



LUI FILGUEIRAS TELES

**ARQUITETURA DE SOFTWARE PARA UMA
SOLUÇÃO DE BILLING E CRM:
APLICAÇÃO DA PLATAFORMA SYDLE ONE
PARA UMA SOLUÇÃO DE BILLING E CRM.**

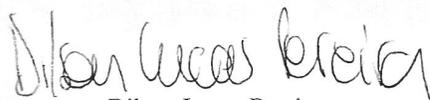
LAVRAS – MG

2019

LUI FILGUEIRAS TELES

**ARQUITETURA DE SOFTWARE PARA UMA SOLUÇÃO DE BILLING
E CRM:
APLICAÇÃO DA PLATAFORMA SYDLE ONE PARA UMA SOLUÇÃO DE
BILLING E CRM.**

Relatório de estágio supervisionado apresentado à
Universidade Federal de Lavras, como parte das
exigências do Curso de Sistemas de Informação,
para a obtenção do título de Bacharel.



Dilson Lucas Pereira

Orientador

LAVRAS – MG

2019

Dedido este trabalho à parentes e amigos.

AGRADECIMENTOS

Agradeço ao apoio de amigos e familiares.

RESUMO

O presente trabalho tem como objetivo apresentar as atividades que foram desenvolvidas durante o período de estágio na empresa SYDLE Sistemas Ltda, sediada na cidade de Belo Horizonte - MG. O sistema foi desenvolvido por meio da plataforma SYDLE ONE, e utiliza conceitos de Billing e CRM (Customer Relationship Management) para otimizar a gestão do faturamento e do relacionamento com o cliente para uma empresa de análise de crédito de mercado. Para o desenvolvimento desse sistema, foram utilizadas as ferramentas JavaScript, Angular JS, Git, MongoDB, BPM (Business Process Management) e BPMN (Business Process Modeling Notation), HTML e ElasticSearch, e a metodologia ágil SCRUM.

Palavras-chave: Estágio. SYDLE ONE. Billing. CRM. Gestão.

ABSTRACT

The following assignment aims to report the activities that were developed during the internship period at the company SYDLE Sistemas Ltda, headquartered in Belo Horizonte - MG. The system was developed using the SYDLE ONE platform and utilizes the concept of Billing and Customer Relationship Management (CRM) to optimize the running of the income and the customer relationship in a company that analyses the market credit. To develop this system, were used JavaScript, Angular JS, Git, MongoDB, Business Process Management (BPM) and Business Process Modeling Notation (BPMN), HTML and ElasticSearch, and the agile SCRUM methodology.

Keywords: *Internship. SYDLE ONE. Billing. CRM. Management.*

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Sistema utilizado para o desenvolvimento	12
1.2	Projeto	12
1.2.1	Empresa Cliente	12
1.2.2	Solução de Billing	13
1.2.3	Solução de CRM	13
1.3	Processo de desenvolvimento	13
2	REFERENCIAL TEÓRICO	15
2.1	HTML	15
2.2	JavaScript	15
2.3	AngularJS	16
2.4	Git	17
2.5	MongoDB	18
2.6	ElasticSearch	19
2.7	BPM	20
2.7.1	BPMN	21
2.7.2	BPMS	22
2.8	SCRUM	23
3	METODOLOGIA	25
3.1	Procedimentos metodológicos	25
3.2	Processo de desenvolvimento	25
4	ATIVIDADES DESENVOLVIDAS	27
4.1	Alteração de plano de assinatura	27
4.2	Atendimento via CNPJ	28
4.3	Envio de e-mails automáticos	29
4.4	Processo de validação por OCR	30
4.5	Processo de suspensão de pedidos	30

5	DESCRIÇÃO E CRONOGRAMA DE ATIVIDADES	33
6	CONSIDERAÇÕES FINAIS	35
6.1	Disciplinas importantes para o desenvolvimento do projeto . . .	35
6.2	Conclusão	36
7	REFERÊNCIAS BIBLIOGRÁFICAS	37

1 INTRODUÇÃO

A SYDLE Sistemas LTDA é uma empresa fundada no ano de 2006, que tem sede em Belo Horizonte, Minas Gerais. A empresa conta, atualmente, com cerca de 80 funcionários, em sua sede, na capital mineira, e nas cidades de São Paulo e Rio de Janeiro, possuindo clientes em mais de 80 países. O produto principal da empresa, no Brasil e no exterior, é uma plataforma digital corporativa chamada SYDLE ONE, que possui arquitetura flexível a outros sistemas digitais. Isto permite integrar as principais soluções corporativas no modelo “*all-in-one*” ou “tudo em um”, englobando diversos conceitos em uma única plataforma, com o objetivo de concretizar a transformação digital das empresas.

Durante a realização do estágio, o time em que o estagiário se encontra realizou o desenvolvimento de um sistema de ecommerce para uma empresa de análise de crédito de mercado, trazendo soluções de CRM (“*Customer Relationship Management*” ou “Gestão do Relacionamento com o Cliente”) e Billing (sistema de cobrança responsável pela geração de controle de faturas, de forma automática) para a empresa cliente.

O sistema foi desenvolvido através da plataforma SYDLE ONE, sendo utilizadas as seguintes ferramentas: linguagem de programação JavaScript, framework Angular JS para o Frontend, sistema de controle de versões Git, banco de dados No SQL MongoDB, modelagem e automatização de processos BPM (“*Business Process Management*” ou “Gestão de Processo de Negócios”) e BPMN (“*Business Process Modeling Notation*” ou “Notação de Processos de Modelagem de Negócios”), metodologia ágil SCRUM, linguagem de formatação HTML, e a ferramenta elasticSearch para a realização de buscas rápidas por meio da indexação dos dados. Esse sistema proporciona uma solução completa de gestão de faturamento de produtos, serviços e assinaturas, além de uma otimização na gestão de relacionamento com o cliente. Com isso espera-se que a organização à qual o ser-

viço foi sendo prestado tenha uma solução tecnológica para obter mais controle de seus dados e qualidade em seu serviço.

1.1 Sistema utilizado para o desenvolvimento

O projeto foi desenvolvido por meio da plataforma SYDLE ONE. Essa plataforma digital utiliza conceitos como BPM, ECM (“*Enterprise Content Management*” ou “Gestão de conteúdo corporativo”), CRM e *Analytics* (ou Inteligência Analítica) para solucionar os problemas das empresas para as quais a SYDLE presta serviço.

A utilização dessa plataforma permite:

- Automatização, instanciação e acompanhamento de todo o ciclo de vida dos processos.
- Possibilidade de ter uma gestão à vista de toda empresa/área/processo.
- Visão temporal e colaborativa dos seus dados.
- Rastreabilidade necessária dos dados.
- Gestão completa de todos os documentos da organização (GED).
- Modelagem e documentação todos os processos gerando ativo organizacional (*as is, to be*).

1.2 Projeto

1.2.1 Empresa Cliente

A empresa cliente é uma empresa de análise de crédito de mercado que oferece soluções de crédito, marketing, certificação e consulta de dados para companhias de todos os segmentos, e que possui indicadores econômicos que são referência tanto para o comércio e a indústria, como para o setor de serviços.

O sistema que foi desenvolvido deve servir como uma solução de *Billing* e CRM para a empresa, contendo diversos tipos de processos, atribuindo ao *software* funcionalidades como: cadastro de uma pessoa ou de uma empresa, venda de um produto, cobrança de serviços, suspensão ou retenção de contratos e aplicação de descontos por determinados períodos de tempo. Alguns exemplos de processos desenvolvidos são: atendimento, negociação, faturamento, venda e arrecadação.

1.2.2 Solução de Billing

Billing é um sistema de cobrança voltado para gestão financeira, que é capaz de gerar e controlar faturas de forma automática. Sistemas de *Billing* são capazes de executar ações como: gestão de contratos, controle de pagamentos de serviços e aplicação de ações programadas a inadimplentes e cobrança de diversos valores, referente a produtos, assinaturas e serviços variados com precificações variadas.

1.2.3 Solução de CRM

CRM é uma ferramenta voltada para o relacionamento com o cliente, sendo utilizada como uma prática para gerir os clientes e conquistar novos, utilizando informações de experiências passadas. Com isso, auxilia as empresas na personalização de seu atendimento para obter um melhor atendimento e se tornarem mais competitivas.

É importante que o CRM seja aplicado de forma integrada a todas as áreas da empresa, mesmo que esse processo de integração ocorra em etapas, para que todas as informações sobre os clientes estejam centralizadas.

1.3 Processo de desenvolvimento

Arquitetura de Software é o processo de definição de uma solução estruturada que atende a todos os requisitos técnicos e operacionais e ao mesmo tempo

otimiza atributos de qualidade padronizados como desempenho, segurança e gerenciamento. Arquitetura envolve uma série de decisões baseadas em uma vasta gama de fatores e cada uma destas decisões pode provocar um impacto considerável no sucesso ou fracasso da aplicação. (*Microsoft Application Architecture Guide*, 2ª edição, 2009).

Nesse contexto o papel do estagiário no projeto foi na arquitetura do software, atuando no desenvolvimento do sistema, realizando as histórias atribuídas em cada Sprint, e atuando na melhoria de funcionalidades existentes, na criação de novas funcionalidades e na correção de bugs encontrados no sistema.

2 REFERENCIAL TEÓRICO

Neste capítulo, serão abordados os conceitos e as tecnologias mais importantes utilizadas tanto pela empresa quanto para o desenvolvimento do projeto.

2.1 HTML

HTML é uma linguagem de marcação utilizada na construção de páginas na Web. HTML é a sigla em inglês para *HyperText Markup Language*, que significa Linguagem de Marcação de Hipertexto.

Segundo Silva (2008), hipertexto pode ser definido como um conjunto de conteúdos inseridos em um documento para a Web, sendo que tais conteúdos se ligam a outros conteúdos por meio de conexões presentes na Web. Silva (2008) ressalta que essas conexões são possíveis devido à utilização dos links presentes nas páginas Web, responsáveis pela ligação entre os conteúdos.

De acordo com Caldeira (2015) o HTML é um conjunto de instruções que são interpretadas pelo browser, definindo como se deve publicar uma página. O HTML possui quatro conceitos fundamentais: os elementos são contentores que definem seções presentes dentro de uma página Web; as *tags* (ou etiquetas) são os demarcadores que definem o início e o final de cada seção, sendo que cada elemento é composto por duas *tags*; os atributos são colocados na *tag* de início de um elemento, e são utilizados para definir características desse elemento; e o aninhamento define regras de não sobreposição de elementos, para mantê-los independentes.

2.2 JavaScript

JavaScript é uma linguagem de programação muito utilizada no desenvolvimento para Web, sendo utilizada pelo lado do *front-end* para atribuir funcionalidades às telas e para manipulá-las. Com essa linguagem, é possível dar dinamismo

a sistemas, que com apenas a utilização de HTML e CSS, são considerados “estáticos”.

De acordo com Silva (2010), essa linguagem foi criada por meio de uma parceria entre a Sun Microsystems e a Netscape, tendo sua primeira versão, chamada JavaScript 1.0, lançada no ano de 1995. Ainda segundo Silva (2010), o HTML limita-se à criação de campos e rótulos, não sendo possível portanto o processamento ou a transferência de dados. Com isso, o objetivo do JavaScript é a adição de dinamismo e interatividade a páginas Web, potencializando o poder do HTML.

Flanagan (2004) afirma que JavaScript faz parte de uma tríade usada para desenvolvimento Web, sendo o HTML responsável pelo conteúdo presente na página, o CSS responsável pela apresentação desse conteúdo e o JavaScript responsável pelo comportamento dos conteúdos da página. Ainda de acordo com Flanagan (2004), JavaScript é uma linguagem de alto nível, além de ser também considerada dinâmica, interpretada e não tipada, o que a torna propícia para estilos de programação orientados a objetos e funcionais.

O JavaScript pode ser usado para melhorar a interação dos usuários com as páginas Web. Prescott (2016) apresenta outras funcionalidades para essa linguagem, afirmando que ela pode ser usada no controle de um navegador, na comunicação com o servidor, na alteração dinâmica de determinado conteúdo de um Website e para o desenvolvimento de jogos e aplicações.

2.3 AngularJS

AngularJs é um framework baseado em JavaScript, usado para criar aplicações web de forma dinâmica. De acordo com dos Santos e Rufino (2016), o objetivo do AngularJs é servir como uma extensão do HTML, tornando-o mais poderoso, e com isso, facilitando o desenvolvimento de telas agradáveis e funcionais para os usuários.

Branas (2014) pontua como uma principais características do AngularJS a utilização de diretivas para levar ao HTML parte da lógica das funcionalidades dos componentes. Branas (2014) também define diretivas como extensões do HTML que possibilitam a adição de comportamentos novos de forma declarativa.

Segundo Pereira (2014), o Angular apresenta características como: desempenho, produtividade, fácil customização e integração com ferramentas e frameworks JavaScript, suporta desenvolvimento de módulos, é testável e plugável, e implementa o conceito de diretivas, ampliando o vocabulário HTML.

Essa ferramenta foi desenvolvida e é mantida por engenheiros da Google. O início de seu desenvolvido foi em 2009, e sua primeira versão foi liberada em 2012.

2.4 Git

Git é um sistema de controle de versões distribuído, que é muito usado no desenvolvimento de *software*. Esse sistema foi desenvolvido por Linus Torvalds (o criador do Linux).

A utilização de um sistema de controle de versão permite um maior controle sobre o desenvolvimento de uma aplicação. Com o uso desse sistema pode-se alterar um código, acompanhar outras alterações realizadas no mesmo, registrar quem realizou quais alterações, e restaurar versões anteriores do código. Usando o Git, cópias dos *scripts* são salvas em repositórios, o que impede que os códigos sejam sobrescritos (MARQUES, 2019).

De acordo com Rocha (2014), um sistema de controle de versões está ligado a três recursos centrais:

- Banco de dados do projeto (repositório): encarregado de guardar todos os artefatos de configuração relevantes.

- Recurso de gestão de versão: recurso utilizado para o armazenamento de todas as versões existentes de um artefato de configuração. Tal recurso possibilita a construção uma nova versão utilizando diferenças de versões passadas.
- Construção: possibilidade de usar com facilidade artefatos de configuração para construir uma versão de *software*.

Pressman (2011, p. 514) diz que “se você não controlar as alterações, elas controlarão você”. Segundo Pressman (2011) é fácil que alterações não controladas prejudiquem a qualidade de um *software* e que isso atrase sua entrega. O autor ainda afirma que é essencial uma boa gestão das alterações como parte essencial para a gestão da qualidade do *software*.

2.5 MongoDB

O MongoDB é um tipo de banco de dados diferente dos banco de dados tradicionais. Ao contrário dos bancos de dados tradicionais, como os que utilizam a linguagem SQL, que seguem o modelo de dados relacional e lidam com registros, o MongoDB lida diretamente com os documentos, formatados de forma semelhante a JSON, sendo portanto considerado um banco de dados orientado a documentos (MEDEIROS, 2014).

A ideia do MongoDB, segundo Boaglio (2015), é possibilitar que os documentos obtenham todos os dados necessários sem a limitação dos bancos de dados relacionais. Boaglio (2015) diz também que os documentos possuem dados variados, sendo que um documento pode possuir valores simples como uma String contendo uma palavra, uma representação do tipo data, um número, ou um conjunto de dados variados dentro do mesmo documento. Os documentos são agrupados por coleções (*collections*), sendo que um conjunto dessas coleções constitui um banco de dados.

O MongoDB armazena, recupera e gerencia dados semiestruturados. Tais dados são armazenados dentro de documentos, usando pares de chave/valor, ou seja, cada documento possui uma chave identificadora única e o valor referente aos dados que constituem o documento. Os documentos do MongoDB estão em formato BSON, que é uma versão binária do JSON. Para se realizar as buscas, além da busca por chave/valor, linguagens de consulta ou *Application Programming Interface* (APIs) podem ser usadas para recuperar os documentos do banco (SOLAGNA; LAZZARETTI, 2016).

O MongoDB inicialmente foi desenvolvido como um componente de serviço pela empresa 10gen em outubro de 2007, passando a ser um *software open source* em 2009.

2.6 Elasticsearch

O Elasticsearch é uma ferramenta para realização de análise de dados e buscas construída para trabalhar com grandes volumes de dados, permitindo indexar documentos e realizar buscas em tais documentos de forma quase instantânea. Santana (2015) define o Elasticsearch como uma ferramenta para mineração e tratamento de textos, cujo principal objetivo é atuar sobre grandes bases de dados, buscando e armazenando dados de forma rápida, simples e eficiente.

Sossoloti (2017) afirma que o Elasticsearch permite a análise e o armazenamento de grandes quantidades de dados quase em tempo real. Tal ferramenta possui um motor de busca de textos escalável, além de possuir uma infraestrutura baseada em *clusters*, o que a torna boa para retornar pesquisas pesadas.

Segundo Boscarino (2018), as buscas realizadas pelo Elasticsearch ocorrem por meio do princípio de Índice Invertido, sendo executado da seguinte forma:

- O Elasticsearch separa todos os termos de um documento em *Tokens* no momento de sua indexação.

- Em seguida, é feita uma medição desses *Tokens* por meio de sua relevância.
- Após a medição, o Elasticsearch organiza os *tokens* em um índice, e informa a cada token quais documentos estão contidos nele.
- Ao se realizar uma busca, ao invés de buscar de forma desordenada em todos os documentos, ela será realizada sobre o índice invertido.

Boscarino (2018) diz que o Elasticsearch é um motor de busca que ocorre em semi-tempo-real, e atribui sua eficiência ao processo de indexação realizado pela ferramenta.

2.7 BPM

BPM (*Business Process Management*, ou Gestão de Processos de Negócio) é o conjunto de práticas focadas na melhoria contínua dos processos de uma empresa. O BPM utiliza as melhores práticas de gestão na sua metodologia, como mapeamento dos processos, modelagem, maturidade, automação, entre outros. Isso tudo focado em aumentar a rentabilidade, criar vantagens competitivas e reduzir custos.

De acordo com Almeida (2015), o BPM objetiva unir a estratégia de uma empresa com as necessidades de seus clientes. Segundo o autor, o BPM reúne práticas e princípios voltados para gestão, que são empregados aos processos de uma organização e permitem que ela realize com mais eficiência a análise, aplicação, definição, gestão e monitoramento dos processos, tornando-a mais competitiva no mercado.

Os principais conceitos para que se possa compreender o BPM, de acordo com Almeida (2015) são:

- Mapeamento de Processos - ferramenta gerencial com o propósito de mapear os processos de uma organização, visando entendê-los e melhorá-los.

- Padronização de Processos – desenvolvimento de um padrão a ser seguido por meio da organização e formalização dos processos.
- Melhoria de Processos – incremento de melhorias, visando reparar defasagens nos processos de uma organização.
- Otimização de Processos – ações realizadas por uma organização, visando a melhoria do desempenho de seus processos.

Com a utilização do BPM processos podem ser mapeados, possibilitando a sua otimização. Processos melhorados tendem a possuir maior agilidade em suas tarefas e integração funcional, ajudando a organização a alcançar seus objetivos de negócio (PIZZA, 2012).

2.7.1 BPMN

BPMN (*Business Process Model and Notation*, ou Modelo e Notação de Processos de Negócio) é uma notação para modelagem de processos que define símbolos para serem utilizados no mapeamento do processo. Cada um desses símbolos se refere a uma regra específica, tendo portando um comportamento diferente dentro do processo (ALMEIDA, 2015).

Segundo Pizza (2012), a utilização BPMN permite uma melhor documentação de modelos e compreensão de processos. A utilização do BPMN possibilita o mapeamento dos processos de forma clara e simples, sendo possível uma fácil compreensão das etapas e dos fluxos presentes em um processo.

Alguns exemplos do que tais símbolos podem representar são:

- Determinar as atividades do processo e apresentar se a tarefa deverá ser executada por meio de uma interação humana ou se será executada via scripts (passos executados por meio de regras implementadas).
- Especificar convergência ou divergência de fluxos, permitindo analisar o comportamento do processo durante sua execução.

- Apresentar eventos iniciais e finais do processo, podendo-se ter uma visão total de todo o processo, do seu começo ao seu término.

2.7.2 BPMS

BPMS (*Business Process Management System*, ou Sistemas de Gestão de Processos de Negócio) são as tecnologias que possibilitam a modelagem e a automação de processos.

Tais sistemas permitem o desenvolvimento de processos com eventos, tarefas e fluxos automatizados, permitindo o mapeamento, a execução e o monitoramento de tais processos. O uso dessa ferramenta permite que uma organização tenha maior controle sobre seus processos e maior agilidade na execução dos mesmos.

Alguns exemplos de utilização de tais processos são:

- Início de atividades automáticas, possibilitando que processos iniciem e terminem sem a necessidade de uma ação humana. Um caso que isso pode ser utilizado é na cobrança de um produto de assinatura, onde a cobrança deve ser realizada mensalmente no dia do faturamento estabelecido do produto.
- Atribuição automática de atividades a áreas responsáveis. Durante a execução do processo, de acordo com a sequência de seu fluxo, atribui automaticamente à área necessária para a sua execução, o que reduz tempo e recursos gastos para tal ação.

A utilização de uma ferramenta como o BPMS apresenta benefícios como a melhora na eficiência da gestão, diminuição de custos, rápido gerenciamento do desempenho e redução do tempo para a tomada de decisões (ALMEIDA, 2015).

2.8 SCRUM

A metodologia SCRUM é uma metodologia de desenvolvimento ágil que tem como objetivo diminuir a burocracia e as documentações excessivas para focar em entregas pequenas, rápidas e incrementais que gerem valor para o cliente.

Uma equipe que utiliza a metodologia SCRUM é composta basicamente de três papéis:

- Product Owner (ou dono do produto) é quem está na liderança sobre o produto, o único responsável por decidir quais recursos e funcionalidades serão construídos e qual a ordem que devem ser feitos.
- ScrumMaster é responsável por ajudar a todos os envolvidos a entender e abraçar os valores, princípios e práticas do Scrum.
- Time Scrum é o time desenvolvimento do produto, sendo a junção de todas as pessoas em uma equipe multidisciplinar, e que são responsáveis pela concepção, construção e testes do produto. Um time Scrum geralmente possui uma composição de 5 a 9 membros.

Os artefatos do processo Scrum são:

- Product Backlog, segundo Batista (2012), é uma listagem definida pelo PO (Product Owner) que contém o que precisa ser feito e a sua prioridade.
- Sprint Backlog, também segundo Batista (2012), é uma listagem definida na Planning que contém o que será realizado durante a Sprint. A Planning é uma reunião que ocorre no início da Sprint em que é definido o que será realizado no decorrer da Sprint.

Os eventos do processo Scrum são:

- Sprint é o nome de um ciclo de desenvolvimento do Scrum. Os Sprints podem ter a duração de 2 a 4 semanas, sendo esse o time box do Ciclo.

- Sprint Planning é a reunião que ocorre para realizar o planejamento da Sprint. Nessa reunião são apresentadas as atividades a serem desenvolvidas no decorrer da Sprint e suas prioridades.
- Daily é uma reunião diária entre os membros da equipe. Essa reunião possui como característica a ter a periodicidade diária, ocorrer sempre em um horário marcado, e durabilidade de 5 a 15 minutos. Nessa reunião cada membro do time deve responder às seguintes perguntas: “O que você tem feito desde ontem?”, “O que você está planejando fazer hoje?”, e “Você tem algum problema impedindo você de realizar seu objetivo?”.
- Sprint Review segundo Batista (2012), é uma reunião que ocorre no fim da Sprint, onde os membros da equipe fazem uma revisão de tudo o que foi feito e que ficou pendente. Além disso, pode-se apresentar o que foi incrementado ao sistema para o cliente, possibilitando que ele acompanhe o desenvolvimento do produto e faça as sugestões que achar pertinente.
- Sprint Retrospective segundo Batista (2012), é uma reunião que ocorre no fim da Sprint, cujo objetivo é analisar os pontos positivos e negativos da Sprint. O objetivo dessa reunião é a melhora do processo de desenvolvimento, podendo analisar o que se pode manter e o que deve melhorar.

Sabbagh (2014) aponta como alguns dos benefícios do Scrum as entregas constantes de incrementos aos clientes, a diminuição das ameaças ao projeto, o aumento da produtividade e da qualidade do produto, a fácil visualização de progresso durante o decorrer do projeto e a utilização de mudanças como vantagem.

3 METODOLOGIA

Nesta seção serão detalhados os procedimentos utilizados durante o processo de desenvolvimento do projeto.

3.1 Procedimentos metodológicos

Este trabalho foi realizado no período de Janeiro a Dezembro de 2019, tendo como finalidade o desenvolvimento de um software para uma empresa de análise de crédito de mercado, que é cliente da SYDLE. O objetivo desse trabalho foi o desenvolvimento de tal sistema, visando entregas constantes ao cliente, atendendo suas demandas e dando suporte às suas necessidades.

Durante o desenvolvimento deste trabalho, foi possível ter uma visão prática de conceitos vistos em sala de aula, além de adquirir habilidades em novas ferramentas, possibilitando o ganho de experiência em arquitetura de software, aprimorando os conhecimentos sobre desenvolvimento de sistemas. A realização deste trabalho também trouxe ao aluno vivência no ambiente empresarial, tendo contato direto com o mercado de trabalho.

3.2 Processo de desenvolvimento

O desenvolvimento do software seguiu a metodologia SCRUM. Foram realizadas Sprints com duração média de uma semana, sendo que, no início de cada Sprint, realizou-se a Planning (reunião de planejamento da Sprint), onde foram passadas as atividades a serem realizadas.

Durante a realização da Sprint, foi realizado o desenvolvimento das atividades passadas, sendo que no final de cada atividade, foram realizados testes buscando constatar que as funcionalidades esperadas estivessem sendo contempladas e a ausência de erros. Caso não houvessem pendências para a atividade, passava-se para a próxima, seguindo o andamento da Sprint.

Todos os dias, exceto o da Planning, foram realizadas Dailys, para que o time pudesse saber como estava sendo o andamento da Sprint, quais atividades estavam concluídas, quais ainda estavam em andamento e quais as pendências ou dificuldades encontradas. No final da Sprint foi realizada a Review, onde os membros do time falaram como foi o andamento da Sprint, os pontos positivos e negativos e as pendências que ficaram para a próxima Sprint.

O SYDLE ONE é dividido em três ambientes, são eles: Desenvolvimento, Homologação e Produção. O ambiente de Desenvolvimento é onde as atividades são desenvolvidas e no término de tais atividades, onde ocorrem os testes específicos das atividades.

Após o término das atividades e a realização de testes bem sucedidos, realiza-se uma cópia das alterações feitas para o ambiente de Homologação. Este ambiente é onde ocorrem os testes do produto como um todo.

Quando o líder do projeto achar pertinente, é realizada a atualização do ambiente em Produção. Esse é o ambiente que o cliente usa diretamente, rodando seus processos, gerindo seus conteúdos, extraindo análises de dados, etc.

4 ATIVIDADES DESENVOLVIDAS

Nesta seção serão apresentadas as principais atividades realizadas durante o estágio.

Durante o desenvolvimento do sistema, o estagiário atuou como um dos integrantes do time de desenvolvimento, trabalhando com *back-end* e com o banco de dados.

O *back-end* é a parte do sistema que é responsável pelas APIs, processamento de dados, regras de negócios e *webservices*. Nele foram realizados o desenvolvimento de melhorias do sistema e a correção de *bugs* encontrados.

O banco de dados armazena os dados do sistema, e nele foram realizadas buscas e atualizações de classes e objetos. Durante o desenvolvimento do sistema, foram realizadas buscas e rodados *scripts* para criar, atualizar e remover dados no sistema. Além de atualizações no próprio ambiente, *scripts* foram utilizados para buscar classes e objetos em determinado ambiente e os dados obtidos eram passados em *scripts* de criação ou atualização, com isso tais dados eram replicados nos demais ambientes.

A seguir serão apresentadas algumas das melhorias desenvolvidas pelo estagiário no decorrer do estágio.

4.1 Alteração de plano de assinatura

A empresa cliente possui assinatura de planos, e desejava que seus clientes pudessem migrar de plano. Alguns dos serviços por assinatura da empresa possuíam categorias e periodicidade diferentes. Para atender a essa necessidade do cliente, foi desenvolvido um processo de alteração de plano por meio de cálculo de pró-rata.

Como o cobrado dos clientes pela assinatura é pré-pago (cliente paga pelo serviço antes de utilizar), o cálculo de pró-rata deveria considerar um valor proporcional a um período já pago e ainda não utilizado, considerando entre a data

da migração ao fim do período pago, e um valor proporcional ao mesmo período considerando o valor proporcional do novo produto a ser consumido. Com isso o cálculo deveria ser uma subtração entre o valor proporcional pago (referente ao serviço atual) e o valor proporcional a se pagar (referente ao serviço a ser pago). Tal cálculo gera um valor que indica se o cliente deve ser cobrado ou se deve possuir um saldo referente ao balanço levantado.

Caso o balanço dê negativo e o cliente deva ser cobrado pela a alteração do produto, o serviço mantém-se o mesmo e deve ser gerado um pedido de migração. Esse pedido é faturado e quando for registrado seu pagamento será realizada a alteração automática do serviço.

Caso o balanço dê positivo e o cliente possua um saldo referente a alteração de plano, o serviço será alterado no ato da solicitação e o cliente começará a usufruir do produto novo. O saldo gerado pelo balanço será descontado nos próximos faturamentos do produto, até que esse saldo seja zerado.

Para o desenvolvimento dessa funcionalidade ao sistema, foram alteradas classes como contrato, pedido e fatura. Nessas classes, foram inseridos campos e métodos para que pudessem realizar as funções solicitadas. Nos métodos das classes, para o desenvolvimento, foi utilizada a linguagem JavaScript, e foram utilizado o elasticSearch para realizar buscas dos objetos no banco de dados.

4.2 Atendimento via CNPJ

A empresa para a qual o sistema foi desenvolvido possui clientes, tanto pessoas físicas quanto empresas, porém o processo de atendimento suportava apenas pessoas físicas. Foi desenvolvida então uma melhoria no atendimento para suportar também pessoas jurídicas.

Ao iniciar o atendimento para empresas, entra-se com o número do CNPJ da empresa e o número desse documento é pesquisado na base, averiguando se já possui cadastro dessa empresa no sistema. Se a empresa não for cadastrada,

abre-se uma página de cadastro onde são preenchidas as informações referentes à organização, sendo tais dados salvos na base. Caso a empresa seja cadastrada, os dados serão buscados para o início do atendimento.

O processo vincula também a empresa que está sendo atendida a um responsável pelo atendimento. Isso permite um registro de quem foi o responsável (pela empresa) que foi atendido.

O desenvolvimento dessa história foi no processo de Atendimento, e para a realização da mesma, foi utilizada a linguagem JavaScript para o desenvolvimento de scripts nos métodos do processo. O elasticSearch foi também utilizado para que se pudessem ser realizadas buscas no banco de dados.

4.3 Envio de e-mails automáticos

No processo de atendimento, foram criados e-mails tanto de envio interno quanto de envio aos clientes da empresa.

Exemplos de e-mails enviados para os clientes:

- E-mail de pesquisa de satisfação;
- Envio de protocolo de atendimento;

Exemplos de e-mails internos:

- Alertar sobre o vencimento do SLA de um atendimento;
- Alertar a atribuição de atendimento a uma determinada área de atendentes;

Os e-mails criados são disparados durante o atendimento ou programados para envio automático. Tais e-mails possuem a finalidade de melhorar o atendimento e o relacionamento da empresa com seus clientes.

Para o envio dos e-mails, foi utilizada a linguagem JavaScript para inserir mensagens na fila de execução, programando o disparo automático dos e-mails nos momentos estabelecidos. O envio de tais mensagens foi programado em script-Tasks presentes no processo de Atendimento do sistema.

4.4 Processo de validação por OCR

Foi desenvolvido um processo de validação por OCR (*Optical Character Recognition*, ou Reconhecimento ótico de caracteres). O processo realiza o cadastro de pessoas validando dados de entrada com dados inseridos com a imagem de um documento.

No processo, o cliente digita os dados de uma pessoa, e em seguida entra com um arquivo de imagem. O processo faz uma requisição a uma API externa e retorna os dados encontrados no documento.

Como o retorno recebido obtém informações como um texto não tratado, após a análise dos documentos são utilizadas expressões regulares para extrair dados relevantes. Com os dados tratados, é realizada a comparação de dados do documento com as informações inseridas no processo, e caso as informações coincidam, o cadastro da pessoa na base é realizado.

Tal processo é importante para a empresa cliente para evitar fraudes e manter dados concretos em sua base.

O processo foi modelado utilizando a notação BPMN para a criação de seu fluxo. Dentro da userTask criada, foi realizado o desenvolvimento de um script, implementado na linguagem JavaScript, para a obtenção dos dados do documento, a comparação com os dados inseridos e a criação do contato na base.

4.5 Processo de suspensão de pedidos

Foi criado um processo para que os pedidos gerados para a alteração de plano, caso não sejam pagos, sejam cancelados. O processo busca os pedidos de alteração de plano que possuem faturas vencidas e realiza o cancelamento do pedido, de sua fatura e das remessas de sua fatura.

Tal processo trata-se de uma rotina, que foi configurada por meio de uma expressão cron, agendando o processo para rodar diariamente. Tal processo per-

mite que pedidos ativos não fiquem acumulados no sistema. Tal processo foi modelado usando-se a notação BPMN, e para o desenvolvimento de seus scripts foi utilizada a linguagem JavaScript e foi utilizada a ferramenta elasticSearch para a realização de buscas no banco de dados.

6 CONSIDERAÇÕES FINAIS

Nesta seção serão apresentadas as considerações finais relacionadas ao estágio e ao sistema desenvolvido.

6.1 Disciplinas importantes para o desenvolvimento do projeto

Como o curso de Sistemas de Informação da UFLA possui grande abrangência de conteúdo, algumas disciplinas vistas no decorrer do curso se destacaram para o aluno no desenvolvimento desse projeto. São elas:

- **Introdução a Lógica e Matemática Discreta:** Essas disciplinas foram de grande importância para o desenvolvimento do sistema por impulsionarem o uso de lógica para atingir determinado objetivo, ou seja, com elas podem-se buscar caminhos e soluções lógicas para solucionar problemas enfrentados durante o desenvolvimento do projeto.
- **Algoritmos e Estrutura de Dados 1, 2 e 3:** São disciplinas que, além de apresentarem o conceito de Introdução aos Algoritmos e de Estrutura de Dados, ensinaram o básico para se realizar a programação para o desenvolvimento de sistemas e a aplicação da lógica para fins de desenvolvimento de softwares.
- **Programação Orientada a Objetos:** Essa disciplina foi muito importante para a realização do estágio, por apresentar o conceito de Orientação a Objetos e ensinar ao aluno como realizar o desenvolvimento de sistemas utilizando tal paradigma. A grande importância dessa disciplina no decorrer do estágio deve-se ao fato de que o projeto foi desenvolvido utilizando esse paradigma.
- **Engenharia de Software:** Nesta disciplina o aluno pode aprender sobre o desenvolvimento de um software como um todo, desde a análise e levantamento de requisitos ao desenvolvimento e entregas ao cliente do sistema.

Durante a realização dessa matéria foi desenvolvido um e-commerce, assim como o sistema desenvolvido no estágio, possibilitando o ganho de conhecimento em desenvolvimento, visão do cliente e suas necessidades e experiência em trabalhar utilizando metodologia ágil.

- Bando de Dados e Sistemas Gerenciadores de Banco de Dados: Apesar de o sistema desenvolvido utilizar um banco de dados que não foi muito enfatizado no decorrer do curso, tais disciplinas foram importantes por apresentar o conceito e a lógica por trás dos bancos de dados, permitindo que o estagiário a tivesse maior facilidade em aprender a utilizar o MongoDB.

6.2 Conclusão

Por se tratar de um desenvolvimento utilizando uma metodologia ágil, o sistema ainda irá adquirir melhorias para suprir com maior qualidade às necessidades do cliente.

Com o desenvolvimento desse projeto, foi possível realizar o avanço tanto no sistema de *Billing* quanto no sistema de CRM da empresa cliente. Junto a isso permitiu o estagiário adquirir experiência na área de desenvolvimento de Software e o conhecimento e prática na utilização de diversas ferramentas.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Vinicius Nóbile de. O que é BPM (Business Process Management): o guia completo para implantar a Gestão de Processos na sua empresa!, 2015. Acesso: 01 de junho de 2019.

BATISTA, Pablo Pinheiro. Entendendo o SCRUM, 2012. Acesso: Acesso: 01 de junho de 2019.

BOAGLIO, Fernando. MongoDB: construa novas aplicações com novas tecnologias. Editora Casa do Código, 2015.

BOSCARINO, Aylan de Sousa. O que e Elasticsearch, 2018. Acesso: 01 de junho de 2019.

BRANAS, Rodrigo. Chapter 1. Getting Started with AngularJS. AngularJS Essentials. Packt Publishing Ltd, 2014. Acesso: 30 de maio de 2019.

CALDEIRA, Carlos. Introdução ao HTML. 2015. Acesso: 01 de junho de 2019.

DOS SANTOS, Lucas Gontijo; RUFINO, Ricardo Ribeiro. O Impacto no Desenvolvimento de Produtos Computacionais Utilizando Angular Js, Spring Framework e Java, 2016. Acesso: 30 de maio de 2019.

FERNANDES, G. R; JUNIOR, J. G. S; TRINDADE, J. G. M; MORAVIA, R. V. CRM- Uma ferramenta fundamental para a captação e manutenção de clientes. Revista Pensar Tecnologia, v.4, n.2, jul.2015. Acesso 03 de maio de 2019.

FLANAGAN, David. JavaScript: O guia definitivo. Bookman Editora, 2004.

MARQUES, Brendon. O Que é GitHub e Para Que é Usado, 2019. Acesso: Acesso: 18 de junho de 2019.

MEDEIROS, Higor. Introdução ao MongoDB, 2014. Acesso: Acesso: 18 de junho de 2019.

MICROSOFT PATTERNS; PRACTICES. Microsoft application architecture guide, 2nd edition. Microsoft Press, 2009. Acesso 03 de maio de 2019.

PEREIRA, Michael Henrique R. AngularJS: Uma abordagem prática e objetiva. Novatec Editora, 2014.

PIZZA, William Roque. A metodologia Business Process Management (BPM) e sua importância para as organizações. São Paulo: Trabalho de conclusão de curso, Faculdade de Tecnologia de São Paulo, 2012. Acesso: 01 de junho de 2019.

PRESCOTT, Preston. Programação em JavaScript. Babelcube Inc., 2016. Acesso: 01 de junho de 2019.

PRESSMAN, ROGER. Engenharia de Software: Uma abordagem Profissional. Sétima Edição. AMGH. Porto Alegre, 2011. 780 p.

ROCHA, Lucas Flach da. Aplicação da integração contínua no desenvolvimento de software. Sistemas de Informação-Pedra Branca, 2014.

SABBAGH, Rafael. Scrum: Gestão ágil para projetos de sucesso. Editora Casa do Código, 2014.

SANTANA, Luiz Henrique Zambom. Explorando Elasticsearch com Java – Introdução ao Desenvolvimento para Big Data. Rio de Janeiro: Editora Ciência Moderna Ltda., 2015.

SILVA, Maurício Samy. Criando sites com HTML: sites de alta qualidade com HTML e CSS. Novatec Editora, 2008.

SILVA, Maurício Samy. JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript. Novatec Editora, 2010.

SOLAGNA, Emanuele Araujo; LAZZARETTI, Alexandre Tagliari. UM ESTUDO COMPARATIVO ENTRE O MONGODB E O POSTGRESQL, 2016. Acesso: 01 de junho de 2019.

SOSSOLOTI, João Paulo. Buscas eficientes com Elasticsearch, 2017. Acesso: 01 de junho de 2019.