



**RAPHAEL FERNANDES REIS RORIZ**

**DESENVOLVIMENTO LOW CODE PARA APOIAR  
TRANSFORMAÇÃO DIGITAL**

**LAVRAS – MG**

**2023**

**RAPHAEL FERNANDES REIS RORIZ**

**DESENVOLVIMENTO LOW CODE PARA APOIAR TRANSFORMAÇÃO DIGITAL**

Relatório de Estágio Supervisionado  
apresentado à Universidade Federal de  
Lavras como parte das exigências do curso de  
Ciência da Computação para obtenção do título  
de Bacharel.

Profa. Dra. Renata Teles Moreira

Orientadora

**LAVRAS – MG**

**2023**

**Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos  
da Biblioteca Universitária da UFLA**

Roriz, Raphael Fernandes Reis

Desenvolvimento Low code para apoiar transformação digital / Raphael Fernandes Reis Roriz. – Lavras : UFLA, 2023.

27 p. : il.

Trabalho de Conclusão de Curso (TCC)–Universidade Federal de Lavras, 2023.

Orientadora: Profa. Dra. Renata Teles Moreira.

Bibliografia.

1. Low Code. 2. Monografia. 3. Dissertação. 4. Tese. 5. Trabalho Científico – Normas. I. Universidade Federal de Lavras. II. Título.

CDD-808.066

**RAPHAEL FERNANDES REIS RORIZ**

**DESENVOLVIMENTO LOW CODE PARA APOIAR TRANSFORMAÇÃO DIGITAL**

Relatório de Estágio Supervisionado apresentado à Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel.

APROVADO em 23 de Fevereiro de 2023.

Prof. Dr. Maurício Ronny de Almeida Souza DCC/UFLA

Prof. Dr. Paulo Afonso Parreira Junior DCC/UFLA

Profa. Dra. Renata Teles Moreira  
Orientadora

**LAVRAS – MG  
2023**

*Dedico este trabalho aos meus pais*

## **AGRADECIMENTOS**

Agradeço aos meus pais e a minha irmã, pelo apoio e incentivo.

Agradeço a professora Dra. Renata Teles Moreira pela disposição em ser minha orientadora e por toda a ajuda.

Agradeço aos meus amigos e colegas de curso e de estágio.

Agradeço a Universidade Federal de Lavras e ao Departamento de Ciência da Computação, por todas as oportunidades e ensinamentos.

Agradeço a empresa Levty, pela oportunidade do estágio.

## RESUMO

Este documento trata-se de um relatório acerca das atividades desenvolvidas durante o estágio na empresa de desenvolvimento de software Levty utilizando a plataforma de desenvolvimento Low Code Sydle One, principalmente em atividades relacionadas a automatização de processos de negócio com o objetivo de realizar-se a transformação digital nas empresas clientes, para isso foram apresentados conceitos referentes à organização, conceitos, metodologias e ferramentas de desenvolvimento de software que foram utilizados durante o período do estágio. O objetivo deste trabalho é apresentar como os conceitos teóricos, e as atividades práticas, abordados durante o curso de bacharelado em Ciência da Computação se relacionam com o desenvolvimento profissional de software, e em mostrar os ganhos além do conhecimento técnico durante o período do estágio, como noções de trabalho em equipe e o alto contato com conhecimentos oriundos de outras áreas que não sejam relacionadas a computação.

**Palavras-chave:** Desenvolvimento de Software. Ciência da Computação. Low Code. Transformação Digital

## ABSTRACT

This document is a report on the activities carried out during my internship at Levty, a software development company. During the internship, I primarily used the Sydle One Low Code development platform to automate business processes and facilitate digital transformation in client companies. This report will provide an overview of the organization, concepts, methodologies, and software development tools that I used during the internship period. The objective of this report is to demonstrate how the theoretical concepts and practical activities covered during my BSc course in Computer Science are relevant to my professional development in software development. In addition to technical knowledge, the report will also highlight other gains from the internship period, such as teamwork skills and exposure to knowledge from other areas outside of computing.

**Keywords:** Software Development. Computer Science. Low Code. Digital Transformation

## LISTA DE FIGURAS

Figura 4.1 – Exemplo de Estória de Usuário . . . . .	18
Figura 4.2 – Exemplo de fluxo de processo BPM no Sydle One . . . . .	20
Figura 4.3 – Exemplo de Script para criar registro de novo cliente . . . . .	22

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>8</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>10</b>
<b>2.1</b>	<b>Transformação Digital</b>	<b>10</b>
<b>2.2</b>	<b>Team as a Service</b>	<b>10</b>
<b>2.3</b>	<b>Desenvolvimento de Software Low-Code</b>	<b>11</b>
<b>2.3.1</b>	<b>Sydle One</b>	<b>11</b>
<b>2.4</b>	<b>Desenvolvimento Ágil</b>	<b>12</b>
<b>2.5</b>	<b>API - Application Programming Interface</b>	<b>13</b>
<b>3</b>	<b>Sobre a organização</b>	<b>15</b>
<b>3.1</b>	<b>Times</b>	<b>15</b>
<b>3.2</b>	<b>Processo de Desenvolvimento</b>	<b>16</b>
<b>4</b>	<b>ATIVIDADES REALIZADAS</b>	<b>17</b>
<b>4.1</b>	<b>Treinamento</b>	<b>17</b>
<b>4.2</b>	<b>Desenvolvimento de software em projetos reais</b>	<b>17</b>
<b>4.2.1</b>	<b>Levantamento de requisitos de software</b>	<b>18</b>
<b>4.2.2</b>	<b>Desenvolvimento de software - BPM</b>	<b>19</b>
<b>4.2.2.1</b>	<b>Desenho do Fluxo</b>	<b>19</b>
<b>4.2.2.2</b>	<b>Configuração do Fluxo</b>	<b>20</b>
<b>4.2.2.3</b>	<b>Escrita dos Scripts</b>	<b>21</b>
<b>4.2.3</b>	<b>Integrações com APIs externas</b>	<b>23</b>
<b>4.2.4</b>	<b>Teste de software</b>	<b>24</b>
<b>4.2.5</b>	<b>Homologação e Deploy</b>	<b>25</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>26</b>
	<b>REFERÊNCIAS</b>	<b>27</b>

## 1 INTRODUÇÃO

Nos dias de hoje, a transformação digital é um dos fatores primordiais para que as empresas possam se manter competitivas no mercado e se adaptem às novas tendências e demandas dos consumidores. Neste contexto a digitalização dos processos de negócio é uma maneira eficiente de alcançar esse objetivo, automatizando tarefas, eliminando atividades manuais e reduzindo custos, além da geração e armazenamento de dados que podem vir a ser transformados em conhecimento para as empresas. Nesse sentido, as plataformas Low Code têm se mostrado uma solução eficiente para ajudar as empresas a se adaptarem a tal realidade, facilitando o processo de transformação digital de tais organizações. Com a possibilidade de desenvolver aplicações de forma mais rápida e simplificada, com recursos que facilitam o trabalho do desenvolvedor, as plataformas Low Code podem ajudar as empresas a acelerarem seus projetos de transformação digital e a reduzirem os custos e o tempo de desenvolvimento. Porém ainda faz-se necessários profissionais que tenham habilidades que os permitam compreender as necessidades do negócio e em utilizarem as ferramentas e tecnologias disponíveis para atender tais necessidades, é neste contexto que atua a empresa Levty, que é especialista na aplicação da plataforma digital Sydle One em projetos de transformação digital para empresas dos mais diversos ramos.

Com isto em mente, o objetivo deste trabalho é relatar as principais atividades desenvolvidas pelo estagiário durante o estágio realizado na empresa Levty no período de 13 de junho de 2022 a 31 de dezembro de 2022. As atividades desenvolvidas pelo estagiário englobaram todo o ciclo de vida do desenvolvimento de software, desde o planejamento do que seria desenvolvido juntamente com os clientes, a elicitación de requisitos do software, o desenvolvimento dos códigos, testes e liberação para uso da solução solicitada. As soluções foram desenvolvidas para diferentes clientes da empresa e consistiam, em sua maioria, na modernização e automatização de processos de negócio, tudo isso focado nos projetos de transformação digital das empresas clientes.

O período de estágio foi composto por um treinamento na plataforma *Sydle One*, onde são desenvolvidos a maioria dos projetos da empresa. Em seguida, o estagiário participou de projetos com clientes, onde integrou equipes de desenvolvimento de software. As equipes na empresa Levty geralmente são compostas por um líder técnico, que é um desenvolvedor com maior conhecimento técnico, dois ou mais desenvolvedores, função ocupada pelo estagiário e que é responsável pelo desenvolvimento em si das soluções propostas pelo cliente, um *agile*

*master*, que é a pessoa responsável por facilitar a prática de metodologias ágeis pelo time e um *product owner*, um funcionário da empresa cliente que tem alto grau de conhecimento dos processos que estarão sendo implementados nas soluções desenvolvidas.

Levando-se em consideração todo o conteúdo abordado durante o curso de bacharelado em Ciência da Computação, bem como a alta aplicabilidade prática da computação no dia a dia de todos, somando-se a crescente demanda por soluções computacionais que auxiliem empresas a agilizarem seus processos, percebe-se um impacto positivo que o estágio oferece para o aluno de Ciência da Computação. O estágio provê ao aluno um contato com desenvolvimento real de software em um contexto específico da área de atuação do cliente, sendo um importante passo para a ampliação do conhecimento técnico dos alunos que desejam seguir carreira com desenvolvimento de software.

Além deste capítulo introdutório, este trabalho está organizado da seguinte forma: no Capítulo 2 são apresentados os principais conceitos e tecnologias utilizados no desenvolvimento das atividades do estágio; no Capítulo 3 são apresentadas as atividades desenvolvidas pelo estagiário durante o estágio; e, por fim, no Capítulo 4 são feitas as considerações finais a respeito do estágio.

## 2 REFERENCIAL TEÓRICO

Este capítulo tem o objetivo de descrever os principais conceitos e tecnologias que foram vistos ou utilizados no desenvolvimento das atividades durante o estágio.

### 2.1 Transformação Digital

A transformação digital é um processo pelo qual as empresas estão passando para se adaptar às mudanças tecnológicas e integrar as tecnologias digitais em seus processos de negócios, também podemos dizer que a transformação digital refere-se à adoção de tecnologias digitais para melhorar a eficiência, a produtividade e a experiência do cliente, além de gerar inovação e novas oportunidades de negócios, adaptando-se as novas dinâmicas de competição no mercado e as novas dinâmicas de interação e entrega de valor para seus clientes (ROGERS, 2017).

A implementação bem-sucedida da transformação digital pode ajudar as empresas a garantirem melhores resultados, melhorando seu desempenho e aumentando seu alcance de mercado. No entanto, a transformação digital também apresenta desafios e riscos, como a exclusão de setores menos favorecidos da sociedade, que podem não ter acesso as tecnologias utilizadas (DAEHN; COSTA; PEREIRA, 2021), dependência da tecnologia de outros países e riscos associados a segurança da informação (GAIWORONSKAYA et al., 2020).

É importante entendermos esse conceito pois projetos de transformação digital são a especialidade da Levy.

### 2.2 Team as a Service

O modelo "time como serviço" é uma nova forma como as empresas oferecem seus serviços aos clientes, sendo serviços de desenvolvimento de software no contexto da Levy. Ao invés de vender um projeto fechado de desenvolvimento de software para seus clientes, entregando ao final do contrato um produto desenvolvido por exemplo, a Levy oferece o acesso a equipes de desenvolvimento de software, que podem ser customizadas de acordo com as necessidades do cliente, para atuar em suas demandas de desenvolvimento.

Este modelo vem sendo cada vez mais utilizado devido a diversas vantagens de sua utilização, como por exemplo o foco nos requisitos pela parte do cliente, onde ele não mais

precisará lidar com contratação, treinamento e gestão da equipe, ficando essas atribuições de responsabilidade da empresa fornecedora do serviço (MACHADO, 2021), além da flexibilidade em se ampliar ou diminuir o tamanho da equipe conforme necessário e também a previsibilidade de custos, onde o cliente paga um valor fechado pelos serviços prestados.

## 2.3 Desenvolvimento de Software Low-Code

Com o aumento da demanda por softwares de computador, pela necessidade rápida de entrega destes e para manter o acelerado ritmo de competitividade do mercado de soluções em software, surgiram plataformas de desenvolvimento *low-code*, ou de pouco código, que são ambientes de desenvolvimento que dão a possibilidade de se desenvolver soluções em software utilizando pouco código de programação (ALVES; ALCALÁ, 2022).

As plataformas de desenvolvimento *low-code* possibilitam um alto rendimento aos desenvolvedores, que conseguem entregar soluções completas de software em menos tempo e podem focar seus esforços na modelagem, funcionalidades, usabilidade, testes e outros aspectos não técnicos da solução. Além disso, outra vantagem das plataformas *low-code* é a facilidade e baixo nível de esforço para a criação de ambientes e implementação de novos recursos, bem como um menor tempo de treinamento dos desenvolvedores (WASZKOWSKI, 2019). No contexto do estágio, o conceito de *low-code* foi importante para o estagiário entender sobre o funcionamento da plataforma *low-code* Sydle One, utilizada durante o desenvolvimento de suas atividade.

### 2.3.1 Sydle One

O *Sydle One* é uma plataforma de soluções digitais corporativas desenvolvida e mantida pela empresa Sydle<sup>1</sup>. É uma plataforma *low-code* que tem o objetivo de entregar soluções de maneira fácil e rápida para seus clientes (Sydle, 2022). Durante o período de estágio, foi utilizada pelo estagiário principalmente para a automatização de processos BPM, auxiliando os clientes da Levty em seus processos de transformação digital, automatizando dentro da plataforma, processos de negócio que antes eram realizadas de maneira manual dentro da organização. Juntamente com este recurso a plataforma também oferece:

---

<sup>1</sup> <https://www.sydle.com/>

- BPM, em inglês *Business Process Management*, que é uma ferramenta utilizada principalmente para automatizar o gerenciamento de processos de negócio, sendo um sistema para a criação de modelos de processos de negócios de maneira gráfica, utilizando a notação BPMN (*Business Process Modeling Notation*) (WHITE, 2004).
- CRM, em inglês *Customer Relationship Management*, que é um sistema responsável por gerenciar as interações com os clientes da empresa, de modo a permitir um maior conhecimento dos cliente, agrupá-los em diferentes perfis e apoiar a tomada de decisões estratégicas baseada nas informações extraídas do sistema.
- ECM, em inglês *Enterprise Content Management*, que é um sistema capaz de coletar e armazenar de maneira sistematizada as informações geradas por uma empresa, além disso, o sistema também é responsável por fazer a gestão, permitir acesso e integração com essa base de informações coletadas.
- *Service Desk*, que é um sistema de centralização de solicitação de suporte técnico, fazendo a comunicação entre empresa, empregados, parceiros e clientes.
- *Billing*, que é um sistema que faz a geração e cobrança de faturas de maneira automatizada e inteligente, além de fazer o controle destas faturas.

## 2.4 Desenvolvimento Ágil

Durante muitos anos, no início do desenvolvimento de software, os projetos eram geridos por modelos advindos da indústria, onde, na maioria dos casos, existe uma esteira de produção, onde a matéria prima passa por uma série de etapas sequenciais, de maneira a se tornar um produto no final da esteira. No contexto do desenvolvimento de software, esse modelo era chamado de “Sequencial”, onde cada etapa do ciclo de vida de um software era executada somente quando a etapa anterior acabasse, e ao final era esperado um software em perfeito funcionamento e de acordo com os requisitos e documentações que foram elaboradas no início e durante o projeto (PRESSMAN, 2021). Dessa forma, caso algum destes requisitos tenha sido levantado de maneira equivocada, os usuários finais só perceberiam quando estivessem de fato usando o software desenvolvido.

Visando resolver este problema, e vários outros como alto custo dos projetos de software, elevado tempo de desenvolvimento e dificuldade em manter e evoluir o software, começaram

a surgir metodologias de desenvolvimento de software que, segundo seus criadores, seriam mais “ágeis” do que os modelos tradicionais, neste contexto temos a escrita do Manifesto Ágil (BECK et al., 2001) que tem como objetivo manter a equipe focada no que realmente entrega valor para o projeto. Essas novas metodologias tratavam o processo de desenvolvimento como um ciclo, sendo possível esse ciclo se repetir diversas vezes durante o desenvolvimento de um projeto, sendo este o modelo de desenvolvimento seguido pelas equipes dentro da Levty.

De forma geral, segundo Valente (2020) este ciclo de desenvolvimento funciona da seguinte maneira: primeiramente ocorre a fase de “*design*”, ou seja, definir o que será desenvolvido, bem como desenhar a solução. Em seguida vem a fase de “*build*”, onde, de fato é desenvolvida a solução. A fase seguinte é a de “*test*”, onde é testado o que foi desenvolvido e, em seguida, ocorre a fase de “*deploy*” e de “*review*”, onde a solução é disponibilizada para o usuário final e avaliada por estes, possibilitando que o ciclo volte para alguma das etapas anteriores.

Vale ressaltar que, um dos principais pontos das metodologias ágeis é dividir o projeto em entregáveis parciais, ou seja, não espera-se que o produto final desses ciclos seja a solução pronta, mas sim, alguma funcionalidade parcial da solução, que seja funcional e passível de uso (VALENTE, 2020).

## 2.5 API - Application Programming Interface

Com os crescentes avanços no desenvolvimento de software e das redes de comunicação, surgiu-se a necessidade de que diferentes componentes de software se comunicassem. No entanto, uma das dificuldades encontradas para foi que as aplicações computacionais eram desenvolvidas em diferentes linguagens de programação, não sendo possível essa comunicação de maneira nativa. Para isso, foram criados modelos de troca de informações que independessem da linguagem utilizada chamadas de API (*Application Programming Interface*).

As APIs são uma camada de acesso aos dados e funcionalidades da aplicação de maneira genérica que utiliza a comunicação em rede por meio de algum protocolo de comunicação. As APIs funcionam como um contrato entre aplicações, definindo como deve ser as informações contidas na solicitação, bem como o modelo do que será enviado como resposta para o solicitante (AWS, 2022).

Durante o desenvolvimento das atividades do estágio foi utilizado o conceito de API REST, utilizando-se o protocolo de comunicação HTTP e o formato de troca de informações JSON, o entendimento correto do conceito foi importante pois ocorreram situações onde fez-se necessário que o estagiário realizasse integrações entre o Sydle One e outros sistemas, tanto para obter quanto para enviar dados dos processos de negócio automatizados dentro da plataforma.

APIs REST são APIs desenvolvidas de acordo com as restrições e definições da arquitetura REST (*Representational State Transfer*), proposta por (FIELDING, 2000) e definida principalmente pelos critérios :

- A arquitetura é do tipo cliente/servidor e as requisições seguem o protocolo HTTP;
- O servidor não armazena informações do cliente entre as requisições, por isso é chamada de comunicação *stateless*;
- Existe o uso de cache para que as interações entre cliente e servidor sejam mais rápidas;
- A interface entre os componentes deve ser uniformizada, para isso quatro critérios devem ser satisfeitos:
  - Os recursos do servidor devem ser identificáveis e serem separados das representações enviadas ao cliente;
  - Os recursos do servidor são acessados pelo cliente através de representações de dados, como o JSON, com informações suficientes para que este seja manipulado no servidor;
  - As respostas do servidor devem ser auto-descritivas, contendo informações necessárias para que o cliente saiba agir de acordo com a resposta;
  - HATEOAS "*hypermedia as the engine of application state*", este critério atua como um motor de navegação entre os recursos fornecidos pelo servidor, e diz que as respostas do servidor devem conter *hyperlinks* e instruções de navegações de acessos entre os recursos (FIELDING, 2000).

### 3 SOBRE A ORGANIZAÇÃO

Este capítulo é dedicado a apresentar a organização onde o estágio foi efetuado, seu modelo de negócio, organização das equipes, bem como, o modo como os serviços de desenvolvimento de software são realizados.

A Levty é uma empresa de desenvolvimento de software focada em transformação digital para empresas, onde o cliente contrata um ou mais times de desenvolvimento de software para atuar internamente em suas demandas, no modelo *team as a service*, de maneira pontual com projetos de curta duração, ou de maneira intermitente, sem um prazo determinado para o fim da prestação dos serviços. Sendo o principal objetivo de cada time auxiliar os clientes no processo de transformação digital de sua organização, aplicando tecnologias digitais, neste caso por meio da plataforma Sydle One, integrando os processos de negócio e o capital humano da empresa ao meio digital. Atualmente a Levty conta com escritórios em Belo Horizonte, Juiz de Fora, Viçosa e Itajubá, que atendem cliente nacionais e internacionais.

#### 3.1 Times

A Levty oferece equipes de desenvolvimento de software de acordo com a necessidade dos clientes e atualmente presta serviços para 32 clientes, com um total de 155 colaboradores. As equipes são compostas dois, três ou cinco integrantes, e alguns clientes possuem mais de uma equipe, onde cada uma pode atuar de maneira autônoma em diferentes projetos dentro da organização, os times geralmente são compostos por:

- *Agile Master*: é o profissional responsável por manter a equipe alinhada com os princípios e práticas ágeis. Ele também atua como facilitador, resolvendo problemas, geralmente juntamente ao cliente, que não estejam relacionados com o processo de desenvolvimento em si.
- *Product Owner*: é o profissional responsável por manter o desenvolvimento do projeto alinhado com a visão que o cliente tem do produto, atuando diretamente com os interessados pelo produto, incluindo os próprios usuários. Ele atua junto com a equipe na priorização do que será desenvolvido.
- Líder de Desenvolvimento: é um membro da equipe com maior senioridade, conhecimento técnico e de produto que também possui maior domínio das tecnologias e ferr-

mentas utilizadas no projeto. É a pessoa responsável por auxiliar os membros do time nos aspectos técnicos mais complexos de serem desenvolvidos.

- Desenvolvedor: Geralmente estão presentes em maior número na equipe e foi a função ocupada pelo estagiário durante o período de estágio. São os membros da equipe responsáveis pelo desenvolvimento e testes dos requisitos do produto.

### 3.2 Processo de Desenvolvimento

Na Levty os times seguem o modelo de desenvolvimento ágil Scrum, uma metodologia ágil baseada em ciclos de desenvolvimento curtos, denominadas *sprints*, onde a cada final de *sprint* espera-se a entrega de parte de produto, por menor que seja, já completa e funcional.

Em um primeiro momento, as novas necessidades para o produto são identificadas pelo *product owner* e são adicionadas ao *backlog* de desenvolvimento. Em seguida é realizada uma reunião com toda a equipe, chamada “reunião de soluções”, onde são discutidos os aspectos técnicos da demanda. Logo após, é realizado o planejamento da *sprint*, onde cada nova demanda é mensurada em termos de esforço pela equipe, em uma unidade de medida conhecida como *Story Point*, que é uma medida de esforço para se quantificar um item do backlog baseado no contexto atual do projeto. Depois disso as demandas são liberadas para o desenvolvimento.

Após o desenvolvimento e testes das demandas é feita uma homologação com os usuários finais, onde verifica-se se o que foi desenvolvido era realmente o que se esperava. Nesta etapa, também podem ser sugeridas mudanças ou até mesmo correções e, caso haja a aprovação dos usuários, há a disponibilização no ambiente de produção para que os usuários possam utilizar o sistema.

Existem ainda algumas reuniões que tem por objetivo acompanhar o andamento do projeto e verificar impedimentos e problemas que ocorreram durante a *Sprint*. Uma delas é a reunião diária, que é uma reunião que ocorre diariamente para o time explicitar o que está sendo desenvolvido e se a equipe está com alguma dificuldade ou empecilho. Outra reunião realizada é a reunião de retrospectiva, onde há uma recapitulação dos principais pontos da *Sprint*. As principais etapas de desenvolvimento serão discutidas em mais detalhes no Capítulo 4.

## 4 ATIVIDADES REALIZADAS

Este capítulo trata das principais atividades desenvolvidas nos diferentes projetos onde houve atuação do estagiário no período do estágio.

### 4.1 Treinamento

Nas primeiras duas semanas, o estagiário participou do treinamento oferecido pela empresa para que houvesse a criação de uma base teórica e prática a respeito das atividades que são desenvolvidas nos projetos dos clientes. Além das palestras e treinamentos a respeito de metodologias ágeis, houve também um treinamento teórico e prático sobre a utilização da plataforma *Sydle One*, plataforma na qual a empresa atua desenvolvendo soluções digitais para os clientes.

Nessas duas semanas do treinamento também houve contato com funcionários já contratados da empresa e com outros estagiários já alocados em projetos. Dessa forma, foi possível aprender e entender um pouco mais sobre o dia a dia e as atividades desenvolvidas pelas equipes de desenvolvimento de software da empresa, assim como, ter a noção de como é a dinâmica de contato e atuação junto com os clientes.

### 4.2 Desenvolvimento de software em projetos reais

Após o treinamento, o estagiário passou a integrar uma equipe de desenvolvimento em um projeto real, desenvolvendo demandas das empresas clientes juntamente com uma equipe composta por um líder de desenvolvimento e outros desenvolvedores, de diferentes senioridades. O desenvolvimento foi feito utilizando-se de conceitos de metodologias ágeis, como por exemplo a divisão do tempo de desenvolvimento em *sprints*, onde desenvolve-se um certo número de demandas.

Os projetos foram desenvolvidos na plataforma *low-code Sydle One*, onde utiliza-se principalmente a linguagem de programação *javascript*, e consistiram em sua maioria na automatização de processos de negócio utilizando-se o sistema de BPM da plataforma.

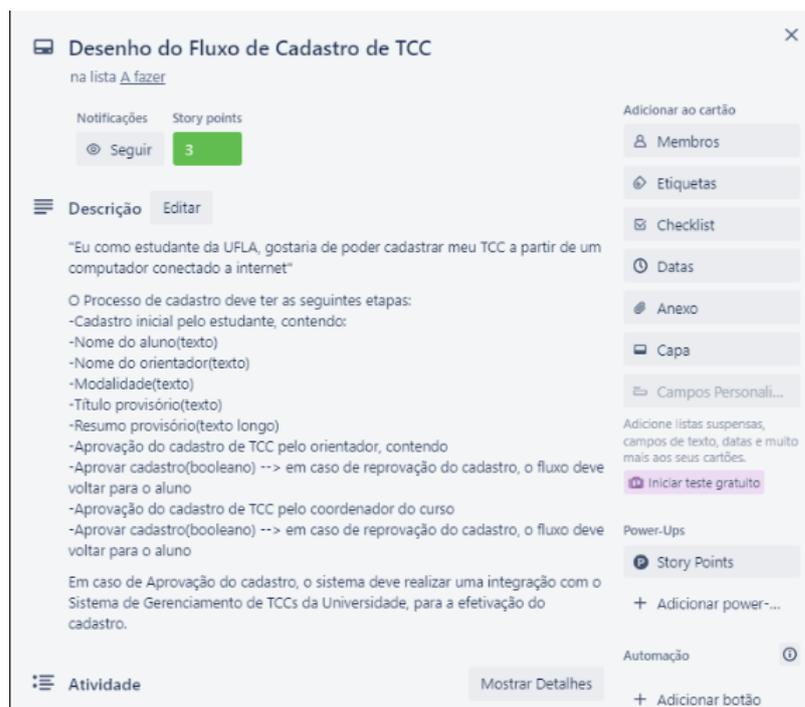
Vale ressaltar que detalhes dos projetos e das implementações não foram discutidos devido a confidencialidade dos clientes, sendo detalhados apenas a visão geral do desenvolvimento.

#### 4.2.1 Levantamento de requisitos de software

O primeiro e um dos mais importantes passos para se desenvolver software de maneira profissional é o levantamento dos requisitos. É nesta etapa que entende-se, juntamente com os futuros usuários e demais partes interessadas, o que deve ser desenvolvido e quais são as necessidades a serem sanadas com o sistema.

No contexto do estágio na Levty, geralmente as conversas iniciais ocorrem entre o líder de desenvolvimento da equipe e as partes interessadas nas demandas, gerando Estórias de Usuário que serão adicionadas ao *Backlog* de desenvolvimento e sua gestão, no contexto dos projetos da Levty, pode ser feita em diversas ferramentas a depender das preferências do cliente e da equipe, sendo as mais comuns o Trello, Azure Devops e o Jira. Em seguida há o envolvimento dos outros membros do time, principalmente para discussões a respeito dos aspectos técnicos dos requisitos e para mensurar a complexidade de cada requisito. Essas duas discussões geralmente ocorrem em reuniões chamadas de Reunião de Soluções e Reunião de Planejamento. Na Figura 4.1 é mostrado um exemplo de Estória de Usuário escrita na ferramenta Trello e utilizando-se um template que foi que é utilizado por diversos times da Levty e consiste na seguinte estrutura "Eu como <ator> gostaria de <necessidade>".

Figura 4.1 – Exemplo de Estória de Usuário



Fonte: Do autor (2023)

Vale ressaltar que não necessariamente as demandas chegam primeiro ao líder de desenvolvimento. Toda a equipe tem a liberdade de sugerir novos requisitos a partir do momento que novas necessidades são identificadas durante o ciclo de desenvolvimento do software, assim como, há a sugestão de melhorias ou de alterações que devem ser discutidas com todos os interessados na solução.

#### **4.2.2 Desenvolvimento de software - BPM**

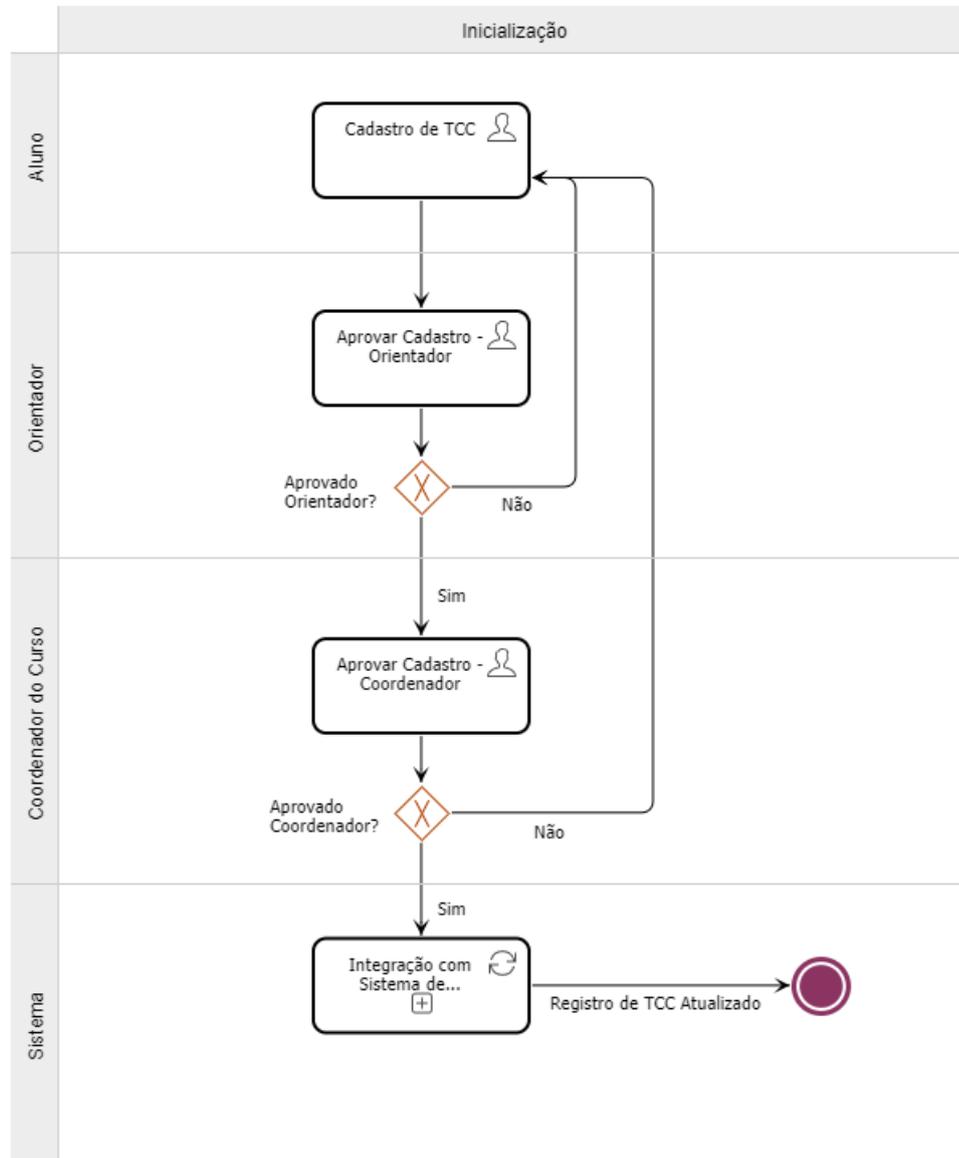
Dentre as principais demandas de desenvolvimento em que o estagiário atuou, encontram-se a automatização de processos BPM, que consiste em trazer para dentro de algum sistema de software, no caso o *Sydle One*, processos que são ou seriam realizados de forma manual na organização do cliente, visando o ganho de tempo, aumento de produtividade e consequentemente uma maior eficiência de custos para o cliente. Outras vantagens secundárias incluem a facilidade de se manter o registro das atividades realizadas, que não seriam fáceis de serem mantidos em processos manuais, e a geração de dados que podem ser analisados e transformados em informação útil para o cliente.

##### **4.2.2.1 Desenho do Fluxo**

Primeiramente é feito o desenho do fluxo utilizando a ferramenta de edição de diagramas, em formato BPMN. O desenho do fluxo é realizado de acordo com o que foi levantado com os usuários. É importante ressaltar que, além das informações do levantamento de requisitos, o fluxo pode representar elementos de cunho mais técnico e desacoplados do escopo das regras de negócio, cabendo ao desenvolvedor analisar cada situação e produzir um desenho que atenda a todos os requisitos de negócio definidos pelos usuários e, também, atenda aos requisitos técnicos do fluxo, que nem sempre são de conhecimento do usuário. Na Figura 4.2 é mostrado um exemplo de um fluxo de processo em BPMN que representa o processo hipotético de solicitação de TCC pelo aluno.

Figura 4.2 – Exemplo de fluxo de processo BPM no Sydle One

Processo: Solicitação de TCC - 1.0



Fonte: Do autor (2023)

#### 4.2.2.2 Configuração do Fluxo

Após a criação do diagram BPMN do fluxo, configura-se o processo para que este funcione de maneira automática, alcançando o objetivo do cliente de ter processos automatizados. Dentre as configurações tem-se, principalmente:

- Criação dos campos dos processo: Os campos do processo funcionam como variáveis globais que podem ser acessadas em qualquer elemento dentro do fluxo.
- Configuração dos *gateways*: os *gateways* do *Sydle One* oferecem vasta possibilidade de customização facilitando, assim, que o processo se adeque exatamente as regras de negócio propostas para o fluxo. A configuração é feita nos ramos do fluxo que saem do *gateway*. Um ramo pode ser definido como padrão, que será a escolha quando nenhum outro ramo atender suas condições de escolha. Também é possível configurar um script que retorne um booleano que represente se o ramo em questão será seguido ou não, dando assim, flexibilidade de configuração. O ramo também pode não ter configuração e neste caso ele sempre será seguido.
- Configuração das atividades de usuário: As Atividades de usuário são as etapas onde ocorrem algum tipo de atendimento que deve ser realizado por um usuário. Dentro delas pode-se configurar seus campos, podendo ser adicionados campos que pertencem ao processo e, também, é possível configurar seus métodos
- Configuração das atividades de *Script*: Também é possível adicionar atividades que executam um *script*, escrito em *javascript* em qualquer lugar do fluxo. Este elemento é de suma importância para uma melhor automação do processo, sendo muito utilizado para realizar todas as atividades que não necessitam de atuação humana.

#### 4.2.2.3 Escrita dos Scripts

*Scripts* são elementos fundamentais dentro da plataforma *Sydle One*. Eles fornecem flexibilidade para a plataforma, já que aceitam qualquer código escrito em *Javascript*, única linguagem de programação aceita até o momento de escrita deste trabalho. Os *scripts* podem ser utilizados em diversos locais, dentre os quais alguns já descritos no item anterior. São amplamente utilizados dentro do *Sydle One* em classes, que semelhante a diversas linguagens de programação, são um tipo de dado que representa determinados tipos de objetos com características semelhantes.

Na figura 4.3 temos o exemplo de um *script* que implementa as regras de negócio para a criação do registro de um novo cliente:

Figura 4.3 – Exemplo de Script para criar registro de novo cliente

```

const moment = require("../_internal/moment");

const DOCUMENT_TYPE_CPF_ID = "5912258ff5ca530b54567bb4";
const DOCUMENT_TYPE_CNPJ_ID = "5912257df5ca530b545667f9";
const MARITAL_STATUS_SINGLE_ID = "5d139ba0cfa5ac1357c569b5";
const MARITAL_STATUS_MARRIED_ID = "5d139bafcf5ac1357c569c6";

if (_input.type === "person") {

  let personAge = getPersonAge(_input.birthDate);

  if (personAge < 18) {
    throw "Serviço cliente: Pessoa deve possuir 18 anos ou mais";
  }

  if (!crm.document.validateCPF({"cpf": _input.cpf}).success) {
    throw "Serviço cliente: Campo 'cpf' é inválido";
  }

  if (personAge > 40 && !_input.maritalStatus) {
    throw "Serviço cliente: Campo 'maritalStatus' obrigatório caso pessoa possuir 40 anos ou mais";
  }

  if (_input.maritalStatus && _input.maritalStatus !== "single" && _input.maritalStatus !== "married") {
    throw "Serviço cliente: Campo 'maritalStatus' aceita apenas os valores 'single' ou 'married';
  }
} else if(_input.type === "enterprise") {

  if (!crm.document.validateCNPJ({"cnpj": _input.cnpj}).success) {
    throw "Serviço cliente: Campo 'cnpj' é inválido";
  }
} else {
  throw "Serviço cliente: Campo 'type' aceita apenas os valores 'person' ou 'enterprise';
}

let newCustomer = createCustomer();

```

Fonte: Do autor (2023)

Dentro de uma classe nem todos os *scripts* podem ser modificados pelo desenvolvedor. Os principais *scripts* que permitem edição são:

- Criar Rascunho: É executado ao início da criação de um objeto. É muito utilizado para preencher dados e também executar trechos de código que devem ser executados neste momento da criação do objeto.
- Atualizar Rascunho: É executado no momento em que se altera o objeto, que ocorre quando algum de seus valores é modificado. Deve-se ter cuidado para que determinados trechos de código, que podem ter um maior custo de processamento, não sejam executados em toda mudança no objeto. Isto é possível utilizando técnicas como, por exemplo, adicionar uma condicional para executar determinado trecho de código, somente se ocorrer alguma mudança específica no objeto.

- **Obter Metadata:** Também é executado a todo momento em que ocorre alguma mudança no objeto, porém, diferente do *script* de “Atualizar Rascunho”, pode-se acessar as propriedades gráficas da tela de edição do objeto. Este *script* é principalmente utilizado para controlar as propriedades de formulário, alterando a visibilidade, obrigatoriedade, dentre outras propriedades dos atributos do objeto.
- **Salvar:** É executado ao se finalizar a edição do objeto, no momento em que o estado do objeto é salvo pela plataforma.
- **Métodos Customizados:** O desenvolvedor também tem a liberdade para criar diferentes métodos customizados e que podem ser chamados em qualquer outro *script* dentro da plataforma, podendo ser chamado por *script* em objetos da própria classe, objetos de classes diferentes e em diversos outros lugares dentro da plataforma.

#### 4.2.3 Integrações com APIs externas

A maioria dos clientes da Levty possuem outros sistemas os quais fornecem e também podem necessitar de dados gerados no *Sydle One*, sendo assim, um importante recurso é a possibilidade de realizar integrações via API com estes sistemas. Como boa prática de programação dentro da Levty, os times costumam desenvolver suas integrações dentro de processos BPM específicos para a integração, assim cada chamada de API é facilmente registrada, logs são mais fáceis de serem rastreados e o tratamento de erro é padronizado.

Como explicado na seção anterior, é possível escrever diferentes tipos de *script* em atividades de *script* em processos BPM no *Sydle One*, assim, no contexto de uma integração, escreve-se um *script* que será o responsável por orquestrar a integração. Além disso, é necessário criar uma classe para servir de fachada para a integração, ou seja, implementar as requisições que podem ser feitas e disponibilizá-las para serem utilizadas em toda a plataforma, a partir de uma chamada de método, como explicado na sub-seção anterior. Por fim, dentro do método desenvolvido na fachada, monta-se a requisição e chama-se o *web service* nativo da plataforma, que é uma classe que disponibiliza a comunicação, via HTTP, dentro do *Sydle One*.

A depender do código HTTP de resposta da requisição, o fluxo do processo pode tanto seguir para o fim, em caso de sucesso, ou alguma outra situação de finalização, quanto seguir para um ramo que chegará na atividade de *script* responsável pela integração, realizando uma nova tentativa de efetuar a integração com sucesso.

#### 4.2.4 Teste de software

Após o desenvolvimento de uma Estória de usuário é importante que o que foi desenvolvido seja devidamente testado, a fim de reduzir ao máximo o número de problemas no ambiente de produção, utilizado pelo usuário final.

Os erros na solução podem ser desde simples problemas, que afetam apenas a experiência do usuário não causando consequências mais graves, até problemas que podem vir a causar prejuízos financeiros e de imagem para a empresa cliente. Visando evitar este tipo de situação, todos os projetos desenvolvidos pela Levty passam por etapas de teste de software visando aumentar a qualidade das entregas do projeto. Para isso são aplicados alguns tipos de teste de software, listados a seguir:

- **Teste de Aceitação:** É o modelo de teste mais simples, porém, ajuda a evitar prematuramente a maioria das falhas. Consiste em replicar manualmente o comportamento do usuário final, simulando o uso do software em caso real, ou seja, é feito com o desenvolvedor utilizando o sistema da maneira como o usuário utilizaria. É importante validar se as regras de negócio foram corretamente desenvolvidas na aplicação e que não há nenhuma falha técnica durante o procedimento. Um ponto importante a ser levado em consideração é o de não viciar o teste apenas com casos bases de uso, ou seja, o desenvolvedor que está realizando o teste, deve fazê-lo de maneira mais fiel ao uso de um usuário real, não utilizando apenas casos de teste cujo os resultados esperados são triviais.
- **Teste de Integração:** São os testes responsáveis por validar se diferentes partes do sistema, que possuem recursos utilizados por outras partes, estão funcionando não apenas de maneira isolada, mas também funcionando quando são utilizadas por estas outras partes. São testes cruciais, principalmente, em classes utilitárias que possuem métodos que são chamados em diversos outros pontos do software. O objetivo do teste de integração é garantir que não ocorram falhas advindas dessa integração.
- **Teste de Carga:** É um tipo de teste realizado quando o objetivo é o de validar o desempenho de determinada parte da integração, quando esta é submetida a uma carga alta que demanda um maior processamento. É importante realizar este tipo de teste em APIs que recebem um grande número de requisições e em métodos que são executados para processamento de diversos objetos simultaneamente.

#### 4.2.5 Homologação e Deploy

Após o desenvolvimento e os testes, chega-se a etapa de homologação e posteriormente ao *deploy* da unidade de software desenvolvida. Na homologação tem-se o sistema sendo utilizado pelo usuário final em um ambiente apartado do produtivo, para que assim, ele consiga simular casos de usos real, porém sem impactos na aplicação de produção. É nesta etapa que os interessados finais no sistema podem validar se o software reflete corretamente as regras de negócio e se a usabilidade está dentro do esperado. Esse *feedback* pode originar novas demandas de desenvolvimento, com melhorias no sistema, ou até mesmo a necessidade de ajustes e correções, sendo assim, faz-se necessário que o desenvolvedor estimule ao máximo o usuário a dar suas opiniões a respeito do sistema.

Caso a unidade do sistema que foi desenvolvida seja devidamente homologada e aprovada pelos usuários finais, chega-se o momento de disponibilizá-la em ambiente produtivo, para que possa ser utilizada em casos de uso reais. Este procedimento de disponibilização tem o nome de *deploy* e a plataforma *Sydle One* oferece um recurso nativo para esta finalidade por meio de arquivos de configuração proprietário chamados de *Sybox*. O desenvolvedor deve gerar um *Sybox* no ambiente de origem desejado, geralmente o ambiente de homologação, tendo o devido cuidado para selecionar as unidades do sistema que de fato devem ser exportadas, e em seguida, importá-lo no ambiente de destino.

## 5 CONCLUSÃO

O objetivo deste trabalho foi elucidar os principais pontos presentes durante o estágio em desenvolvimento de software do autor. Ao decorrer das seções pode-se perceber como as disciplinas lecionadas durante o curso de bacharelado em Ciência da Computação são aplicadas em projetos reais, como por exemplo as disciplinas de Introdução aos Algoritmos e a disciplina de Algoritmos e Estruturas de Dados, que ajudaram o estagiário a desenvolver o raciocínio lógico aplicado a programação, bem como as disciplinas de Engenharia de Software e a de Modelagem e Implementação de Software, onde o estagiário teve o primeiro contato com temas como Desenvolvimento Ágil, Levantamento de Requisitos de Software e Testes de Software. Possibilitando ao estagiário desenvolver suas habilidades na área e aprofundar ainda mais seu conhecimento adquirido durante o curso, aplicando-o em cenários reais de desenvolvimento de software.

Também é possível perceber como o estágio é capaz de enriquecer o currículo do estudante de Ciência da Computação, como por exemplo, permitindo-o ter contato com tecnologias e ferramentas não visitadas durante o curso, neste caso, a plataforma *low code Sydle One*. Além disso, o estagiário pôde adquirir habilidades não técnicas, as quais não são possíveis de serem amplamente abordadas em sala de aula, como o trabalho em equipe e desenvolver projetos com outras pessoas que não necessariamente tem conhecimento técnico na área da Computação.

Além disso, durante o desenvolvimento das atividades do estágio com a utilização da plataforma *low-code Sydle One*, o estagiário passou a ter uma melhor compreensão acerca do assunto, entendendo e vendo os ganhos de produtividade advindos da utilização de tal tecnologia, a facilidade durante o desenvolvimento e a menor curva de aprendizado se comparado ao desenvolvimento com linguagens de programação tradicionais, bem como a percepção de bons casos de uso para este tipo de tecnologia, como por exemplo a automatização de processos de negócio das empresas, ou seja, no desenvolvimento de ferramentas corporativas que são utilizadas internamente nas empresas pelos seus funcionários. Também podemos citar exemplos de casos onde o desenvolvimento tradicional de software pode ser melhor aproveitado, como no desenvolvimento de um aplicativo móvel para ser utilizado pelos clientes da organização e nas vantagens desse tipo de desenvolvimento, onde os desenvolvedores tem uma maior liberdade para desenvolver recursos, não ficando restritos a utilizar apenas os fornecidos pela plataforma utilizada, além de terem acessos a funcionalidades desenvolvidas por terceiros, como *frameworks* de desenvolvimento e bibliotecas de código aberto .

## REFERÊNCIAS

- ALVES, F. R.; ALCALÁ, S. G. S. Análise da abordagem low-code como facilitador da transformação digital em indústrias. **Revista e-TECH: Tecnologias para Competitividade Industrial - ISSN - 1983-1838**, v. 15, n. 1, 2022. Disponível em: <<https://etech.emnuvens.com.br/revista-cientifica/article/view/1186>>.
- AWS, A. **O QUE É UMA API?** 2022. <<https://aws.amazon.com/pt/what-is/api/>>.
- BECK, K. et al. **Manifesto for Agile Software Development**. 2001. <<https://agilemanifesto.org/>>. Accessed on February 27, 2023.
- DAEHN, C.; COSTA, A.; PEREIRA, R. Transformação digital e sustentabilidade: desafios e tendências. In: **XXIII Engema**. [S.l.: s.n.], 2021. p. 1–16.
- FIELDING, R. T. **Architectural styles and the design of network-based software architectures**. [S.l.]: University of California, Irvine, 2000.
- GAIWORONSKAYA, Y. V. et al. Typology of risks and threats caused by digitalization. **Revista Turismo Estudos e Práticas-RTEP/UERN**, n. 5, p. 1–12, 2020.
- MACHADO, J. **Team as a Service: uma mudança de mentalidade**. 2021. <<https://www.computerworld.com.pt/2021/05/19/team-as-a-service-uma-mudanca-de-mentalidade/>>.
- PRESSMAN, R. S. **Engenharia de Software: uma abordagem profissional**. 9. ed. Porto Alegre: AMGH, 2021.
- ROGERS, D. L. **Transformação digital: repensando o seu negócio para a era digital**. 1. ed. São Paulo: Autêntica Business, 2017.
- VALENTE, M. T. **Engenharia de Software Moderna**. [S.l.: s.n.], 2020.
- WASZKOWSKI, R. Low-code platform for automating business processes in manufacturing. **IFAC-PapersOnLine**, v. 52, n. 10, p. 376–381, nov. 2019. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2405896319309152>>.
- WHITE, S. A. Introduction to bpmn. **BPTrends**, p. 1–14, jul. 2004. Disponível em: <<https://www.bptrends.com/bpt/wp-content/publicationfiles/07-04%20WP%20Intro%20to%20BPMN%20-%20White.pdf>>.