturamento com nota fiscal eletrônica, informações gerenciais e *E-commerce* integrado.

Figura 2: Exemplo de tela do sistema de Planejamento de Recursos Empresarias da Mestre Digital



Fonte: Mestre Digital (2018).

Este produto é o sistema utilizado pela empresa cliente do projeto de formatura para suas atividades operacionais. No entanto, segundo o próprio gestor da empresa de uniformes escolares, grande parte dos recursos oferecidos pelo sistema da Mestre Digital não são utilizados, pois os usuários do sistema não sabem como navegar pelo produto de maneira eficaz, limitando-se somente às atividades prioritárias como registro de compras, controle financeiro e controle de estoque.

Atualmente a empresa não oferece soluções envolvendo tecnologias de inteligência artificial.

3.3 Conclusão

Após o estudo do contexto interno da empresa de uniformes escolares e do contexto externo das empresas que oferecem sistemas de informação para auxílio nas operações de seus clientes, a próxima seção aborda um maior aprofundamento do contexto interno, ao especificar os requisitos da empresa cliente.

4 ESPECIFICAÇÃO DOS REQUISITOS

As especificações de requisitos do sistema foram coletadas no contexto do negócio da empresa cliente, ou seja, tanto os requisitos funcionais quanto os requisitos não funcionais são específicos à empresa, não necessariamente sendo problemas que outros potenciais clientes podem enfrentar.

Primeiramente descreve-se como foi utilizado a técnica de História de Usuário explicada no capítulo 2.3 para coletar os requisitos funcionais da empresa cliente. Dentro dos requisitos funcionais foi feita uma categorização das histórias por componente de negócio, por exemplo, há uma divisão entre histórias que envolvem o sistema de vendas e histórias que envolvem a geração de relatórios para a análise de dados. Ao final são apresentados os requisitos não funcionais.

4.1 Requisitos Funcionais

Para os Requisitos Funcionais do projeto, utilizamos a técnica de histórias de usuários. Utilizando a estrutura já descrita na seção 2.3.2, existem no contexto do negócio dois papéis de usuários distintos, o vendedor e o administrador. Dependendo da história de usuário, somente um dos papéis exerce a determinada história, no caso de ambos exercerem a mesma história, usou-se a denominação de usuário.

As histórias de usuários foram coletadas a partir de entrevistas com a empresa cliente. Primeiramente, foi entrevistado o gestor da empresa, que apresentou o sistema já utilizado pelo negócio e explicitou quais as funções que os funcionários utilizam desta aplicação. Posteriormente foram entrevistados dois vendedores que se encontravam na loja, eles explicaram como usavam as funcionalidades do sistema, relataram dificuldades e sugeriram melhorias. O grupo do projeto sintetizou as entrevistas em histórias de usuários.

Para a facilidade de organização, as histórias foram subdivididas em categorias de negócio de acordo com o contexto.

4.1.1 Sistema de Vendas

Na categoria de sistema de vendas, foram organizadas as histórias de usuário que concernem a atividade operacional de venda da empresa.

- COMO administrador do comércio, QUERO poder adicionar meus produtos PARA depois poder vendê-los.
- COMO vendedor, QUERO poder adicionar os produtos PARA poder realizar a venda.
- COMO vendedor, QUERO consultar os clientes por nome ou cpf PARA adicioná-los à venda.
- COMO vendedor, QUERO poder registrar clientes novos PARA depois adicioná-los à venda.
- COMO vendedor, QUERO poder adicionar a forma de pagamento do cliente PARA poder finalizar a venda.
- COMO vendedor, QUERO ser autenticado no software PARA que uma venda realizada por mim seja registrada para o meu nome.
- COMO vendedor, QUERO poder cancelar uma compra PARA o cliente poder ter uma oportunidade de desistir da compra.
- COMO vendedor, QUERO poder apagar um produto que tenha adicionado errado na hora da venda PARA não ter que fazer toda a venda de novo.
- COMO administrador, QUERO poder supervisionar o dinheiro que se encontra em caixa PARA que eu possa retirar ou depositar novas quantias, além do que já é adicionado pela venda ou retirado pela fabricação.

4.1.2 Previsão de Vendas

As histórias de usuário da categoria de previsão de vendas surgiram da necessidade do gestor de controlar a fabricação de seus produtos, visto que, segundo o próprio administrador, existe uma perda considerável de receita ao fabricar mais de um produto que o necessário, e também há uma perda de receita em potencial quando o cliente da loja quer comprar um produto que está indisponível e precisa ser fabricado. As histórias de usuário a seguir tentam resolver esses dois

problemas utilizando técnicas de inteligência artificial com dados históricos da empresa para gerar uma análise mais precisa do que prever que a venda será igual ao do ano anterior, análise usada pelo administrador atualmente.

- COMO administrador da loja, QUERO saber o volume de vendas futuro PARA que eu possa diminuir os custos com estoque, produzindo somente o necessário.
- COMO administrador da loja, QUERO ter um registro comparativo das vendas de um período com o da previsão de vendas PARA eu poder verificar se a previsão está sendo eficaz.

4.1.3 Gerência de Estoque

As histórias de usuário de gerência de estoque são relacionadas ao gerenciamento tanto do estoque interno de cada loja, como também do estoque entre lojas.

- COMO administrador da loja, QUERO poder alterar facilmente o estoque PARA poder corrigir eventuais erros nos períodos de recontagem de estoque.
- COMO vendedor, QUERO poder fazer transferência de estoque entre duas lojas PARA enviar o produto que está faltando de uma loja para outra.
- COMO administrador, QUERO que haja baixa do estoque automaticamente ao ser feita a venda PARA garantir o controle sobre minhas mercadorias.
- COMO administrador, QUERO ser avisado quando está faltando estoque em alguma unidade PARA produzir mais ou transferir de outra loja.
- COMO vendedor, QUERO poder fazer encomendas PARA realizar a venda mesmo quando não há o produto em estoque.
- COMO vendedor, QUERO poder trocar as mercadorias PARA clientes que receberam produto com defeito ou querem trocar o tamanho do produto.

4.1.4 Relatórios para a Administração

As histórias de usuário que concernem os relatórios para a administração têm como função auxiliar a tomada de decisão da gerência da empresa cliente a partir da coleta de dados das atividades operacionais.

- COMO administrador, QUERO ter um registro completo das vendas do dia por produto PARA eu saber exatamente quais são os produtos que estão sendo vendidos.
- COMO administrador QUERO ter um registro completo das vendas do dia por unidade PARA saber exatamente em qual loja foi cada venda.
- COMO administrador QUERO ter um registro completo das vendas do dia por forma de pagamento PARA eu saber quanto de dinheiro e cheque tem em caixa, bem como quanto entrou de crédito e débito.

4.2 Requisitos Não Funcionais

Os requisitos não funcionais do projeto foram definidos a partir da interpretação do estudo das necessidades da empresa de vendas de uniformes, não somente suas demandas funcionais, mas também seus problemas cotidianos do comércio.

Diferente dos requisitos funcionais, não foi utilizado a técnica de histórias de usuário, pois julgou-se mais apropriado estruturar os requisitos não funcionais como uma especificação formal, evitando ambiguidades, visto que esses requisitos são essenciais para a definição da arquitetura da solução.

A seguir foram listados os requisitos não funcionais considerados essenciais para o contexto do negócio.

- O sistema precisa ser uma aplicação desktop local para não depender completamente da conexão da rede das lojas, visto que a conexão à rede não possui uma disponibilidade garantida.
- O sistema precisa ter um banco de dados que unifique os dados de todas as unidades físicas.

- O sistema precisa ter um mecanismo de armazenamento de dados local caso a conexão de rede apresente problemas, atualizando o banco de dados central assim que a conexão voltar.
- O sistema precisa diferenciar o usuário entre vendedor e administrador, e limitar as ações possíveis dependendo de cada tipo de usuário.
- O sistema precisa atualizar o algoritmo de inteligência artificial periodicamente para evitar problemas na precisão de previsão.

4.3 Conclusão

Com os requisitos funcionais e não funcionais especificados para o projeto, o próximo capítulo apresenta a organização do processo para que a solução fosse desenvolvida.

5 ORGANIZAÇÃO DO DESENVOLVIMENTO DO PROJETO

Utilizando o conceito de *Releases* como explicado na seção 2.3, foram estruturadas as *releases* do sistema definindo quais requisitos funcionais e não funcionais precisavam ser desenvolvidos.

Após estruturar as *releases*, foram definidas as *sprints*, conceito explicado com mais detalhe na seção 2.3, para organizar o desenvolvimento do trabalho.

5.1 Definição das Releases

De acordo com o contexto do negócio, visto que a empresa já possui um sistema de gestão empresarial, julgou-se necessário que a primeira *release* contivesse pelo menos os mesmos recursos já existentes do sistema descrito na seção 3.1 e 3.2.3. Também julgou-se necessário, de acordo com o administrador da empresa cliente, que a primeira *release* também contivesse o recurso da previsão de vendas por inteligência artificial. Foi definido uma segunda *release* com as histórias de usuário e requisitos não funcionais não tratados na primeira *release*.

A primeira *release* teve como objetivo implementar o sistema de vendas, o módulo de inteligência artificial para previsão de vendas e a geração de relatórios gerenciais. São as histórias de usuário presentes na seção 4.1.1, 4.1.2, 4.1.4. Na 4.1.3 somente os requisitos de alterar o estoque, transferir estoque e a baixa de estoque foram julgados essenciais para essa *release*.

Também foram planejados todos os requisitos não funcionais da seção 4.2 com exceção do requisito do funcionamento da venda sem conexão à rede.

Posteriormente, foi planejado uma segunda *release* que aborda todos os requisitos funcionais e não funcionais que ainda não foram completados.

Definidos os requisitos funcionais e não funcionais de cada *release* estruturou-se as *sprints* necessárias para o desenvolvimento do projeto e ao final decidiu-se as datas das *releases* ilustrado no cronograma do apêndice A.

5.2 Definição das Sprints

Definidas as histórias de usuário e requisitos não funcionais para cada release, foram estruturadas as *sprints* de desenvolvimento do projeto, primeiramente decidiu-se que a duração de uma *sprint* seria de duas semanas.

O cronograma do Apêndice A estrutura as *sprints* de acordo com o começo do desenvolvimento na metade de Maio, ou seja, foram definidas no total nove *sprints* para a primeira *release* ao final de Setembro e mais três *sprints* para a segunda *release* na metade de Novembro.

5.3 Conclusão

Seguindo o planejamento e as histórias definidas, projetou-se e implementou-se o sistema. No capítulo seguinte é descrito a escolha de arquitetura e tecnologia pelo grupo para iniciar o desenvolvimento da solução.

6 ARQUITETURA E TECNOLOGIA

Após especificar os requisitos do cliente e também definir as metas de projeto em conjunto com a estrutura de desenvolvimento do trabalho, foi feito um estudo contemplando as possíveis arquiteturas e tecnologias para a solução. A escolha final foi influenciada por fatores críticos do projeto como os requisitos não funcionais.

A decisão arquitetural levou em conta alguns elementos como o armazenamento de informações locais considerando as unidades físicas da empresa, porém sincronizadas com um banco de dados central compartilhado. Decidiu-se também pesquisar padrões de arquiteturas de software que pudessem ser interessantes para o cumprimento dessas necessidades.

O projeto foi dividido em dois módulos, cada um abordando tecnologias diferentes. Eles foram integrados utilizando um protocolo de comunicação somado com uma tecnologia de encapsulamento de dados, cuja finalidade foi de compartilhar a informação facilmente entre os componentes.

O primeiro módulo envolve a parte da interação do usuário com o sistema. Por exemplo, ao tentar realizar uma venda, um vendedor teria que interagir diretamente com essa parte do software, enviando as informações necessárias e recebendo a resposta de suas submissões. Este módulo será chamado de *Front-End*.

O segundo módulo engloba a interação do sistema de *Front-End* com o banco de dados. Se trata do componente que está conectado por uma camada intermediária com o banco de dados, ao mesmo tempo ligado ao primeiro módulo para receber e enviar as informações necessárias, e também engloba o serviço de inteligência artificial. Este módulo será chamado de *Back-End*.

A escolha de cada tecnologia para cada componente será aprofundada mais adiante. O protocolo de comunicação e a ferramenta de encapsulamento de dados serão abordados em conjunto com os módulos, explicitando como estes serão utilizados em cada contexto.

6.1 Front-End

A interface precisou ser uma aplicação local dentro de cada máquina da loja devido ao fato da possibilidade dos estabelecimentos não possuírem um conexão à rede estável. Antes de considerar tal limitação, foi explorado a opção deste módulo ser em *web*, utilizando tecnologias como HTML (*Hypertext Markup Language*), CSS (*Cascading Style Sheets*) e JavaScript. Considerando a limitação, a alternativa escolhida foi utilizar o Electron, que, segundo o GitHub (2018), permite a construção de aplicativos em plataforma *desktop* com todos os recursos das tecnologias JavaScript, HTML e CSS. Isso permite que o desenvolvedor crie um sistema sem precisar de mais conhecimentos além de desenvolvimento web, o qual o grupo deste projeto já possui a familiaridade.

Outra vantagem de utilizar o Electron é que, devido à integração com a plataforma de software aberto GitHub, é possível atualizar o módulo automaticamente desde que o usuário esteja conectado à internet. Esse recurso faz com que uma janela de notificação apareça sempre que o usuário abrir o aplicativo para este ser informado que existe uma nova atualização e possa realizar a operação de renovação rapidamente.

Para o transporte dos dados foi utilizado a tecnologia HTTP (*Hypertext Transfer Protocol*). Já para o encapsulamento de dados a serem transportados pelo HTTP, escolheu-se utilizar a notação JSON (*JavaScript Object Notation*) que, segundo Refsnes Data (2018), é um padrão de texto estruturado que facilita a organização da informação e permite que ela seja transportada pelos canais de comunicação. Sua maior vantagem é a capacidade de converter objetos de uma linguagem de programação para a estrutura da notação JSON, e também a capacidade de converter facilmente deste padrão para o mesmo objeto, sendo facilmente manipulável dentro do ambiente de código.

Nota-se que, para o requisito do funcionamento da venda sem internet, foi necessário estudar soluções envolvendo o armazenamento local de informações e a atualização com o banco central quando possível. Essa discussão é feita na seção 7.2.

6.2 *Back-End*

Este módulo pode ser subdividido em três partes menores, o componente de lógica de negócio, o componente banco de dados e o componente de inteligência artificial.

6.2.1 Componente de lógica de negócio

A tecnologia de servidor escolhida foi o Django, um *framework* baseado na linguagem de programação Python que encapsula a maioria das funcionalidades de um servidor web dentro de métodos de fácil utilização. O Django também possui comandos integrados com vários serviços de armazenamento, servindo como uma camada de abstração entre a aplicação e o banco de dados (DJANGO SOFTWARE FOUNDATION, 2018).

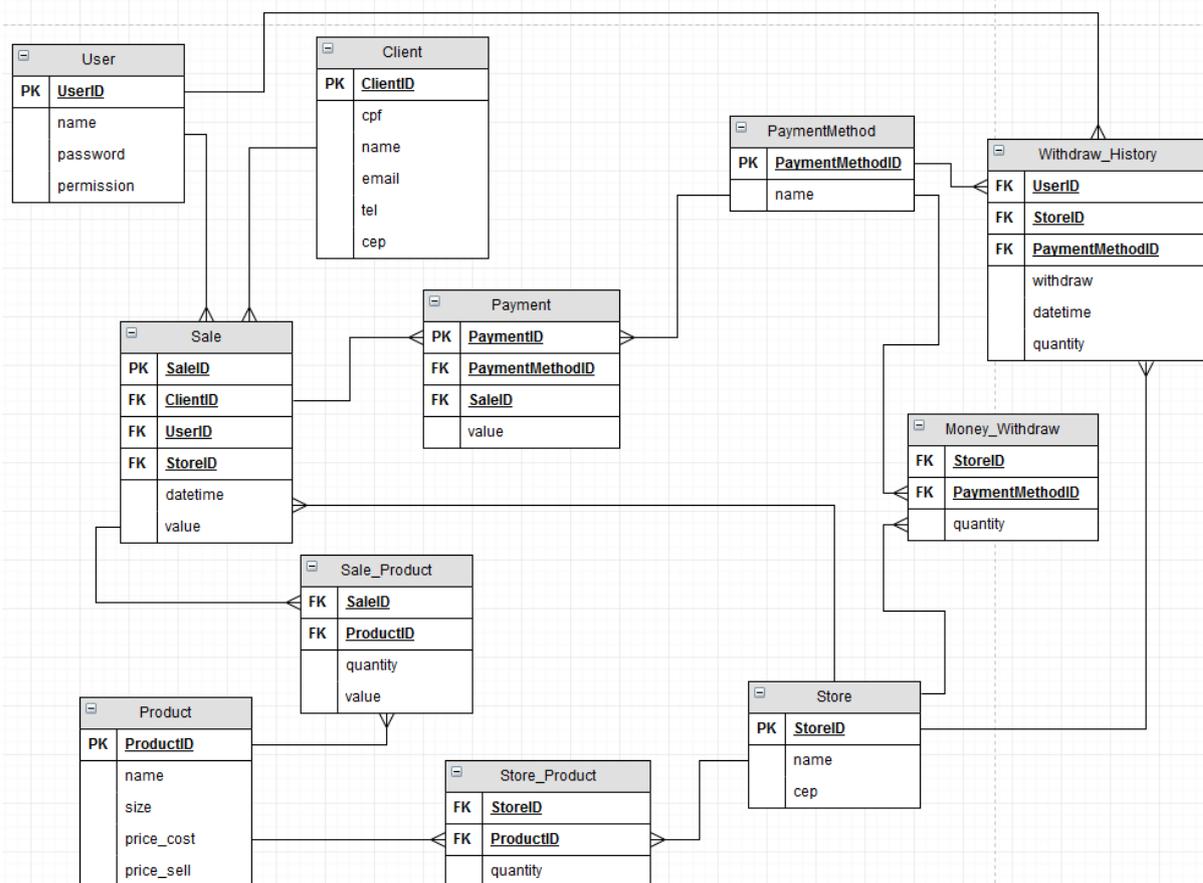
O componente de lógica de negócio realiza a comunicação entre o *Front-End* e o *Back-End* através da tecnologia HTTP com dados encapsulados por JSON. Já sua comunicação com o banco de dados é suprida com as funções integradas da tecnologia Django.

6.2.2 Banco de Dados

A tecnologia de banco de dados utilizada foi o MySQL (2018), o qual trabalha com bancos de dados relacionais. Essa escolha é justificada pelo fato da empresa cliente já utilizar um sistema de gestão empresarial com um banco de dados em MySQL e a necessidade de utilizar seus dados históricos para a previsão de vendas.

A partir do esquema de banco de dados já utilizado pelo sistema vigente na empresa, foi esquematizado na figura 3 um diagrama de Entidade Relacionamento para o entendimento das necessidades do cliente, facilitando o desenvolvimento deste componente e de suas integrações.

Figura 3: Diagrama Entidade-Relacionamento do projeto baseado no esquema já existente



Fonte: Autoria própria (2018).

Na imagem temos as seguintes tabelas:

- *User* representa o usuário do sistema, com nome, senha e a categoria à qual pertence.
- *Client* representa um cliente cadastrado na loja, e possui informações dele como cpf, nome, email, telefone e CEP.
- *Sale* representa uma venda, com informações como o cliente, o usuário e a loja envolvidos, o horário e data em que foi realizado e o valor total da venda.
- *Payment* representa um pagamento, com informações como o método de pagamento, a venda em que foi realizada tal pagamento e o valor.
- *PaymentMethod* representa os métodos de pagamento existentes da loja.
- *Withdraw_History* representa o histórico de movimentação do caixa de uma loja, com informações como o usuário, a loja e o método de pagamento envolvido, se foi uma operação de retirada ou depósito e o horário em que foi realizado a movimentação.

- *Money_Withdraw* representa o valor em caixa de determinada loja, com informações da loja, o método de pagamento envolvido e a quantidade.
- *Store* representa a loja, com cada entrada sendo o nome e o CEP da loja física.
- *Store_Product* representa o estoque de cada loja, contendo informações de uma loja com um produto tendo determinada quantidade.
- *Sale_Product* representa a venda relacionada ao produto, contendo informações como a venda e o produto envolvidos, a quantidade do produto e o valor total.
- *Product* representa os produtos que a loja oferece, contendo informações do produto como nome, tamanho, preço de fabricação (custo) e preço de venda (receita).

6.2.3 Inteligência Artificial

A funcionalidade de previsão de vendas com inteligência artificial foi uma inovação, que visava otimizar a quantidade de mercadoria e reduzir o estoque parado em cada loja, conforme consta na seção 1.

Para o desenvolvimento dessa solução, a tecnologia escolhida foi a linguagem de programação Python, uma ferramenta multifuncional, multiplataforma e de fácil aprendizado (PYTHON SOFTWARE FOUNDATION, 2018).

A justificativa para essa escolha é que, segundo Protasiewicz (2018), a linguagem Python possui uma abundância de bibliotecas e estruturas prontas que podem ser reutilizadas para aplicações de aprendizado de máquina ou inteligência artificial, como o NumPy para computação científica, SciPy para computação avançada, pandas para coleta e análise de dados e bibliotecas específicas que abordam os algoritmos existentes de aprendizado de máquina como Sci-kit Learn, Keras e Tensorflow. Com essa vantagem, alterna-se o foco de desenvolver o algoritmo de *machine learning* para entender e resolver o problema e testar com as várias soluções disponíveis. Por fim, existe uma extensa comunidade de desenvolvedores nessa área que utilizam a linguagem Python, o que garante maior

disponibilidade de conteúdos e documentação da área específicos para essa linguagem.

Para implementar o algoritmo, utilizou-se dados das vendas dos últimos dois anos (2016 e 2017) – obtidas a partir do sistema antigo - como base de treinamento, e as vendas do início de 2018 até o momento como base de validação.

Os dados utilizados seguem uma sequência temporal. Como a validação dos resultados do algoritmo precisa necessariamente ser os dados de vendas de 2018, não foi possível utilizar a técnica denominada de validação cruzada, que aloca aleatoriamente conjuntos para treinamento e conjuntos para teste a fim de refinar o programa de inteligência artificial, limitando-se pela separação simples descrita no parágrafo acima.

A previsão de vendas foi feita por produto e por loja, mês a mês. Como dados de treino foram utilizados o código do produto, seu tamanho e sua quantidade de vendas em cada mês, ano e loja. Quanto à engenharia de dados, também inserimos, para cada linha, a quantidade do produto vendido no mesmo mês do último ano, a média mensal de venda deles nos últimos anos e a quantidade de vendas dele nos últimos quatro meses. Isso pode ser visto na figura 4, que apresenta um trecho dos dados.

Figura 4: Tabela que mostra exemplo de dados

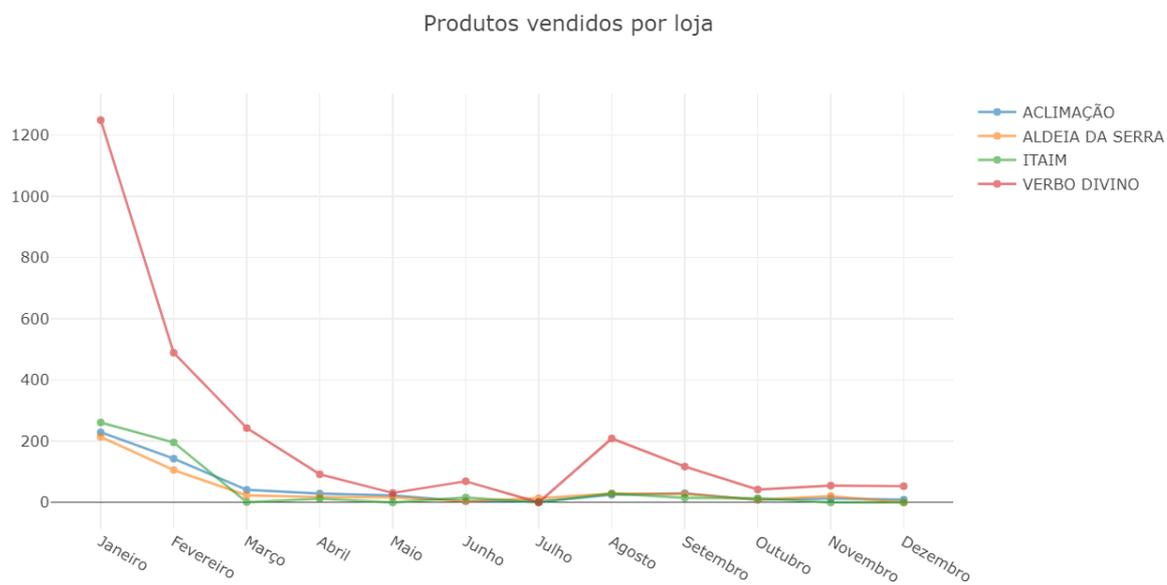
ANO	MES	DESCRICAO	TAMANHO	NOME	LAST_MONTH	LAST_MONTH_2	LAST_MONTH_3	LAST_MONTH_4	LAST_YEAR	LAST_YEAR_2	LAST_YEAR_MEAN	QUANTIDADE
2016	3	JALECO	PP	ITAIM	0.0	0.0	0.0	1.0	0.000000	NaN	NaN	6
2016	3	JALECO	PP	VERBO DIVINO	8.0	14.0	0.0	1.0	0.882353	NaN	NaN	7
2016	3	REGATA	00	ACLIIMAÇÃO	1.0	9.0	0.0	3.0	0.588235	NaN	NaN	0
2016	3	REGATA	00	ALDEIA DA SERRA	3.0	0.0	0.0	0.0	0.117647	NaN	NaN	0
2016	3	REGATA	00	VERBO DIVINO	10.0	9.0	1.0	1.0	1.000000	NaN	NaN	0

Fonte: Autoria Própria (2018).

Um dos desafios para o desempenho do algoritmo de previsão foi a pequena quantidade de dados da empresa. Muito embora existiam dois anos de venda na base de treino, o que resulta em 24 meses de observação, foi notado uma grande sazonalidade nessas vendas. Elas eram muito concentradas no começo do ano, com algum pico entre julho e agosto, principalmente para os produtos mais voltados

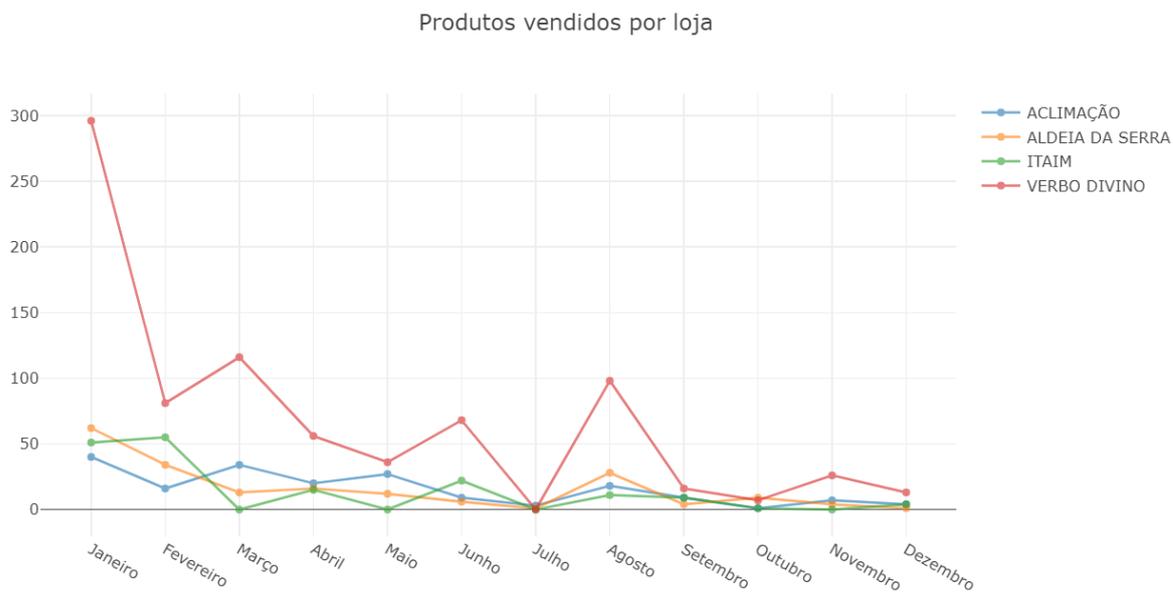
para o inverno. Isso pode ser exemplificado nas figuras 5 e 6, que mostram as vendas de dois tipos de produtos voltado para o calor e para o frio respectivamente.

Figura 5: Índices de vendas de camiseta manga curta em 2017.



Fonte: Autoria Própria (2018)..

Figura 6: Índices de vendas de calça de moletom em 2017.



Fonte: Autoria Própria (2018)..

Portanto, é muito importante coletar a informação de quanto o produto vendeu exatamente no mesmo mês do ano anterior. E só ter duas informações dessas na base de treino não é o ideal, inclusive facilitando o *overfitting*, ou seja, quando a previsão tende a ficar muito similar com o que foi visto na base de treino.

Uma das maneiras de contornar esse problema foi utilizar a técnica de *Data Augmentation*. Para isso, criou-se dados novos de venda, a partir das informações de 2016 e 2017. O novo ano criado se baseou em uma mistura de dados desses anos em proporção aleatória, com adição de um ruído gaussiano aleatório para reduzir o *overfitting*. Essa técnica levou a um pequeno aumento na performance, mas pouco significativo no geral o ideal seria trabalhar com dados reais.

Para a criação do modelo, optou-se por utilizar o algoritmo do *XGBoost*, através da biblioteca homônima, implementado para Python. Por ser um algoritmo baseado em árvores de decisão, facilita o tratamento de dados, inclusive dispensando o escalamento dos mesmos. Além disso é fácil manipular os parâmetros, e obter um bom resultado desejado.

Assim, foi obtido um resultado satisfatório, superando facilmente a metodologia antiga do cliente de sempre se basear no ano anterior, assim como outras tradicionais (por exemplo fazer a média dos últimos anos). Isso é apresentado na Figura 7, que apresenta os resultados desses métodos na base de validação.

Figura 7: Erros em peças de roupas previstas de diferentes formas de previsões para base de validação (ano de 2018).



Fonte: Autoria Própria (2018).

Observando a figura, observa-se que o erro baseado em estimar de acordo com o ano passado seria superior a 5,5 peças de roupa por produto. Por exemplo, se um produto vendeu 30 peças estaríamos com uma estimativa, em média, entre 24,5 a 35,5. Com o algoritmo de previsão de venda, esse erro cai para próximo de 4, uma melhoria de aproximadamente 27%.

A integração desse módulo necessitou com que se avaliasse cuidadosamente os recursos consumidos pelo nosso algoritmo. Utilizar uma hospedagem especializada para aplicações de inteligência artificial e aprendizado de máquina aumentariam o custo do servidor, o que seria prejudicial para a empresa cliente.

Assim, havia a necessidade de se trabalhar com um plano simples de hospedagem, o que incluía apenas 512 MB de memória RAM e processamento compartilhado. Dependendo da aplicação de aprendizado de máquina, isso não é suficiente, principalmente se tratando volumes muito grandes de dados e de uma grande quantidade de colunas na análise, podendo causar erros de falta de memória, ou deixar o processo de treinamento do algoritmo mais lento.

A solução alternativa seria implementar essa funcionalidade localmente no computador do cliente, visto que o Electron também possui facilitadores para a

integração com módulos externos de linguagem Python. Ainda assim isso estaríamos limitados ao processamento do cliente, além de nos dificultar quanto às atualizações do sistema e a necessidade de implementar mais uma forma de integração.

No entanto, avaliando os recursos consumidos no treinamento em nossas primeiras hipóteses, conseguiu-se observar que era possível, nos nossos volumes de dados atuais, manter o algoritmo no servidor sem problema de memória e dentro do tempo esperado. Por conseguirmos uma previsão satisfatória com 2 anos de dados, e por só ter 4 lojas, conseguimos fazer o treinamento em menos de 5 minutos – o qual só precisará ser feito uma vez por mês, além de uma previsão quase instantânea. Ademais, consumimos uma quantidade baixa de memória RAM e estimamos que poderíamos trabalhar com cerca de mais 10 anos de dados antes de sofrer problemas relacionados ao servidor.

Portanto, esse módulo também foi alocado no servidor, e essa aplicação é ativada mediante uma requisição originada no Front-end, que é processada pelo módulo de controle e, visto que a tecnologia do deste último é também baseada na linguagem Python, nenhuma conversão de informação precisará ser feita, já que ambos compartilham das mesmas definições de encapsulamentos de dados.

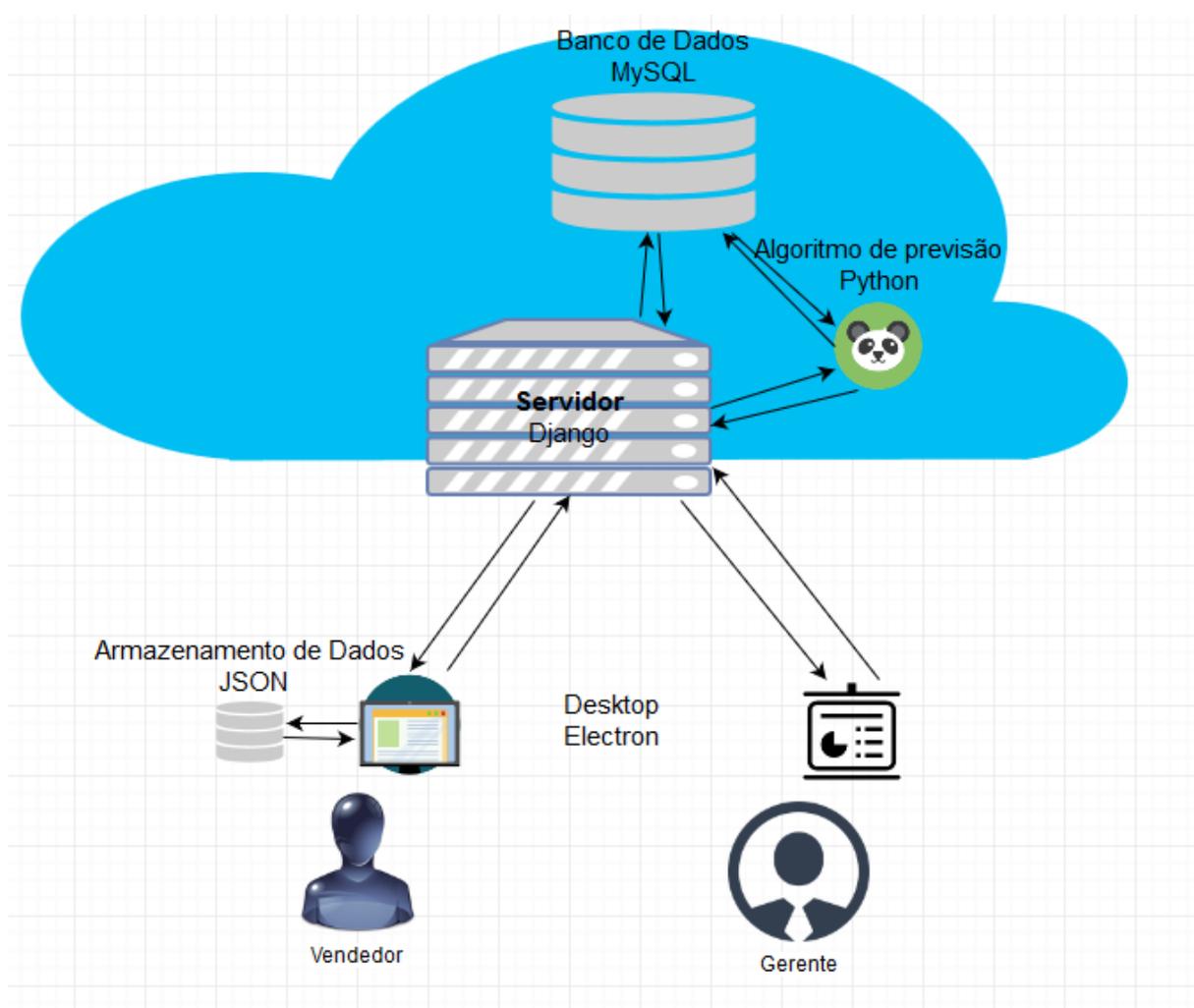
Para seu funcionamento, é necessário que o módulo de interface requisite a atuação do componente de aprendizado de máquina com os dados necessários para treinamento a partir da comunicação deste com a camada modelo. Ao final, o componente de aprendizado de máquina realizaria uma previsão baseado no seu treinamento e enviaria a informação de volta ao Front-end, que pode enviar para o usuário e auxiliar na sua tomada de decisão.

6.3 Integração dos Componentes

A figura 8 mostra como é a arquitetura específica do projeto. Um servidor centralizado realiza todas as operações com o banco de dados e também se conecta com um módulo de previsão de demanda utilizando inteligência artificial. Um usuário interage com o aplicativo *desktop* em Electron, e este realiza as requisições necessárias para o servidor em Django. O servidor faz as conexões com o banco de

dados em MySQL e se comunica de volta com o cliente. Há também a comunicação com o módulo de inteligência artificial em Python que é ativado pelo cliente e realiza a sua operação diretamente no banco de dados. Nota-se no também, que foi representado um armazenamento de dados local perto do vendedor, pois é um componente necessário para suprir o requisito não funcional da ininterruptão do sistema sem conexão à rede. Porém, não se julgou necessário ter esse armazenamento para o gerente, visto que suas funcionalidades não são de ação imediata obrigatória, ou seja, as operações do gerente não são urgentes como a de um vendedor, sendo possível esperar que a conexão volte ao normal.

Figura 8: Diagrama representando a arquitetura do projeto.



Fonte: Autoria Própria (2018).

6.4 Conclusão

Neste capítulo foi apresentado a escolha arquitetural para o projeto e as tecnologias envolvidas na solução. No capítulo seguinte será apresentado a fase de desenvolvimento do trabalho.

7 DESENVOLVIMENTO DO PROJETO

Para o desenvolvimento do projeto, primeiramente é explicado como foi o desenvolvimento do trabalho considerando o cronograma do apêndice A e o planejamento desenvolvido ao longo das seções anteriores. Posteriormente são explicitados e aprofundados os recursos do sistema resultados do desenvolvimento do mesmo.

7.1 O *Scrum* Aplicado ao Projeto

Uma vez estudado a metodologia *Scrum* e concluído que ela deveria ser aplicada no projeto, foi necessário estruturar a aplicação do mesmo ao projeto. No entanto, houveram desafios na implementação do método e logo foi constatada a diferença entre a prática e o que fora planejado.

O *Sprint planning*, a reunião realizada entre *Sprints*, foi o momento no qual se discutiu quais histórias de usuários e tarefas seriam realizadas no período seguinte, estimando o tempo ocupado por cada uma com *pontos de história*, explicado na seção 2.3.2. Nela também reestruturamos nosso Product Backlog, criando novas histórias e retirando ou reduzindo outras.

Um dos membros do grupo foi designado para ser o Product Owner. Ele tinha proximidade com o dono da empresa cliente e já havia familiaridade com o negócio, inclusive frequentando as lojas por um período. Isso fez com que as histórias do Product Backlog fossem, num geral, bem adequadas à realidade da empresa e não houve a necessidade de nenhuma grande reestruturação. Entretanto, a relação do Product Owner com o cliente e os usuários do produto foi majoritariamente informal: notamos futuramente que a falta de reuniões de negócio prejudicou o refinamento das histórias de usuário.

Nos mesmos encontros, já era feito o *Sprint review*, quando cada integrante da equipe avaliava o andamento do projeto nesse espaço de tempo, comentando os pontos positivos e negativos, e os últimos são debatidos de forma que se encontre o porquê destes problemas e maneiras de evitá-los.

De uma maneira geral, a metodologia foi bem aplicada no projeto, mas houveram algumas complicações. Primeiramente, como os integrantes da equipe tiveram disponibilidade variável e com agendas por muitas vezes divergentes, a organização de algumas *Sprints* foi comprometida, o que resultou em atrasos eventuais, *Sprint plannings* feitas de maneira apressada e eventos como a *Daily*, reunião de frequência diária em que se informa o que cada membro fará no dia, serem negligenciados.

Esses problemas de planejamento acarretaram no atraso da primeira *release*, prevista para o final de setembro após nove *Sprints*, sendo necessário mais um período para que se tivesse o MVP (*Minimum Viable Product*, o software com as funcionalidade mais básicas, mas que já pode ser aplicado). Esse fato fez com que a entrega fosse adiada para a metade de outubro.

Durante a primeira *release*, obtivemos feedbacks dos usuários, principalmente dos vendedores da loja, sobre algumas funcionalidades apresentadas. Assim, foi possível reestruturar algumas histórias de usuário para próxima *release*. Neste momento, pode-se concluir que a falta de encontro formais impossibilitou que algumas histórias de usuário fossem reestruturadas com antecedência, como citado acima.

A segunda *release* planejada para metade de novembro, teve que ser adiada para o final por causa do atraso anterior. A última inconsistência de planejamento que ocorreu foi o fato do requisito funcional de troca de produto planejado para este segundo lançamento, como citado na seção 5.1, não foi completado até o prazo final, ficando fora do escopo do projeto atual.

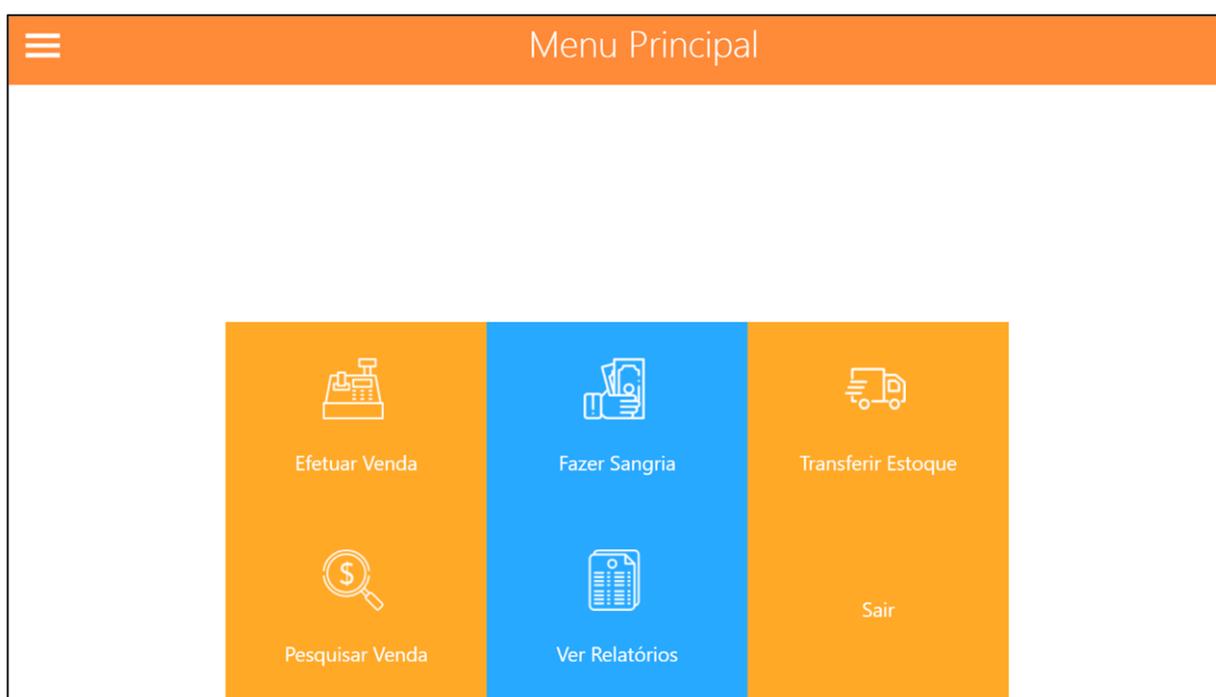
7.2 Sistema

No ponto de vista do negócio, pode-se separar o sistema em duas partes principais: a parte do vendedor, que encapsula todas as funcionalidades feitas pelos funcionários no dia a dia, e a parte gerencial, utilizada pelo dono da loja, que é responsável pela análise financeira, emissão de relatórios e controle de estoque, o que inclui o módulo previsão de vendas.

7.2.1 Funcionalidades do vendedor

Para esta parte do sistema, foram implementadas cinco funcionalidades principais da tela do vendedor, as quais podem ser visualizadas na figura 9.

Figura 9: Tela mostrada para o vendedor ao se autenticar.



Fonte: Autoria Própria (2018)

As telas relativas ao vendedor foram feitas com objetivo de serem simples, e de fácil entendimento para o vendedor. Isso foi necessário principalmente para a tela de venda, com o objetivo de facilitar e agilizar o processo de compra do produto enquanto o cliente está na loja.

Para uma venda, foi necessário implementar a consulta de clientes e produtos para serem adicionados à venda, e também a possibilidade de registrar novos consumidores. Pode-se também, optar por fazer uma encomenda – cujo pagamento será efetuado só futuramente na ocasião da retirada dos produtos, todos esses recursos podem ser verificados na figura 10.

Figura 10: Tela de adição de produtos da funcionalidade de venda.

2 - Michel Chieregato Gretsichschkin

Encomenda

ADICIONE O PRODUTO

CAMISETA COLEGIAL MASCULINO M

ID: 3173

1

Adicionar

PRODUTO

Cam

ID	Descricao	Tamanho	R\$
3165	CAMISETA COLEGIAL FEMININO	PP	43
3166	CAMISETA COLEGIAL FEMININO	P	43
3167	CAMISETA COLEGIAL FEMININO	M	43
3168	CAMISETA COLEGIAL FEMININO	G	43
3169	CAMISETA COLEGIAL FEMININO	GG	43
3170	CAMISETA COLEGIAL FEMININO	ESP	43

VENDAS

ID	Descricao	Tamanho	Qnt	R\$
3006	BERMUDA CICLISTA COLEGIAL	PP	3	210.00
3173	CAMISETA COLEGIAL MASCULINO M	M	1	46.00
Total:				256.00

Finaliza venda

Fonte: Autoria Própria (2018).

Após adicionar os produtos e os clientes, a figura 11 mostra o próximo passo, que é a finalização da transação com o pagamento. Para este recurso, é possível que o comprador escolha um dos cinco métodos de pagamento e o valor daquele escolhido. O troco é calculado automaticamente, cabendo ao vendedor finalizar a compra ou cancelá-la no caso de informações inconsistentes. Especificamente a figura 11 mostra uma venda com desconto, por isso o valor original riscado e o valor descontado logo ao lado.

Figura 11: Tela de pagamento da funcionalidade de venda.



R\$ ~~256.00~~ 230.4

Dar desconto


 Dinheiro


 Cheque


 Débito


 Crédito


 Transferência

PAGAMENTOS		
	Forma de pagamento	R\$
1	Dinheiro	250.00
Total:		250.00

Recebido: R\$250.00

Falta Receber: R\$0.00

Troco: R\$19.60


 Voltar a tela


 Finaliza venda

Fonte: Autoria Própria (2018).

Ao finalizar uma venda ou encomenda, ela será, a princípio, registrada no banco de dados, via comunicação com o servidor Django. Porém, devido ao requisito não funcional apresentado na seção 4.2, era necessário que as vendas fossem feitas mesmo sem conexão com a internet.

Dessa forma, foi desenvolvida uma solução para que a venda continuasse a ser feita mesmo sem conexão com a internet e manter ao mesmo tempo os bancos atualizados, devido a comunicação entre lojas.

As informações necessárias para as vendas são os produtos, os clientes e os usuários do sistema que podem realizá-la. Assim, optou-se por salvar essas opções localmente no computador do usuário, via JSON. Como essas informações são sempre modificadas, o sistema irá, sempre que for reiniciado, verificar se há diferença entre o arquivo local e o banco de dados remoto e, se houver, atualizar os arquivos JSONs.

No momento de finalização da venda, o sistema detecta automaticamente caso o programa esteja sem conexão à rede, utilizando ferramentas da linguagem JavaScript. Caso esteja, a venda será registrada em um arquivo local no padrão JSON. Clientes também podem ser registrados dessa maneira.

Na próxima conexão com o sistema, caso haja internet, as vendas e os clientes adicionados offline serão, um a um, adicionados ao banco de dados remoto central. Essa requisição HTTP é feita através do processo principal do Electron denominada ipcMain, não afetando em nada a usabilidade das telas do restante do programa. A cada venda registrada com sucesso, ela é apagada do arquivo local.

As outras funcionalidades mostradas na figura 5 ficam indisponíveis nesse estado sem conexão.

As vendas realizadas podem ser acessadas a partir da tela de pesquisar venda. Ela é mostrada na figura 12.

Figura 12: Tela de pesquisar venda.

C	Data	CPF	Cliente	Valor	Encomenda
81	2018-12-02 20:27	275.239.958-88	Flavia Micheletto	98.00	✓
80	2018-12-02 20:26	408.364.748-56	Michel Chierigato Gretschischkin	256.00	✗
79	2018-12-02 20:26	408.364.748-56	Michel Chierigato Gretschischkin	230.40	✗

Fonte: Autoria Própria (2018).

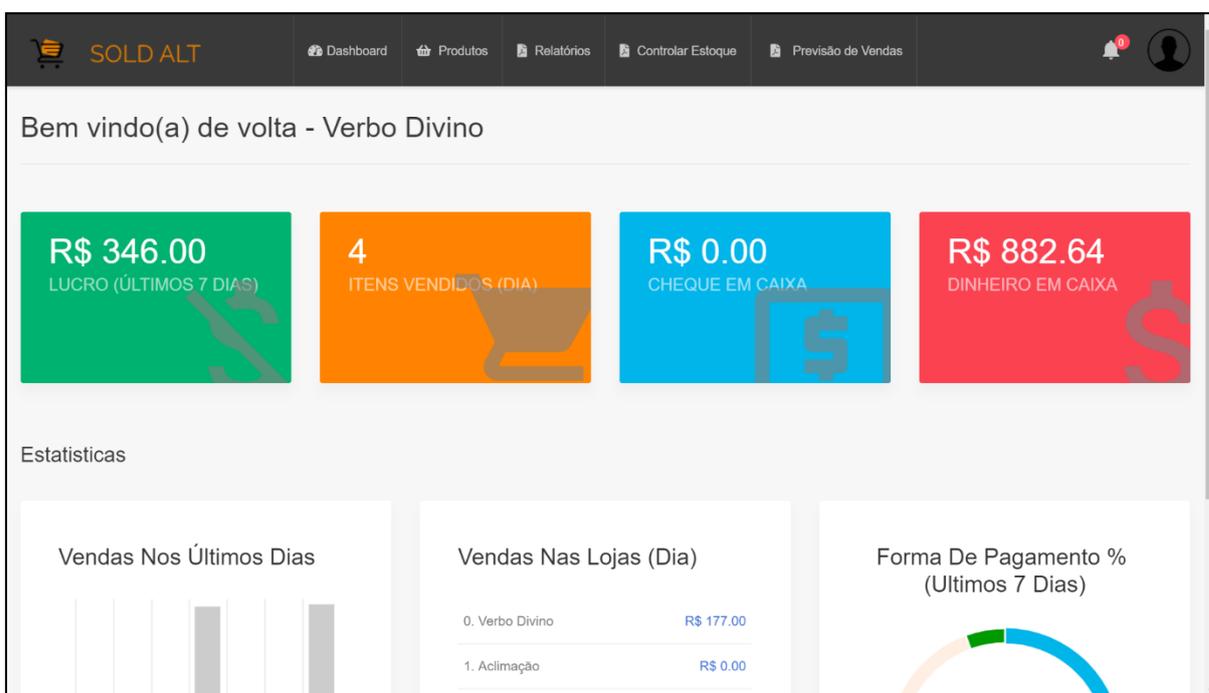
Ao acessar cada venda, pode-se observar os produtos vendidos, finalizar a venda, caso ela fosse uma encomenda, e realizar uma troca de mercadoria. No entanto, devido a problemas no desenvolvimento do projeto como explicado na seção 7.1, a funcionalidade de troca não foi implementada.

Além disso, é possível verificar o caixa da loja, através da tela de sangria, transferir estoque entre as unidades (incluindo também o recebimento de novas mercadorias pela loja) e gerar relatórios das vendas do dia.

7.2.2 Funcionalidades do administrador

As funcionalidades do administrador têm como objetivo fazer a análise de dados financeiros, gerar relatórios e facilitar o controle do estoque por parte do gerente da loja. Na figura 13, pode-se observar a tela inicial do administrador. Essa tela resume as principais informações pelo ponto de vista do administrador como lucro da semana, itens vendidos no dia, quantidade de dinheiro e cheque em caixa. Além disso, apresenta gráficos para acompanhar as vendas ao longo da semana, como estão sendo distribuídas as formas de pagamento e uma tabela que apresenta as vendas diárias em todas as lojas, da maior para a menor.

Figura 13: Tela apresentada ao administrador após autenticação.



Fonte: Autoria Própria (2018).

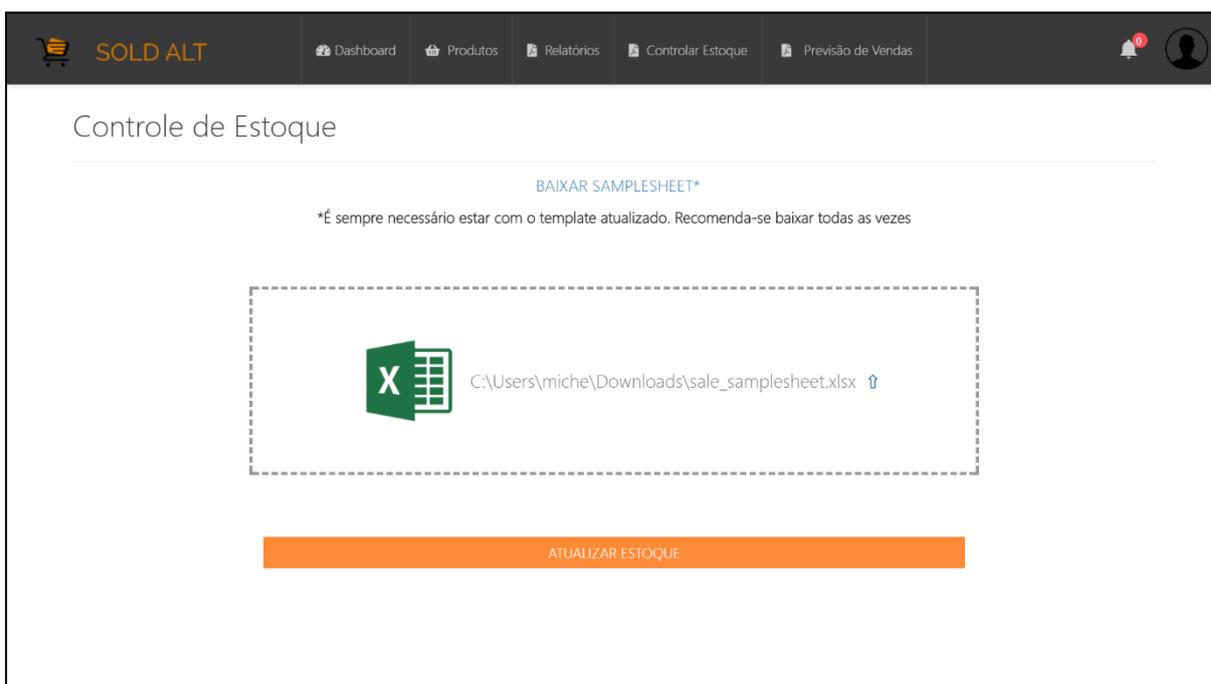
Dentro das outras funcionalidades, há a criação e modificação do registro dos produtos cadastrados, geração de relatórios de vendas, controle de estoque e o módulo de previsão de vendas.

A forma de controle e contagem de estoque foi especialmente pensada e adaptada para melhor usabilidade do cliente. Havia uma queixa em relação ao sistema antigo da loja, devido a dificuldade de mexer em longas tabelas dentro do próprio sistema, o que dificulta na logística de revisão e recontagem por outras pessoas, levando muitas vezes a atrasar a atualização de estoque.

Para resolver isso, adaptou-se uma solução de acordo com a demanda do cliente, fez-se com que o sistema lesse uma planilha de Excel similar à utilizada pelo dono da loja para fazer o controle interno dele. Assim, sempre que o usuário quiser fazer uma contagem ele baixa uma planilha com um padrão fixo, preenche e faz o upload.

O *Back-end* irá assegurar que o modelo foi seguido corretamente e atualizar o estoque com base na planilha. A biblioteca *pandas*, do Python, facilitou essa integração no processo de desenvolvimento, tornando uma funcionalidade simples e segura. Um exemplo dessa tela pode ser visto na figura 14.

Figura 14: Tela de controle de estoque do administrador.



Fonte: Aatoria Própria (2018).

O módulo de inteligência artificial, cujo desenvolvimento é detalhado na seção 6.2.3, foi integrado de maneira simples na parte gerencial, como pode ser visto na figura 15, na qual única necessidade de interação do usuário é selecionar caso queira retreinar o algoritmo através de um botão de gerar relatório. O resto será feito automaticamente pela aplicação Django. Uma vez treinado, o administrador pode abrir sempre um arquivo pdf com a previsão do mês seguinte, como consta na figura 16.

Figura 15: Tela de previsão de vendas.



Fonte: Autoria Própria (2018).

Figura 16: Trecho de relatório de previsão de vendas por produto.

Vendas do mês de dezembro			
Produto	Tamanho	Vendas em Verbo Divino	total
BERMUDA CICLISTA COLEGIAL	G	0	0
BERMUDA CICLISTA COLEGIAL	GG	1	1
BERMUDA CICLISTA COLEGIAL	M	2	2
BERMUDA CICLISTA COLEGIAL	P	1	1
BERMUDA CICLISTA COLEGIAL	PP	3	3
BERMUDA MOLETOM LISO	06	2	2
BERMUDA MOLETOM LISO	GG	0	0
BLUSAO COLEGIAL COM FLANELA	M	2	2

Fonte: Autoria Própria (2018).

8 TESTES E AVALIAÇÃO

O critério de pronto adotado pela equipe para se considerar uma *história de usuário* terminada envolve testar a funcionalidade para cada tarefa realizada. Desta forma, a rotina de testes e o formato dos mesmos foram essenciais para o desenvolvimento do projeto.

8.1 Teste de Garantia de Qualidade

O QA (*Quality Assurance*, ou garantia de qualidade, do inglês) foi uma das etapas da conclusão de tarefas aplicada em todas as *sprints*.

Uma vez que um serviço fosse finalizado por um membro da equipe, um segundo indivíduo ficaria encarregado do QA, testando a funcionalidade desenvolvida no sistema como se fosse um usuário, porém de maneira minuciosa simulando todos os cenários possíveis, para que se garanta que nenhuma hipótese foi descartada.

Após os testes da tarefa feita, as falhas encontradas ou melhorias sugeridas se tornam novas tarefas que serão realizadas e testadas, e somente ao não existir falhas nos testes da tarefa, declarar-se-ia que o serviço foi terminado com êxito.

8.2 Teste de Sistema

Quando todas as histórias de usuário previstas para qualquer uma das *releases* foram feitas e devidamente testadas, começaram-se os testes de sistema.

Apesar de toda tarefa ter passado pela etapa de QA, o que significa que todas as funcionalidades novas devem estar conforme o esperado isoladamente, os testes de sistema são necessários pois avaliam o software em uma escala maior, isto é, não somente os processos desenvolvidos separadamente, mas a interação entre eles. Eles são essenciais para garantir que o fluxo de atividades no sistema está adequado, e que a execução de uma atividade não impactou inesperadamente em outra.

Os testes de sistema contam com todos os fluxos que o programa possibilita, o que inclui a realização de uma venda ou encomenda (com ou sem cliente, com diversos modos de pagamento, com ou sem desconto, etc.); a geração de relatórios; a consistência das telas de análise de venda; a realização da sangria; o acesso a vendas passadas; a finalização de encomendas; dentre outras funcionalidades.

Uma vez feitos os testes e corrigidos os eventuais erros, o sistema está completamente funcional.

8.3 Teste de Aceitação

Após todos os testes em ambiente de desenvolvimento, o sistema novo é implantado na loja e após algum tempo de uso é feito o mais importante teste de todos, o do usuário.

Mais importante que testes que asseguram a funcionalidade do produto em si, o UAT (*User Acceptance Testing*) ou teste de aceitação do usuário é feito diretamente com o usuário do sistema para conferir que ele atende as expectativas do cliente e exercem sua função da maneira mais adequada possível.

Nas duas *releases* feitas, foram realizadas visitas à loja para entrevistar os funcionários acerca da relação dos mesmos com o sistema. A recepção da aplicação em ambas as ocasiões foi muito positiva, pois sugeriram melhorias das funcionalidades existentes e apontaram algumas falhas.

Ao instalar o sistema na loja foi observado algumas falhas em telas que tiveram menor tempo de teste dedicado. Além disso, apesar do *feedback* positivo por parte do cliente, foram apontadas algumas pequenas melhorias no relatório de vendas. Alguns desses pontos foram consertados no mesmo momento, outros foram encaminhados para a *sprint* seguinte.

Tanto os funcionários quanto o gerente da loja elogiaram bastante o sistema, enfatizando que ele é fácil de usar, pouco burocrático e supre as necessidades do dia-a-dia do estabelecimento.

Além das visitas às lojas, conversas com o administrador de empresa foram frequentes, e nelas foram possíveis assegurar a qualidade do produto e coletar as próximas funcionalidades necessárias.

Todos os pedidos dos usuários e as falhas encontradas são escritos em formato de histórias de usuário e são planejados para serem implementados em *sprints* futuras.

8.4 Avaliação Final

O sistema está funcionando na loja há aproximadamente um mês e meio e os resultados gerais foram bem positivos. Os funcionários afirmaram que o programa é fácil de se usar e bem pouco burocrático, e o gestor da loja disse que a parte da análise das vendas e de predição está muito boa e será muito útil no ano seguinte, quando haverá um grande volume de transações devido à volta às aulas.

Houveram poucas falhas do sistema relatadas pelos usuários, o que indica uma razoável robustez na rotina de testes. Em contrapartida, o ânimo com o novo software fez com que os funcionários da loja sugerissem funcionalidades que podem ser implementadas em etapas futuras.

9 CONCLUSÕES

O projeto de desenvolvimento de um sistema de informação real aplicado em uma empresa de fabricação e comercialização de uniformes escolares teve seu desenvolvimento completado graças aos estudos sobre os assuntos pertinentes do projeto incluindo uma pesquisa aprofundada da empresa cliente e do contexto do mercado de sistemas de gestão empresarial, à utilização da metodologia ágil e técnicas da mesma para especificar requisitos do projeto e determinar o ciclo de trabalho para a conclusão do projeto, à decisão da equipe na arquitetura e tecnologias envolvidas no projeto, ao desenvolvimento seguindo todos os elementos explicitados anteriormente, e por fim a uma definição da validação das tarefas dos ciclos de trabalho por meio de testes.

O trabalho atingiu seu objetivo principal, foi implementado com sucesso o seu sistema de vendas e controle de estoque utilizando inteligência artificial para a empresa de pequeno porte que fabrica e comercializa uniformes escolares desde o dia nove de outubro de 2018. O sistema está atualmente auxiliando os funcionários em suas atividades operacionais e a administração do negócio no controle de estoque a partir da previsão das vendas mensais.

9.1 Trabalhos Futuros

Para melhorar o software desenvolvido e continuar a auxiliar a empresa cliente a aprimorar sua eficiência, alguns trabalhos sobre o sistema ainda podem ser desenvolvidos.

Inicialmente, deve-se terminar a implementação das histórias de usuário de menor prioridades não finalizadas, como a troca de mercadoria. Ademais, pode-se aumentar a quantidade de soluções suportadas sem a necessidade de conexão à internet, ampliando o requisito não funcional e dando maior flexibilidade ao sistema. Por fim, deve-se continuar a coleta de dados, ampliando a base histórica, além da possibilidade de se obter outras informações e conseqüentemente melhorar a previsão de estoque, refinando o algoritmo desenvolvido.

Além disso, pode-se pensar futuramente em como generalizar esse software para atender diversas demandas e setores. Com pequenas modificações é possível ampliar o benefício atual dado a empresa cliente e assim conseguir auxiliar potenciais novos clientes de diferentes segmentos de mercados.

No entanto, antes de seguir a ambição de aumentar o escopo de clientes do projeto, será necessário um aprofundamento maior na área técnica para tornar o desenvolvimento do sistema mais eficaz e ágil, e principalmente nas áreas de finanças e estratégia de negócios para determinar um modelo de negócio e um plano sustentável para tornar o projeto um sistema comercializável.

9.2 Contribuições do Trabalho

Para a empresa cliente, o sistema se tornou uma ferramenta valiosa para suas operações e tomadas de decisão, o que pode, diretamente e indiretamente, ser convertido em maiores lucros.

O projeto foi desenvolvido em um site que hospeda projetos de software gratuitamente e pode ser encontrado a partir do seguinte endereço de internet: https://github.com/gkhonda/electron_sold_alt. O conteúdo do site em questão contém uma pequena introdução do sistema e o endereço dos outros componentes do projeto, desenvolvidos em páginas de internet separadas.

Este projeto levou a um aprendizado extremamente importante para os membros do grupo. O seu tema revelou para os integrantes, por exemplo, que um sistema de vendas aparentemente simples na visão de um usuário é bem mais complexo na visão de desenvolvedor, visto que o mesmo precisa considerar as tecnologias envolvidas e os contextos de negócio.

A estratégia de metodologia do projeto também se mostrou valiosa ao ensinar o valor não só de se especificar e planejar o projeto minuciosamente, como também todo o procedimento do processo de desenvolvimento.

9.3 Conclusão Final

Por fim, espera-se que este trabalho de conclusão de curso da Escola Politécnica da USP contribua para o meio acadêmico da mesma maneira que contribuiu para a empresa privada e para o aprendizado dos integrantes da equipe de projeto.

REFERÊNCIAS

BOURQUE, PIERRE; FAIRLEY, RICHARD E. **Guide to the Software Engineering Body of Knowledge**. 3ª. Ed. IEEE Computer Society, 2014. Disponível em <www.swebok.org>. Acesso em 12 de nov. de 2018.

BREIMAN, L. **Random Forests**. 2001. 33f. Universidade da Califórnia em Berkeley. 2001.

CHEN, TIANQI; GUESTRIN, CARLOS. **XGBoost: A scalable tree boosting system**. 2016. 10f. Universidade de Washington. 2016.

COLE, ANIELA. **Avaliação de Aderência de Práticas do Agile Testing em Times que Utilizam Scrum**. Monografia – UNISINOS. São Paulo, p. 47. 2015.

DANTAS, RODRIGO. Totvs Protheus: como funciona?. **Vindi**, 11 jun. 2015. Disponível em <<https://blog.vindi.com.br/totvs-protheus-como-funciona/>>. Acesso em: 02 nov. 2018.

DAVIS, FRED D. VENKATESH, VISWANATH. **Theoretical acceptance extension model: field four studies of the technology longitudinal**. Em: Management Science, v.46, n.2, pp. 186-204, fev. 2000.

DJANGO SOFTWARE FOUNDATION. **Django**. Disponível em: <<https://www.djangoproject.com/>>. Acesso em: 13 de nov. de 2018.

EQUIPE BENNER. Benefícios do uso da Inteligência Artificial na gestão jurídica. **Benner**, 5 abr. 2018. Disponível em: <<https://blog.benner.com.br/beneficios-do-uso-da-inteligencia-artificial-na-gestao-juridica>>. Acesso em: 02 de nov. de 2018.

GITHUB. **Electron**. Disponível em: <<https://electronjs.org/>>. Acesso em: 13 de nov. de 2018.

GRUPO BENNER. **Produto de sistema ERP da empresa Benner**. Disponível em: <<https://www4.benner.com.br/erp>>. Acesso em: 28 de out. de 2018.

LAW, C.; NGAI, E. **ERP systems adoption: an exploratory study of the organizational factors and impacts of ERP success**. Em: Information and Management, v.44, n.4, pp. 418-432, jun. 2007.

MENDES, JULIANA VEIGA; ESCRIVÃO FILHO, EDMUNDO. **Atualização tecnológica em pequenas e médias empresas: proposta de roteiro para aquisição de sistemas integrados de gestão (ERP)**. Em: Gestão & Produção, v.14, n.2, pp. 281-293, 2007.

MESTRE DIGITAL. **Produto de sistema ERP da empresa Mestre Digital**. Disponível em: <<http://www.mestredigital.com.br/index.html>>. Acesso em: 28 de out. de 2018.

MITCHELL, T.M. **Machine Learning**. 1ª. Ed. McGraw-Hill Science/Engineering/Math, 1997.

MORESI, EDUARDO A. D. **Delineando o valor do sistema de informação de uma organização**. Em: Ciência da Informação, v.29, n.1, pp. 14-24, abr. 2000. Disponível em: <<http://www.scielo.br/scielo.php?pid=S0100-19652000000100002>>. Acesso em 7 de out. de 2018.

ORACLE. **MySQL**. Disponível em <<https://www.mysql.com/>>. Acesso em: 18 de nov. de 2018.

PROTASIEWICZ, JAKUB. Why is Python So Good for AI, Machine Learning and Deep Learning? **Netguru**, 31 ago. 2018. Disponível em <<https://www.netguru.co/blog/why-is-python-so-good-for-ai-machine-learning-and-deep-learning>>. Acesso em: 18 de nov. de 2018.

PSOINAS, A.; KERN, T.; SMITHSON, S. **An exploratory study of information systems in support of employee empowerment**. Em: Journal of Information Technology, v.15, n.3, pp. 211-230, dez. 2010.

PYTHON SOFTWARE FOUNDATION. **Python**. Disponível em: <<https://www.python.org/>>. Acesso em: 13 de nov. de 2018.

RAJAN, CHRISTY A.; BARAL, RUPASHREE. **Adoption of ERP system: An empirical study of factors influencing the usage of ERP and its impact on end user**. Em: IIMB Management Review, v.27, n.2, pp.105-117, jun. 2015.

REFSNES DATA. JSON – Introduction. **W3Schools**, 2018. Disponível em <https://www.w3schools.com/js/js_json_intro.asp>. Acesso em: 17 de nov. de 2018.

RUBIN, KENNETH S. **Essential Scrum**. Pearson Education Inc., 2013. 498 p.

RUSSEL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3ª. Ed. Pearson, 2009.

STAIR, RALPH M.; REYNOLDS, W. GEORGE. **Princípios de Sistemas de Informação**. Tradução de Flávio Soares Corrêa da Silva. 9ª. Ed. Cengage Learning, 2011. 590 p.

SUTHERLAND, JEFF; SCHWABER, KEN. **The Scrum Guide**. Scrum Guides, nov. 2017. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>>. Acesso em 8 de out. de 2018.

TOTVS. **Produto de sistema ERP da empresa TOTVS**. Disponível em: <<https://www.totvs.com/erp/>>. Acesso em: 28 de out. de 2018.

WINSTON, P.H. **Artificial Intelligence**. 3ª. Ed. Addison-Wesley. 1992.

APÊNDICE A - CRONOGRAMA INICIAL DO PROJETO

