



ARLAN DA SILVA TRINDADE

**RELATÓRIO DE ESTÁGIO - SITE DE LOCALIZAÇÃO DE
PESSOAS LPP**

LAVRAS – MG

2023

ARLAN DA SILVA TRINDADE

RELATÓRIO DE ESTÁGIO - SITE DE LOCALIZAÇÃO DE PESSOAS LPP

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.

Prof. Dr. Julio Cesar Alves

Orientador

LAVRAS – MG

2023

ARLAN DA SILVA TRINDADE

RELATÓRIO DE ESTÁGIO - SITE DE LOCALIZAÇÃO DE PESSOAS LPP

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.

APROVADA em 04 de Dezembro de 2023.

Prof. Dr. Julio Cesar Alves UFLA
Prof. Dra. Renata Teles Moreira UFLA
Prof. Dr. Bruno de Abreu Silva UFLA

Prof. Dr. Julio Cesar Alves
Orientador

**LAVRAS – MG
2023**

Dedico à toda minha família, principalmente a minha vó Terezinha Ribeiro de Jesus Silva.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por sempre me abençoar e guiar em meus caminhos. Agradeço a minha vó Terezinha Ribeiro de Jesus Silva por sempre me apoiar e não ter me deixado desistir, a minha mãe Ivanete Aparecida da Silva, por sempre ter me ajudado quando precisei. Também agradeço aos meus amigos Anna Flávia Lopes Ferreira, Larissa Narciso de Oliveira, Maycon Henrique Santos Adriano, Marcos Vinicius de Souza Pereira, Samir Avelino Carvalho e Welerson Francisco da Silva, por terem sempre me apoiado e ajudado nessa etapa da minha vida, e ao Professor Dr. Julio Cesar Alves por ter me orientado no processo de elaboração deste documento.

Por fim, quero agradecer à Universidade Federal de Lavras e a todos os professores e funcionários por tornar o processo da minha formação profissional e pessoal possível.

*“Um homem é tão grande quanto o seu sonho.”
“Peça a Deus que abençoe os seus planos, e eles darão certo.”(Provérbios 16:3)*

RESUMO

No cenário atual, vivenciamos uma era marcada pela rápida evolução tecnológica, onde as metodologias ágeis emergem como protagonistas na busca pela eficiência e inovação. As tecnologias ágeis, fundamentadas em princípios colaborativos, adaptação contínua e entrega iterativa, tornaram-se indispensáveis para empresas que buscam prosperar em um ambiente dinâmico e competitivo. O objetivo deste relatório é apresentar, do ponto de vista do desenvolvedor como é trabalhar nesse cenário dentro da empresa DTI Digital, com foco está principalmente no ciclo de uma sprint, apontando seus processos, tecnologias e dificuldades. A fim de auxiliar outras pessoas, principalmente alunos, a entender melhor o funcionamento desse ciclo e apresentando alguns paralelos com disciplinas cursadas no curso de Sistemas de Informação. O projeto realizado dentro do estágio trata-se da manutenção e melhoria de dois sites, LPP e MPP, que atendem um mesmo cliente e objetivo final. O Relatório conta também com alguns exemplos breves de atividades realizadas para um maior entendimento do contexto apresentado.

Palavras-chave: Metodologias ágeis, Sprint, Scrum.

ABSTRACT

In the current scenario, we are experiencing an era marked by rapid technological evolution, where agile methodologies emerge as protagonists in the search for efficiency and innovation. Agile technologies, based on collaborative principles, continuous adaptation and iterative delivery, have become indispensable for companies seeking to thrive in a dynamic and competitive environment. The objective of this work is to present from the developer's point of view what it is like to work in this scenario within the company DTI Digital, the focus is mainly on the sprint cycle, pointing out its processes, technologies and difficulties, in order to help other people, especially students to better understand the functioning of this cycle and presenting some parallels with subjects taken in the Information Systems course. The project carried out within the internship involves the maintenance and improvement of two websites LPP and MPP, which serve the same client and final objective. The Report also includes some brief examples of activities carried out to provide a greater understanding of the context presented.

Keywords: Agile methodologies, Sprint, Scrum.

LISTA DE FIGURAS

Figura 1.1 – Tela inicial LPP	11
Figura 1.2 – Tela de busca LPP	11
Figura 1.3 – Tela de viagens LPP	12
Figura 1.4 – Tela de dashboard MPP	12
Figura 3.1 – Diagrama do ciclo da sprint	18
Figura 3.2 – Exemplo de quadro Azure devOps	20
Figura 3.3 – Exemplo apontamento kick off	21
Figura 3.4 – Adicionando teste no cartão na azure	22
Figura 3.5 – Item do roteiro de testes	22
Figura 3.6 – Exemplo apontamento desk check	23
Figura 3.7 – Exemplo de um PR	24
Figura 3.8 – Exemplo de um quadro da Retro	26
Figura 4.1 – Mensagem e-mail inválido	28
Figura 4.2 – Mensagem Primeiro caractere do e-mail inválido	29
Figura 4.3 – Tela de métricas	29
Figura 4.4 – Exemplo componente em Angular	29
Figura 4.5 – Exemplo código em Angular	30
Figura 4.6 – Datepicker	31
Figura 4.7 – Mapa com cercas	31
Figura 4.8 – Dicionário	32
Figura 4.9 – Chamada do método	32
Figura 4.10 – Método highlightPolygon	32
Figura 4.11 – Mapa com cerca destacada	33

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Sobre a empresa	9
1.2	Projeto	10
1.3	Objetivo	11
1.4	Organização do texto	13
2	Tecnologias de desenvolvimento	14
2.1	Angular	14
2.2	DotNet	14
2.3	Postman	15
2.4	Git	15
2.5	Azure	16
3	Processo de desenvolvimento	17
3.1	Scrum	17
3.2	Sprinting planning	19
3.2.1	Pontuação de histórias	19
3.3	Daily	19
3.4	Kick off	21
3.5	Roteiro de testes	22
3.6	Desk Check	22
3.7	Code review	23
3.8	Teste em QA	24
3.9	Publicação em produção	25
3.10	Retro	25
3.11	Sprint Review	26
4	Exemplos de atividades realizadas	28
4.1	Validação de e-mail	28
4.2	Criação da tela de métricas	29
4.3	Destaque da cerca selecionada	30
5	CONCLUSÃO	34
	REFERÊNCIAS	35

1 INTRODUÇÃO

O presente relatório visa relatar a experiência de estágio vivenciada na empresa DTI Digital¹. O foco central deste trabalho consiste na apresentação dos processos envolvidos em uma *sprint*. Ao longo deste relatório, serão apresentados os procedimentos técnicos e metodologias adotadas durante uma *sprint* de desenvolvimento, evidenciando exemplos concretos que ilustram os processos. Adicionalmente, são discutidas as dificuldades encontradas ao longo do estágio, proporcionando uma visão das nuances do ambiente profissional e contribuindo para o enriquecimento do conhecimento acadêmico e prático. Portanto, o relatório busca não apenas documentar as experiências vividas, mas também analisar criticamente o impacto dessas vivências no processo de formação profissional do estagiário, visando contribuir para a compreensão aprimorada dos processos ágeis por parte de outros alunos em formação.

1.1 Sobre a empresa

A DTI digital é uma empresa brasileira criada em 2009, em Belo Horizonte, com mais de mil funcionários. O principal ramo de atuação da empresa é a criação de soluções digitais para empresas para acelerar a eficiência digital e, para isso, ela usa métodos como: *Design* de experiência, gestão de produtos e engenharia de software (DTI, 2023). A DTI digital foi adquirida pela WPP², que é uma grande empresa de publicidade e relações públicas com sede no Reino Unido que controla várias agências de comunicação em todo o mundo.

A estrutura segue uma estrutura horizontal, resumidamente os funcionários são organizados em squads, tribos e alianças. Squads são uma equipe multidisciplinar, tribos são formadas de vários squads, e alianças são formadas por várias tribos que atuam de maneira independente mas com o mesmo propósito.

A DTI Digital ilustra sua competência em colaborações bem-sucedidas ao aprimorar o suporte a pequenos empreendedores. Em alguns cenários, a empresa desenvolveu aplicativos ágeis e intuitivos como parte de sua habilidade em criar soluções personalizadas dada a sua natureza ágil de desenvolvimento. Um desses processos ágeis é a base deste relatório.

O estágio na empresa está sendo realizado no período de 17/01/2023 com data prevista de término 30/01/2024.

¹ <https://www.dtidigital.com.br/>

² <https://www.wpp.com/>

1.2 Projeto

O projeto que é relatado neste relatório trata do desenvolvimento e manutenção de dois sites para uma empresa cliente da DTI digital. Um destes sites é de localização de pessoas chamado *Locate People Portal* (LPP)³ e um de gestão de pessoas e dispositivos chamado de *Management People Portal* (MPP)⁴. O LPP exibe informações de pessoas localizadas dentro de uma área delimitada chamada de cercas virtuais. Essa localização é feita por meio de crachás que são passados em cancelas ou dentro de ônibus e dispositivos individuais que fazem comunicação via satélite para enviar a localização em tempo real do usuário. Já o MPP faz a exibição e gestão de dados de pessoas e dos dispositivos, permitindo a associação de um crachá para uma pessoa e a alteração de dados pessoais da mesma.

A estrutura geral do projeto⁵, consiste em várias APIs que processam eventos recebidos, e enviam esses eventos para outras apis que os armazenam em banco de dados. Depois que esses eventos são processados e armazenados, ele são usados por outras apis para serem disponibilizados no portal.

O principal foco dessa solução é ajudar em casos de acidentes e desastres, permitindo o levantamento de dados, tais como: quem estava dentro da área da empresa cliente durante o ocorrido? Onde essas pessoas estão localizadas? Quem são essas pessoas? quem deve ser contactado? Em seguida são mostradas algumas telas e funções dos portais. A Figura 1.1, por exemplo, mostra a tela inicial do LPP, que contém algumas informações sobre a localização das pessoas. A esquerda mostra o total de localizações, e por quais dispositivos as pessoas foram localizadas, e a direita mostra o mapa, e uma cerca (linha amarela) que delimita a área de localização. Já na Figura 1.2 mostra uma tela de busca; do sistema, onde o usuário é capaz de pesquisar uma pessoa e ver as últimas localizações dela no mapa, seu trajeto e com qual dispositivo ela foi localizada.

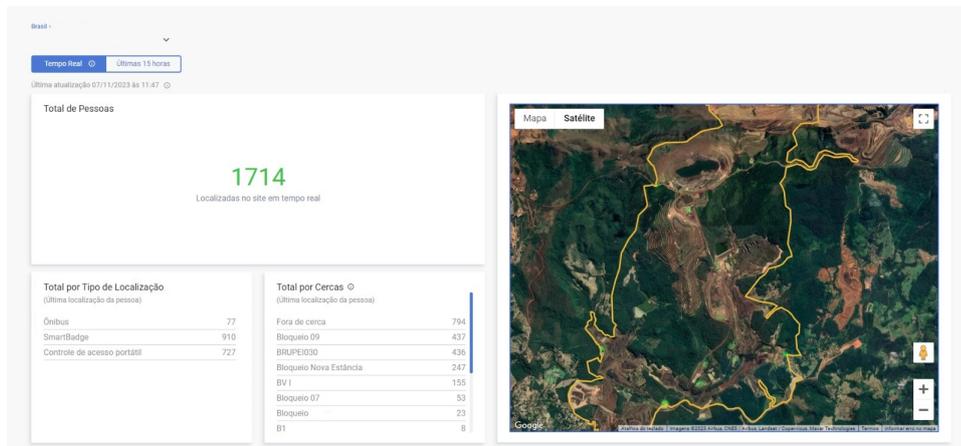
O projeto do LPP é muito grande, contém mais de 10 repositórios dentre repositórios de *front end*, *bff*, *apis* e *functions*, isso faz com que se tenha vários repositórios baixados no computador e, ainda é preciso baixar outros se caso a nova historia a ser desenvolvida tiesse um outro contexto. Então houve uma grande curva de aprendizado até se conhecer a estrutura geral do projeto.

³ O nome do portal foi alterado por motivos de segurança.

⁴ O nome do portal foi alterado por motivos de segurança.

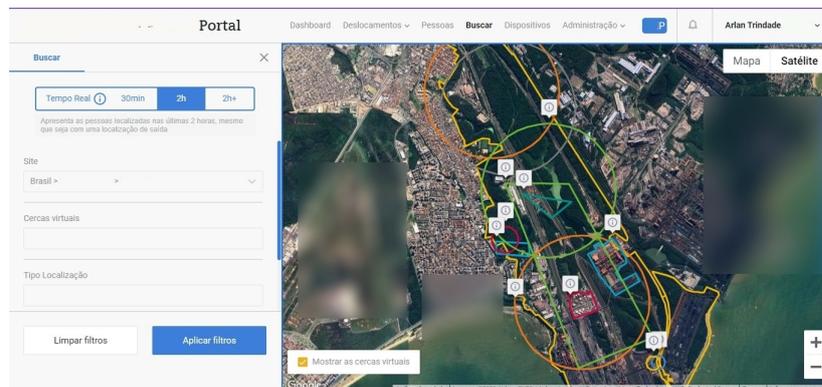
⁵ Detalhes foram omitidos por motivos de segurança.

Figura 1.1 – Tela inicial LPP



Fonte: Autor 2023

Figura 1.2 – Tela de busca LPP



Fonte: Autor 2023

A Figura 1.3, apresenta uma tela do LPP que mostra várias informações das viagens de ônibus que circulam dentro daquela área. A Figura 1.4 apresenta um DashBoard Gerencial do MPP, que mostra informações relevantes para a área de negócios da empresa cliente.

A equipe do projeto é composta por mais de 5 desenvolvedores, por um Desenvolvedor líder, um *Scrum Master*, um arquiteto de software, e uma representante da empresa cliente. Ao longo do projeto muitas pessoas são alocadas e realocadas para suprir necessidades de negócios, então o tamanho da equipe variou bastante durante o período de estágio.

1.3 Objetivo

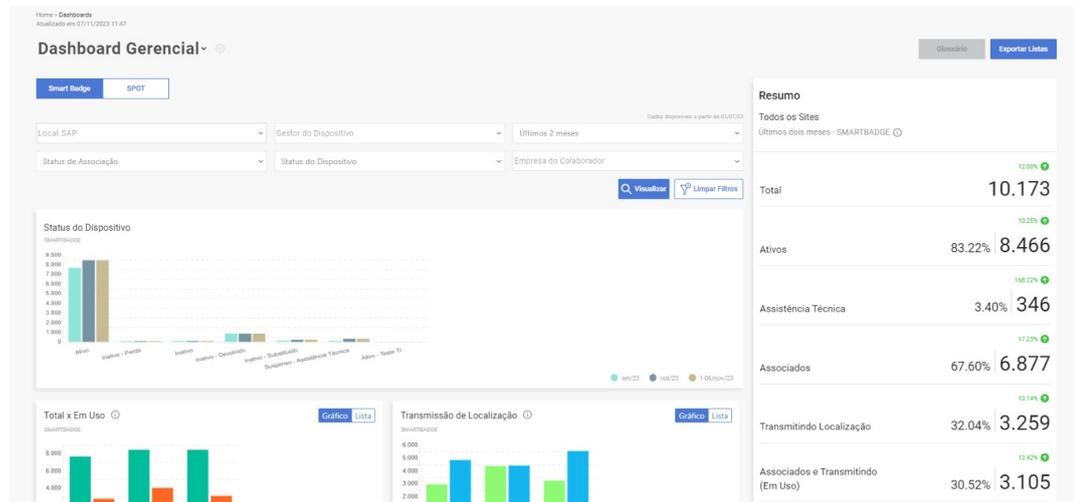
O objetivo deste relatório é mostrar um ciclo completo de *sprint* e as atividades realizadas durante esse ciclo, com a finalidade de apresentar uma visão abrangente para alunos interessados em realizar um estágio ou que não tiveram a oportunidade de fazê-lo anteriormente.

Figura 1.3 – Tela de viagens LPP

Nome da Linha	Telemetria	Código da viagem	Placa	Empresa Ônibus	Direção	Pessoas	Última localização do ônibus	Último registro do ônibus	Início da viagem	Fim da viagem	Pont
4463	Testemola	930799-133	GBR4E83	...	ENTROU	1	...	06/11/2023 16:55	06/11/2023 16:55	-	-20.34
0a434480-c9d3-48c9-9dc6-1b25263197a954cff15-2f81-4de8-9df9-ae428b8a3b36954cff15-2f81-4de8-9df9-ae428b8a3b368cddb5f3-5b23-	Testemola	b4042252-e555-4...	testemola	...	ENTROU	3	...	31/10/2023 16:45	31/10/2023 16:44	31/10/2023 16:45	-20.24
...	Testearlan	c96a4707-c148-4...	testearlan	...	ENTROU	3	...	31/10/2023 16:39	31/10/2023 16:38	31/10/2023 16:39	-20.24
...	Testearlan	fbcb32e8-28aa-4...	testearlan	...	ENTROU	2	...	31/10/2023 16:37	31/10/2023 16:36	31/10/2023 16:36	-20.24
...	31/10/2023	31/10/2023	31/10/2023	...

Fonte: Autor 2023

Figura 1.4 – Tela de dashboard MPP



Fonte: Autor 2023

Nesse contexto, descrevemos como é um ciclo de desenvolvimento de uma *sprint* em uma empresa como essa, destacando os principais pontos desse processo. Vale destacar, que o processo aqui relatado não segue completamente o padrão Scrum, pois alguns pontos foram adaptados para se adequar às necessidades da empresa. Além disso, o presente texto também aborda a importância e como o curso de Sistemas de Informação oferece ferramentas e conhecimentos abrangentes para auxiliar em toda a jornada de estágio, e destacar como isso foi aplicado nesse contexto específico do estágio.

1.4 Organização do texto

Além desta introdução, a estrutura deste documento se desdobra da seguinte maneira: o segundo capítulo discute as tecnologias empregadas no projeto durante o estágio, enquanto o terceiro capítulo aborda o processo personalizado da sprint dentro da empresa, considerando o contexto do projeto. No quarto capítulo, são apresentados exemplos de atividades realizadas, proporcionando uma compreensão mais ampla das histórias que são desenvolvidas no texto.

2 TECNOLOGIAS DE DESENVOLVIMENTO

Neste capítulo são abordadas as principais tecnologias usadas no desenvolvimento e, em particular, como são usadas no contexto do projeto. A disciplina de Engenharia de software garante uma base de conhecimento para entender o funcionamento em conjunto de algumas das tecnologias aqui apresentadas, em específico as de desenvolvimento do software, conhecimentos teóricos de padrões de projetos, diagramas e documentação.

2.1 Angular

No contexto do projeto do estágio, o *front end* dos portais é feito usando o *framework* Angular (ANGULAR, 2023), usando a linguagem de programação *typescript*. O Angular é um renomado *framework* de código aberto, desenvolvido e mantido pela Google, que se destaca pela sua versatilidade na construção de SPA (*Single Page Applications*), ou seja, Aplicações de Página Única. Uma característica marcante do Angular é que ele se baseia em *TypeScript*, um superconjunto de JavaScript que oferece benefícios como tipagem estática e recursos avançados de programação. Em resumo, o Angular é uma ferramenta poderosa para o desenvolvimento de aplicações web, oferecendo um conjunto abrangente de recursos e uma comunidade ativa de desenvolvedores.

2.2 DotNet

Dot net (MICROSOFT, 2023b) é uma plataforma de desenvolvimento de software de código aberto gratuita criada pela Microsoft, que une linguagem de programação e bibliotecas, permitindo o desenvolvimento de vários tipos de aplicativos como web, mobile desktop, dentre outros. No projeto do estágio, a plataforma Dot net é usada para criar aplicativos web. A linguagem de programação usada para para programar é C#, uma linguagem de programação orientada a objetos também criada pela Microsoft baseada na linguagem C. Todas as APIs, BFF e *Functions* do projeto são criadas com Dot NET.

Resumidamente, no contexto do projeto, as APIs são usadas para comunicação com os bancos de dados. Tais requisições sempre passam pelo BFF, que é um tipo de API, porém mais centrada em fazer serviços para o *front end*, como por exemplo fazer contas matemáticas baseadas nos dados retornados pelas APIs e enviá-los para o *front end*. E, por último, as *Functions*

também são uma espécie de API porém elas não esperam uma requisição, elas executam suas ações de forma independente.

2.3 Postman

O Postman (POSTMAN, 2023) é uma ferramenta essencial no mundo da desenvolvimento de API (Interface de Programação de Aplicativos) e é amplamente utilizada por desenvolvedores, testadores e equipes de desenvolvimento de software, principalmente por meio de Requisições HTTP, que são a base da comunicação na web. As requisições HTTP usam métodos para especificar a ação que o cliente deseja realizar. Alguns dos métodos mais comuns incluem:

- GET: Solicita a recuperação de dados de um servidor.
- POST: Envia dados para serem processados pelo servidor (como o envio de um formulário da web).
- PUT: Atualiza ou cria um recurso no servidor.
- DELETE: Remove um recurso no servidor.
- HEAD: Semelhante ao GET, mas solicita apenas os cabeçalhos da resposta, sem o corpo da mensagem.
- PATCH: Usado para aplicar modificações parciais a um recurso. É útil quando você deseja atualizar apenas parte de um recurso.

Durante o desenvolvimento das histórias do estágio, o Postman é usado para criar solicitações HTTP personalizadas para as APIs que precisam ser testadas durante o desenvolvimento das histórias. Pode-se, por exemplo, criar uma requisição de GET e verificar se os dados são retornados corretamente ou criar uma requisição POST e verificar se os dados são salvos corretamente no banco de dados.

2.4 Git

Git (VALENTE, 2020) é um sistema de controle de versão distribuído essencial para o desenvolvimento colaborativo. Ele gerencia mudanças em projetos ao longo do tempo usando

commits para representar registros do código, cada um com uma mensagem descritiva. A utilização de **branches** permite o desenvolvimento isolado de novas funcionalidades, evitando conflitos diretos. **Pull requests (PR)** facilitam revisões, discussões e aprovações antes da integração de alterações ao código principal, enquanto operações de **merge** combinam alterações de branches distintos. Em resumo, o Git simplifica o controle de versão e promove colaboração eficiente.

- **Commit:** Representa uma alteração específica em um conjunto de arquivos no repositório. Cada commit é acompanhado por uma mensagem descritiva das alterações realizadas. Fornece um histórico claro e compreensível do desenvolvimento do projeto.
- **Branch:** Ramificação isolada do desenvolvimento principal. Permite que desenvolvedores trabalhem em funcionalidades ou correções de bugs sem interferir diretamente no código principal.
- **Pull Request (PR):** Solicitação para revisão e mescla das alterações de uma branch específica no branch principal.
- **Merge:** Operação para combinar as alterações de uma branch com outra branch. Utilizado após a revisão e aprovação de um PR para incorporar as mudanças no branch principal.

2.5 Azure

A Azure (MICROSOFT, 2023a) é uma plataforma de serviços em nuvem criada pela Microsoft. No estágio são usados três serviços principais da plataforma Azure: serviços de banco de dados, infraestrutura como serviço e o devOps. Os serviços de banco de dados da Azure usados são o *Msql database*, que armazena os principais banco de dados do projeto e o Azure cosmos DB que é usado para salvar dados menos estruturados como *insights* do projeto, como por exemplo logins, erros do portal e transações mais importantes. Todo o projeto é hospedado na infraestrutura da Azure e todas as APIs são hospedadas em seus servidores, permitindo atualizar suas versões, executar ou parar algumas APIs quando necessário e os repositórios do projeto são salvos no Azure Repos. E, por último, todo o gerenciamento da *sprint* é feito por meio do Azure DevOps. É através dele que os desenvolvedores têm acesso aos quadros da *sprint*, *pipelines* e *releases*.

3 PROCESSO DE DESENVOLVIMENTO

Neste capítulo, exploraremos o ciclo de desenvolvimento da empresa em que o estágio está sendo realizado, abordando seus conceitos, ferramentas e processos, com ênfase especial na fase da *sprint*, alguns processos foram omitidos por não fazerem muito sentido a este relatório. Em metodologias ágeis, como o Scrum, no qual uma equipe de desenvolvimento trabalha em um conjunto específico de tarefas com objetivos claros. Uma "*sprint*" é um período de tempo fixo, geralmente com duração de duas a quatro semanas, durante a *sprint*, a equipe colabora intensamente para produzir um incremento do produto que pode ser potencialmente liberado para o cliente. A disciplina de Processos de Softwares fornece uma base teórica enriquecedora para todos os processos que serão abordados nas subseções a seguir, pois termos como *Scrum* (SCHWABER, 2020), *Sprint* e *Scrum Master* são comuns durante todo estágio em uma empresa como essa, e conhecer tudo isso de forma teórica facilita a adaptação do estagiário. A seguir são apresentados tópicos do processo de desenvolvimento de uma história, onde cada etapa complementa uma a outra, abordando as tecnologias usadas, pontos importantes de cada processo e detalhes, cada processo é dependente do anterior completando assim o processo maior que é a *sprint*.

3.1 Scrum

Scrum é uma metodologia ágil que promove a geração adaptativa de valor por meio de soluções eficazes para problemas complexos. Desenvolvido por Ken Schwaber e Jeff Sutherland na década de 1990, o Scrum é fundamentado em valores como empirismo e pensamento lean.

O Scrum, opera com base em uma série de princípios orientadores e papéis bem definidos: Papéis Principais:

- Scrum Master: Atua como facilitador, garantindo que a equipe compreenda e siga os princípios do Scrum, enquanto remove impedimentos que possam prejudicar o progresso.
- Product Owner: Responsável por maximizar o valor do produto, definindo prioridades e gerenciando o backlog.
- Desenvolvedores: Uma equipe multifuncional que converte itens do backlog em incrementos de valor durante os Sprints.
- Product Goal: Descreve um estado futuro desejado para o produto.

- Definition of Done: Estabelece critérios de qualidade e prontidão para entrega.

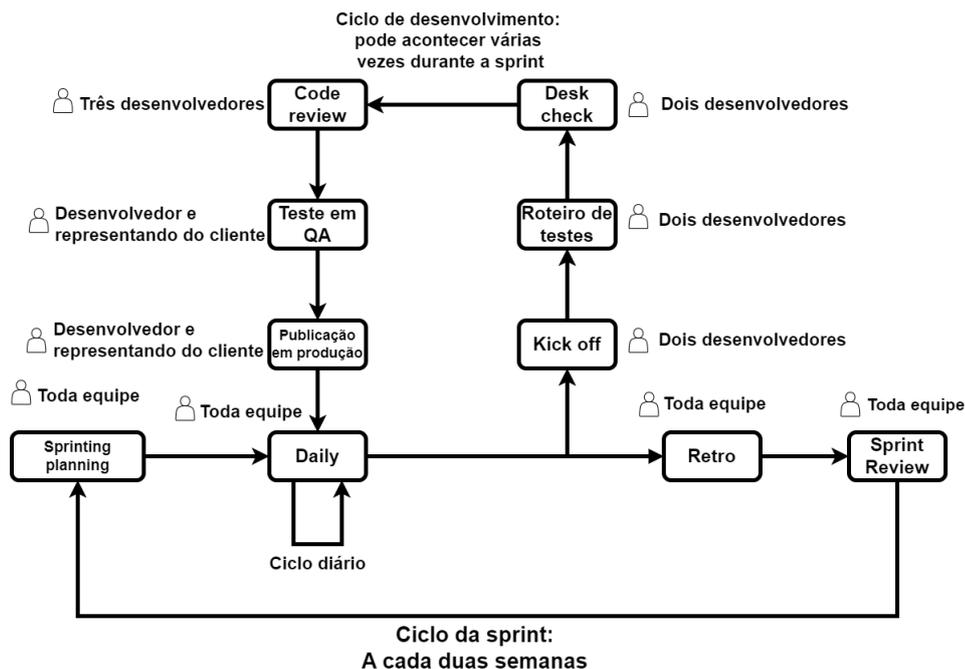
Os estágios do Scrum são dados por:

- *Sprint Planning*: Inicia o ciclo, definindo o trabalho a ser realizado e estabelecendo um objetivo claro.
- *Daily Scrum*: Ocorre diariamente para manter a equipe alinhada e facilitar a comunicação.
- *Sprint Review*: Avalia os resultados do Sprint, permitindo ajustes no *backlog* conforme necessário.
- *Sprint Retrospective*: Oferece uma oportunidade para a equipe refletir e aprimorar continuamente seus processos.

Essa abordagem iterativa e colaborativa do *Scrum*, combinada com os papéis claramente definidos e estágios bem estruturados, visa otimizar a entrega de valor de maneira eficiente e adaptável.

No contexto da empresa estagiada, dentro do projeto, os processos são dados por: *Sprint planning*, *Daily*, *Kick off*, *Roteiro de testes*, *Desk Check*, *Code review*, *Teste em QA*, *Publicação em produção*, *Retro*, *Sprint Review*, como mostrado na Figura 3.1.

Figura 3.1 – Diagrama do ciclo da sprint



Fonte: Autor 2023

3.2 Sprinting planning

A *sprint* começa com a *sprint* planning, que é uma reunião de uma a duas horas feita no começo da *sprint*. Esta reunião é muito importante, pois ela vai ditar quais histórias devem entrar na *sprint* e suas prioridades. Nela, no contexto da empresa diferente do Scrum e o Scrum *master* que apresenta as histórias que foram refinadas e que estão prontas para o desenvolvimento pela equipe presente na reunião e, nesse momento, a equipe pode tirar dúvidas sobre as histórias e regras de negócio que se aplicam a elas. Depois desse processo, é feita a pontuação das histórias, detalhada na subseção a seguir

3.2.1 Pontuação de histórias

A pontuação das histórias é muito importante para estimar o tempo de desenvolvimento de cada história. Muitos times optam pelo jogo conhecido como *planning poker*, no qual todos votam nas histórias secretamente e depois o valor de todo mundo é revelado. Dentro da equipe do projeto essa pontuação é feita de uma maneira parecida. A medida de pontos usada para classificar a complexidade das histórias são números fibonacci (1, 2, 3, 5...), e cada desenvolvedor vota em um valor dentre esses números com base no quão complexo ele considera a história, levando em conta todos os processos que vão do início ao fim do desenvolvimento. Existe uma história de referência que equivale a 1 ponto e os desenvolvedores a utilizam como base para pontuar a história que está sendo considerada. Ao final da pontuação, se houver uma diferença muito grande entre os votos, tanto para cima quanto para baixo, é feita uma discussão para sustentar os argumentos, com o objetivo de alcançar um acordo. Lembrando que se trata de uma estimativa, uma ideia do tempo e da complexidade que podemos esperar das histórias.

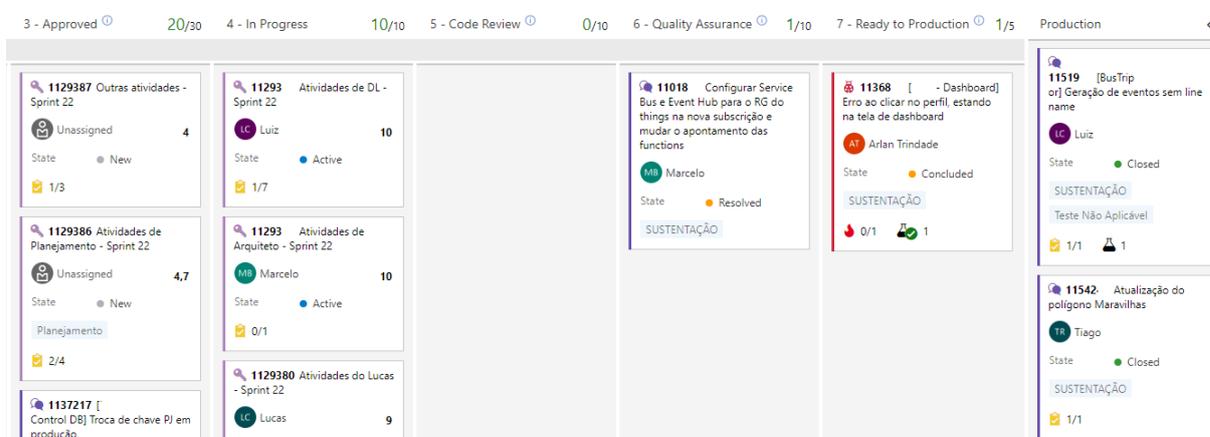
3.3 Daily

A daily é uma reunião diária de 15 minutos, que envolve toda a equipe do projeto. O objetivo dela, durante o estágio, era atualizar o andamento do desenvolvimento da história no dia anterior, indicando os passos que foram executados e as dificuldades encontradas. Nela também há a possibilidade de tirar dúvidas ou requisitar ajuda para o desenvolvimento.

Durante essa reunião é utilizado um “quadro” no Azure DevOps onde é mostrado em qual etapa do desenvolvimento cada história de cada desenvolvedor está. A Figura 3.2 exemplifica como este quadro se apresenta. As etapas são subsequentes e são dadas por:

- *Ready*: É quando a história está aprovada e pontuada para ser desenvolvida.
- *In progress*: É o estado que a história está sendo desenvolvida, o desk check também entra nessa etapa (à frente há um tópico explicando do que se trata o desk check).
- *Code review*: Essa etapa é quando o desenvolvimento da história está completo e seu teste funcional já foi feito, e é onde se faz uma auditoria do código que foi escrito.
- *Quality assurance*: Nessa etapa é feito o teste com o representante do cliente em ambiente de QA.
- *Ready to production*: Essa etapa é quando a história já está pronta para ir para o ambiente de produção.
- *In production*: Essa é a última etapa que indica que a história já foi finalizada e está disponível no projeto.

Figura 3.2 – Exemplo de quadro Azure devOps



Fonte: Autor 2023

Depois de cada integrante da equipe falar da atualização sobre suas atividades executadas, é utilizado um *dashboard* que exibe irregularidades dos cartões das histórias, como por exemplo, histórias sem *kick off*, sem roteiro de testes e sem *code review*. Essa parte é toda automatizada pelo Azure DevOps (MICROSOFT, 2023c), que aponta isso com base nos cartões. Ainda na *Daily* também é apresentado o progresso da *sprint*. Dada a sua pontuação total e a capacidade diária de toda a equipe é possível saber se a *sprint* está atrasada ou não e, com isso, pode-se traçar planos para destravar a *sprint* em caso de atraso ou aproveitar mais o tempo da *sprint* em caso de adiantamento, adicionando mais histórias a ela. A *Daily* também é usada

para dar avisos da gerência, atualizações, avisos pessoais, desligamento ou interrompimento de algum recurso para testes.

3.4 Kick off

O "*kick off*" é uma reunião de pouco mais de 15 minutos que envolve dois desenvolvedores. Ela é feita quando um dos desenvolvedores está pronto para iniciar uma nova história, com isso ele convida outro desenvolvedor para realizar o *kick off* junto a ele. Durante essa reunião, o objetivo principal é fazer um repasse completo da história que será desenvolvida pelo programador responsável. Esse é um estágio fundamental no desenvolvimento de software porque visa alinhar às expectativas, garantir que todos compreendam os requisitos e objetivos da história e, assim, estabelecer uma base sólida para o trabalho a ser realizado.

Além disso, o *kick off* é uma oportunidade para esclarecer dúvidas relacionadas às regras de negócios, requisitos funcionais e técnicos, bem como discutir possíveis sugestões para melhorar o processo de desenvolvimento. Os desenvolvedores podem aproveitar essa reunião para trocar informações, compartilhar conhecimento e definir os próximos passos. Ao final deve ser apontado o desenvolvedor com o qual foi feito *Kick off* no cartão referente a história, como na Figura 3.3.

Figura 3.3 – Exemplo apontamento kick off



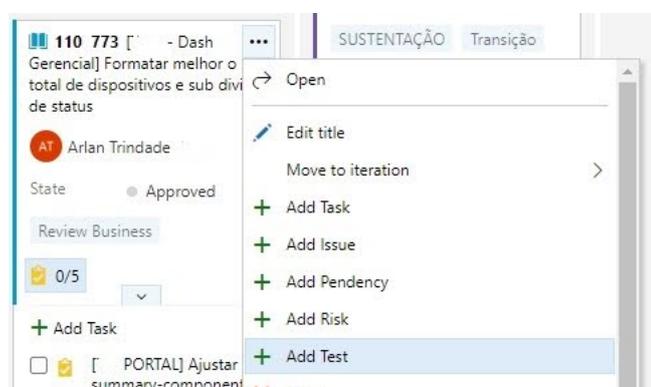
Fonte: Autor 2023

No geral, o *kick off* desempenha um papel crucial na garantia de uma colaboração eficaz entre os membros da equipe de desenvolvimento, promovendo a compreensão mútua e a entrega de software de alta qualidade. É um ponto de partida importante que contribui para o sucesso da *sprint*.

3.5 Roteiro de testes

Depois de realizado o *kick off* é feito o roteiro de testes, que consiste em uma lista com o que deve ser executado e o resultado esperado. Ele é feito com base nas entradas e saídas esperadas pela história, mas também pode ser incrementado com alguma verificação que não faz parte do desenvolvimento da história, mas que garante o funcionamento de uma parte delicada que a história pode interferir. Ele é adicionado ao cartão da história na Azure DevOps como mostrado na Figura 3.4.

Figura 3.4 – Adicionando teste no cartão na azure



Fonte: Autor 2023

O roteiro é uma lista com dois itens principais: a ação que será executada e o resultado esperado, como mostrado na Figura 3.5. O roteiro será usado como referência para se fazer o *Desk Check*.

Figura 3.5 – Item do roteiro de testes



Fonte: Autor 2023

3.6 Desk Check

Desk check ou teste de mesa é um teste feito com dois desenvolvedores. Primeiramente, o desenvolvedor que terminou de desenvolver a história precisa executar a *pipeline* de dev no

Azure DevOps na sua *branch* para que suas alterações no código sejam replicadas no portal que é o ambiente de desenvolvimento. Após isso, ele convida um outro desenvolvedor para testar junto a ele e validar a conclusão de sua história. Com isso, ambos passam pelo roteiro de testes, fazendo a ação do roteiro e comparando com o resultado esperado. Podem ser feitos mais testes além dos que estão no roteiro se os desenvolvedores julgarem necessário para validar completamente a história. Caso algum teste falhe, o desenvolvedor da história terá que corrigir o código, certificar-se que o erro foi corrigido e chamar novamente o mesmo desenvolvedor para finalizar o teste. Após terminado o *desk check*, o desenvolvedor da história deve apontar no cartão de história que o roteiro de testes foi concluído e também apontar o desenvolvedor que colaborou com o *desk check*, assim como nas Figura 3.6.

Figura 3.6 – Exemplo apontamento desk check



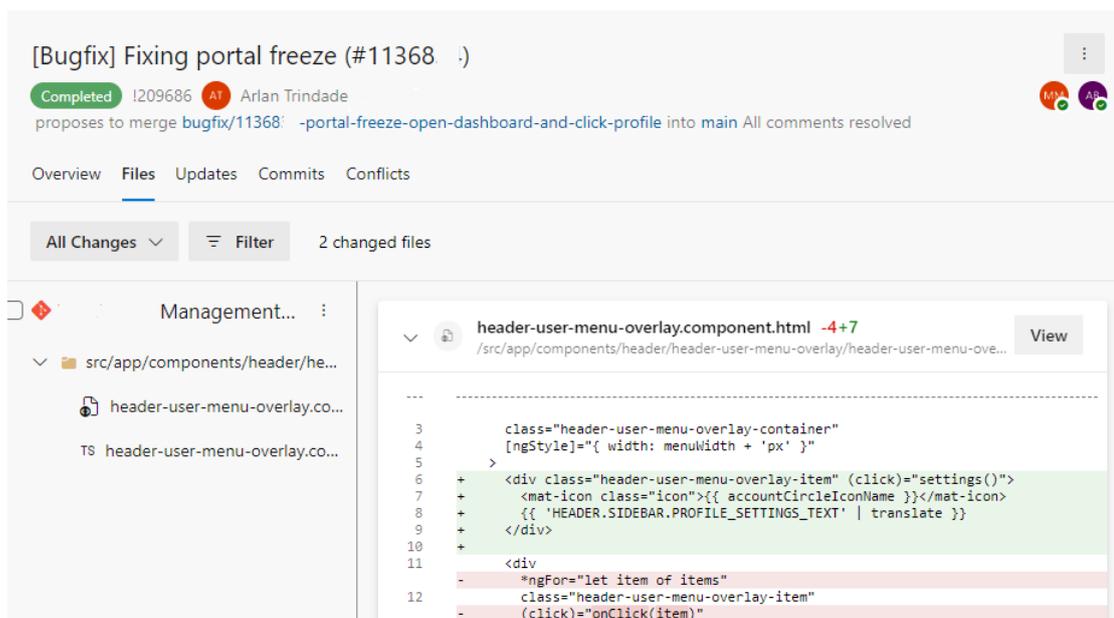
Fonte: Autor 2023

3.7 Code review

O Code review (revisão de código) não faz parte oficialmente da metodologia ágil Scrum, mas é comumente adicionado às metodologias ágeis por ser uma ferramenta essencial para controle de qualidade e aprendizado para a equipe, assim também reduzindo fragilidades no sistema. O code review é feito com o desenvolvedor da história e mais dois membros da equipe. Ao terminar o *desk check* o desenvolvedor abre um PR (*Pull Request*) para que a *branch* de sua história possa ser mesclada na *branch* principal do projeto. Os membros da equipe convidados analisam o código escrito, e tiram dúvidas com o desenvolvedor da história, corrigem erros do código (como *magic numbers* que são números inseridos diretamente no código e não por meio de uma variável constante, nomes ruins de classes, métodos, ou variáveis e também podem ser feitas sugestões gerais de melhorias no código). Participar desse processo, avaliando um código ou tendo o seu código avaliado, é muito enriquecedor para a troca de conhecimento, o que é muito apreciado na empresa estagiada. Um exemplo disso é caso esteja analisando um

PR, e se deparar com uma parte do código que não entendeu muito bem, pode-se perguntar ao desenvolvedor responsável pelo código para explicar do que se trata. A Figura 3.7 mostra um exemplo de um PR na Azure DevOps. No canto superior direito, são visíveis as aprovações do PR e abaixo em verde a parte do código que foi adicionada e em vermelho as partes do código que foram removidas.

Figura 3.7 – Exemplo de um PR



Fonte: Autor 2023

Esses pontos de melhoria no código são feitos por meio de comentários que devem ser marcados como resolvidos para só assim ser possível aprovar o PR. São necessárias duas aprovações para se completar o PR e, só depois disso, que se é permitido fazer o merge da *branch* de desenvolvimento na *branch* principal.

3.8 Teste em QA

Depois da história ter sido aprovada no *code review*, o desenvolvedor deve subir a sua versão da *branch* para a release de QA. Isso é feito por meio de uma *pipeline* na Azure DevOps que após executada com a sua *branch* gera uma release em QA que deve ser aprovada por algum membro que tem autoridade para isso. Muitas vezes essa aprovação é feita pela representante da empresa cliente. Depois da release estar rodando no ambiente de QA, o desenvolvedor da história entra em uma reunião com a representante do cliente e testa todos os pontos da história novamente.

A disciplina de Qualidade de software é uma grande aliada para um entendimento mais profundo desse processo. Ela apresenta ferramentas para validações de qualidade, como normas e modelos de referência, que traçam um paralelo com alguns detalhes desse processo, também dá um entendimento de que a qualidade pode ser relativa, pois ela precisa atender às expectativas do cliente. Nesse ponto, a representante pode pedir alterações que ela achar conveniente para o cliente, como mudar uma palavra ou colocar alguma dica (tool tip) de explicação da nova funcionalidade. Depois da história ser aprovada em QA, o desenvolvedor pode completar a PR, e mesclar sua *branch* de desenvolvimento com a *branch main/master*, partindo assim para a última etapa do desenvolvimento da história.

3.9 Publicação em produção

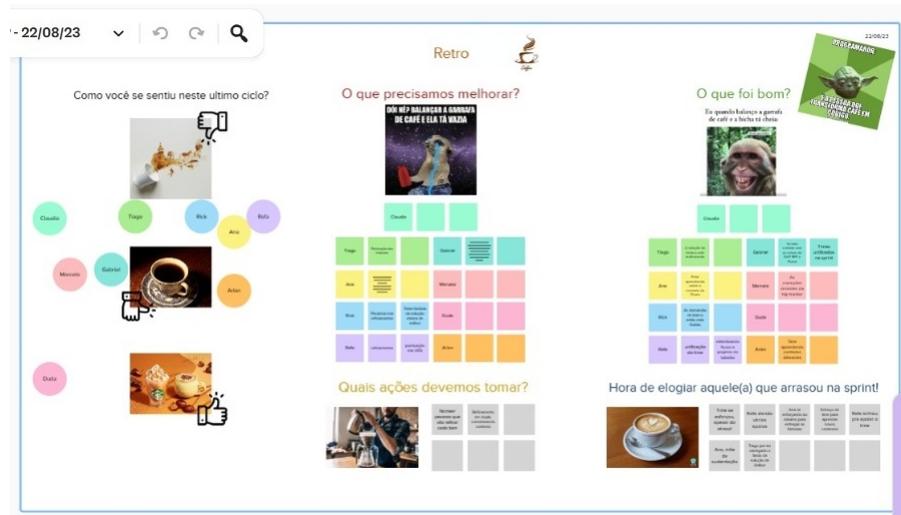
Depois da história ter sido aprovada em QA, ela pode ir para a fase final do desenvolvimento, que é a publicação em produção. Nessa etapa são executadas as pipelines de produção na *branch* principal (*main/master*) e é gerada uma release de produção, que novamente precisa ser aprovada por membros com nível para isso e testada juntamente com a representante da empresa cliente. Nessa etapa também é criado um “plano de *rollback*”, no qual são levantadas as releases de produção anteriores para que, caso ocorra algum problema ou algum *bug* em produção causado pela nova subida, seja possível voltar na versão anterior à subida dessa nova *release*.

3.10 Retro

A Retro ou retrospectiva se trata de uma reunião que acontece no último dia da *sprint*, feita com toda a equipe. Ela tem duração de 1 a 2 horas, no máximo, e possui um tema específico com um tom mais descontraído. O principal objetivo da retro é levantar pontos de melhoria da *sprint*, e é composta por um quadro como o da Figura 3.8 que contém vários elementos.

Os elementos desse quadro são os seguintes: o primeiro deles se trata de uma barra de humor que vai de feliz, normal a triste. Os membros da equipe posicionam um cartão com seu nome indicando o seu humor nessa barra referente ao último ciclo da *sprint*, e a partir disso cada um dos membros da equipe justifica seu humor, falando o que aconteceu para estar se sentindo assim. Outro elemento desse quadro é uma área composta por 3 divisões divididas em “O que podemos melhorar?”, “O que foi bom?” e “Quais ações devemos tomar?”. Na divisão

Figura 3.8 – Exemplo de um quadro da Retro



Fonte: Autor 2023

“O que podemos melhorar?”, os desenvolvedores apontam o que foi ruim na *sprint* do ponto de vista individual dele. Algo como, por exemplo: “Tivemos alguns impedimentos não previstos na história x”. Já na divisão “O que foi bom?” são apontadas as coisas que foram bem na *sprint* como, por exemplo, “Tivemos muitas entregas”. Por fim, e na última “Quais ações devemos tomar”, são apontadas ações do que se pode fazer para que não aconteça mais os problemas apontados no “O que podemos melhorar?”. O último elemento do quadro da retro é a parte de reconhecimento, onde os membros da equipe podem reconhecer colegas que tiveram uma boa atuação na *sprint*, seja ela apoiando outros membros, ou se destacando em alguma atividade realizada.

3.11 Sprint Review

A *sprint review* é uma reunião que acontece ao final da *sprint* feita com toda a equipe, com duração máxima de uma hora. O objetivo dessa reunião é passar por todas as atividades desenvolvidas na *sprint* para que todos os membros tenham conhecimento do que foi desenvolvido, e também algum ponto do desenvolvimento que faça sentido ser compartilhado. Por exemplo, a passagem de conhecimento sobre alguma biblioteca que foi utilizada e a equipe ainda não tinha conhecimento.

Também é feita uma *sprint review* apresentando as histórias feitas durante a *sprint* para o cliente. Essa reunião é menos técnica e, desenvolvedores e estagiários não participam dessa reunião.

Por mais que exista conhecimento teórico sobre como funcionam metodologias ágeis e sprints, na prática, obviamente, ainda são encontradas dificuldades. Acostumar-se com a rotina de reuniões e processos é um grande desafio, saber quais decisões tomar e o que poderia ser dito em reuniões é algo complexo.

Por exemplo, em situações práticas, enfrentar decisões como determinar se deveria pular ou refazer algum processo da sprint devido a contratempos inesperados. Além disso, em reuniões específicas, surgirem questionamentos como: 'Posso pedir ajuda nesta reunião?' ou 'Onde posso expressar preocupações, ou reclamações sobre determinado aspecto do projeto?' Esses cenários demonstram como a aplicação prática das metodologias ágeis envolve uma constante tomada de decisões e comunicação eficaz. Infeliz, ou felizmente, esse desafio só é superado com a prática.

4 EXEMPLOS DE ATIVIDADES REALIZADAS

Neste capítulo são abordados exemplos de atividades realizadas ao longo do período de estágio mostrando em resumo passos de como elas foram desenvolvidas e as dificuldades encontradas.

4.1 Validação de e-mail

No contexto do projeto, um e-mail é considerado como e-mail inválido se ele não começar com letra ou se contiver acentos e caracteres especiais com exceção do '@'. O primeiro passo do desenvolvimento da história foi a escrita do texto que será exibido caso o e-mail seja inválido. Esse texto foi adicionado a um arquivo *json* onde todos os textos dos portais são colocados. O texto salvo no arquivo *json* contém um nome e um identificador, por exemplo: INVALID_EMAIL: “O e-mail não pode conter acentos ou caracteres especiais”. Após isso, analisando e percebendo que essa alteração impactava muitas telas, usando conhecimentos aprendidos na disciplina de Programação orientada a objetos, para evitar a replicação desnecessária de código, foi criada uma classe ‘emailHelper’ que continha as funções de validação do e-mail. Essa classe é usada em todos os componentes que precisam validar os e-mails. Por fim com o uso da classe ‘emailHelper’, que valida se um e-mail é válido ou não, a partir disso é ativado e desativado o botão de salvar, no caso específico do e-mail ser inválido e é exibido também a mensagem abaixo do campo de input, além de mudar a classe CSS do input para que se destaque em vermelho, para isso foi utilizando a diretiva ngClass do Angular, onde é possível aplicar condicionalmente classes CSS ao componente. Abaixo nas Figuras 4.1 e 4.2 é mostrado como as mensagens de e-mail inválido são exibidas no portal.

Figura 4.1 – Mensagem e-mail inválido

Adicionar integrantes

Nome

E-mail @ .com Membro Ponto Focal Gestor

O e-mail não pode conter acentos ou caracteres especiais.

Fonte: Autor 2023

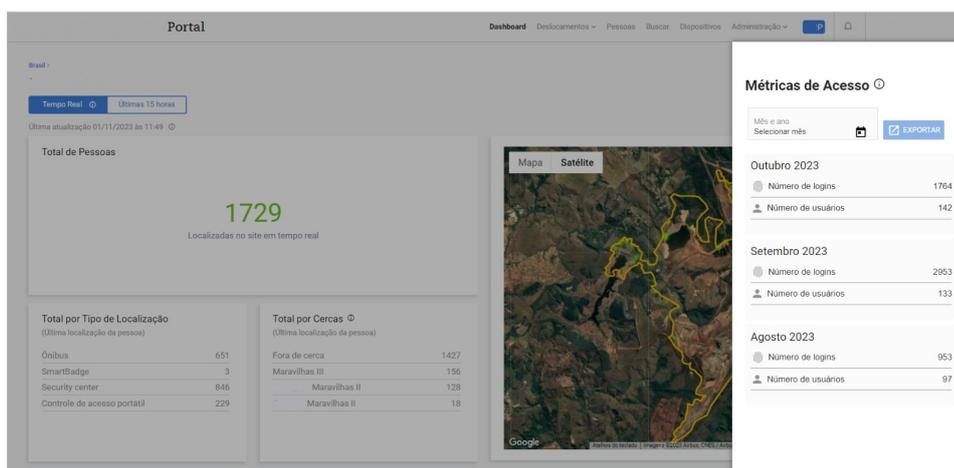
Figura 4.2 – Mensagem Primeiro caractere do e-mail inválido

Fonte: Autor 2023

4.2 Criação da tela de métricas

Foi desenvolvida de uma tela lateral que sobrepõe todas as outras telas e exibe as métricas mensais de acesso dos últimos 3 meses. Essa tela possui também um filtro para um mês específico, como mostrado na Figura 4.3 no canto direito da tela. Primeiramente foi criado o

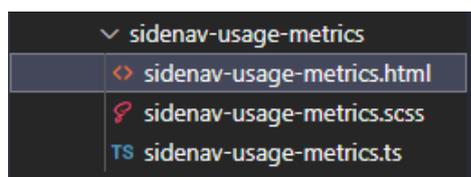
Figura 4.3 – Tela de métricas



Fonte: Autor 2023

componente *angular* que representa essa barra lateral, composto de um arquivo HTML, um arquivo CSS e um arquivo *typeScript* como mostrado na Figura 4.4. A partir disso, foi criada

Figura 4.4 – Exemplo componente em Angular



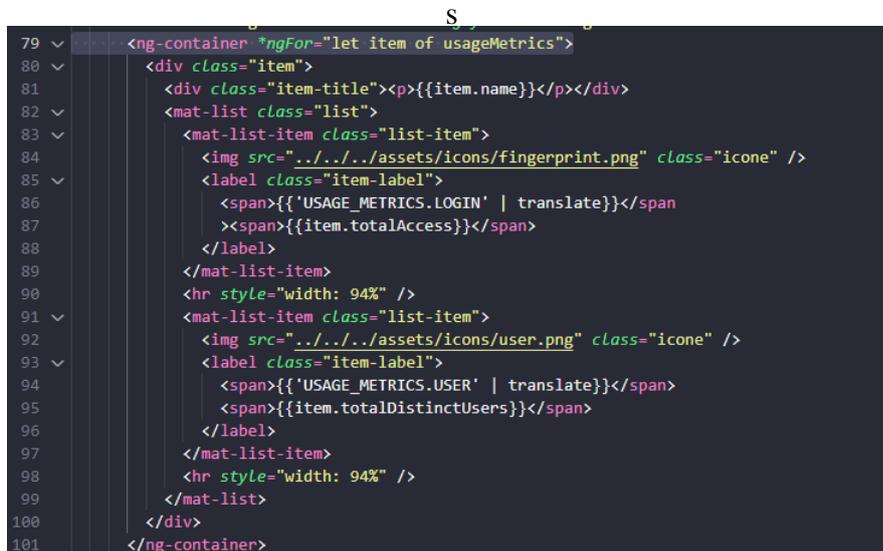
Fonte: Autor 2023

uma estrutura básica para a exibição dos campos na tela.

A listagem do mês se trata de uma lista expansível implementada com a diretiva ‘ngFor’ do Angular, que permite iterar dinamicamente sobre um conjunto de elementos e renderizá-los

no template com base em uma fonte de dados. A Figura 4.5, mostra o uso do ‘ngFor’ e abaixo dela o acesso dos atributos dos objetos da lista para popular a tela. A fonte de dados é retornada

Figura 4.5 – Exemplo código em Angular



```

79 <ng-container *ngFor="let item of usageMetrics">
80   <div class="item">
81     <div class="item-title"><p>{{item.name}}</p></div>
82     <mat-list class="list">
83       <mat-list-item class="list-item">
84         
85         <label class="item-label">
86           <span>{{'USAGE_METRICS.LOGIN' | translate}}</span>
87           <span>{{item.totalAccess}}</span>
88         </label>
89       </mat-list-item>
90       <hr style="width: 94%" />
91       <mat-list-item class="list-item">
92         
93         <label class="item-label">
94           <span>{{'USAGE_METRICS.USER' | translate}}</span>
95           <span>{{item.totalDistinctUsers}}</span>
96         </label>
97       </mat-list-item>
98       <hr style="width: 94%" />
99     </mat-list>
100   </div>
101 </ng-container>

```

Fonte: Autor 2023

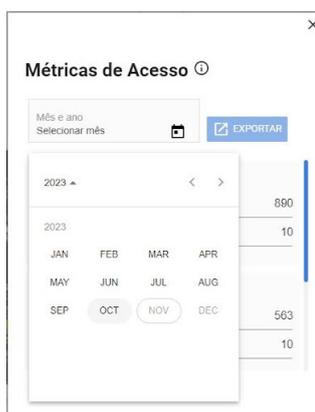
pelo BFF em forma de uma lista. A camada de service do *front end* faz uma requisição para a camada de *repository* referente a essa tela. Como o BFF já estava implementado, esses dados para popular essa tela já vinham da rota que era chamada. Essa rota tinha um parâmetro opcional que era o número do mês, que se fosse passado um valor vazio retornava os últimos 3 meses e se recebesse um valor de 1 a 12 retornava a lista contendo apenas o mês específico. Com isso, foi preciso criar um *datepicker* como o da Figura 4.6. O *framework angular* contém um *datepicker*, que retorna a data selecionada, porém foi preciso adaptá-lo para que fosse escolhido apenas o mês e o ano. Outra limitação para esse *datepicker* era que a data máxima dele seria o mês anterior, pois os dados do mês atual ainda estavam sendo coletados.

4.3 Destaque da cerca selecionada

A necessidade dessa história veio por meio do cliente que dada a quantidade de cercas, ele não conseguia distinguir qual cerca estava sendo selecionada, como mostrado na Figura 4.7. Dependendo da quantidade de cercas a visualização fica comprometida, então, a solução foi destacar a área da cerca segundo a cor da cerca, com uma menor opacidade.

Os mapas e as cercas são gerados por uma *api* do *google maps*. Com base na documentação dessa *api* era possível mudar a cor do preenchimento do polígono usando as propriedades *fillColor* e *fillOpacity*.

Figura 4.6 – Datpicker



Fonte: Autor 2023

Figura 4.7 – Mapa com cercas



Fonte: Autor 2023

Como todas as cercas vinham com uma *fillColor* padrão cinza, foi necessário mudar a cor de cada uma de acordo com a cor da “*strokeColor*”, que é a cor da linha que desenha a cerca que era recebido como parâmetro ‘*color*’. Foi necessário somente atribuir esse valor da ‘*color*’ para a ‘*fillColor*’. Outro desafio era que todas as cercas são salvas em uma lista. Então, houve uma dificuldade para identificar qual cerca da lista era equivalente à cerca selecionada no mapa. Foi então criado um dicionário que tinha o nome da cerca e o índice dela que foi populado como na Figura 4.8 e, como já existia uma função que abria e fechava o menu que descrevia a cerca, bastou adicionar a chamada da função ‘*highlightPolygon*’ (Figura 4.9). A função, recebia o nome da cerca e um booleano, se necessário destacar a cerca, o booleano vinha como verdadeiro e por meio do nome da cerca destacava o polígono referente a ela. A função que cuida desse comportamento está na Figura 4.10.

Figura 4.8 – Dicionário

```

for (let i = 0; i < this.geofences.length; i++) {
  this.geofences[i].setValues({
    id: geofences[i].id,
    categoryId: geofences[i].categoryId
  });
  this.fencesDictionary[geofences[i].name] = i;
}

```

Fonte: Autor 2023

Figura 4.9 – Chamada do método

```

marker.addListener('click', () => {
  infoWindow.open(this.map, marker);
  this.highlightPolygon(geofence.name, true);
});
infoWindow.addListener('closeclick', () =>
  this.highlightPolygon(geofence.name, false)
);

```

Fonte: Autor 2023

Figura 4.10 – Método highlightPolygon

```

private highlightPolygon(fenceName: string, highlight: boolean): void {
  const index = this.fencesDictionary[fenceName];

  if (highlight) {
    this.geofences[index].setOptions({ fillOpacity: 0.35 });
  } else {
    this.geofences[index].setOptions({ fillOpacity: 0 });
  }
}

```

Fonte: Autor 2023

Ao final, o resultado de quando se seleciona uma cerca pode ser visto na Figura 4.11. Quando a cerca é selecionada, são mostradas as informações dela e toda a sua área é destacada com a sua cor.

Figura 4.11 – Mapa com cerca destacada



Fonte: Autor 2023

5 CONCLUSÃO

O presente relatório buscou proporcionar uma visão geral do processo ágil adaptado da DTI Digital, com o intuito de auxiliar o leitor a entender melhor essa prática específica. Ao abordar cada etapa, desde o planejamento até a revisão, pretendia-se oferecer uma narrativa esclarecedora sobre como a empresa implementa e personaliza as práticas ágeis para seu contexto. Este objetivo foi alcançado ao apresentar todos os passos de forma a destacar desafios, decisões cruciais e a importância da colaboração eficiente. Acredita-se que este estudo contribuirá significativamente para o entendimento prático do funcionamento de sprints em ambientes empresariais, proporcionando *insights* valiosos para o leitor que busca aprimorar suas habilidades no contexto ágil.

O estágio foi de muita importância. Trabalhar em uma empresa de grande porte como a DTI proporcionou uma perspectiva única e valiosa das operações em larga escala, enquanto a adoção de metodologias ágeis me permitiu adquirir habilidades essenciais em gestão de projetos e desenvolvimento de software. A adoção de metodologias ágeis proporciona benefícios imediatos e duradouros, promovendo uma produção de projetos mais organizada e eficiente, o que é fundamental para se destacar no mercado de trabalho. O destaque vai para a ênfase na aprendizagem contínua, pois a empresa encoraja constantemente a aquisição de novos conhecimentos e a busca por soluções inovadoras. Além disso, a colaboração entre os colegas e equipes foi uma parte fundamental da cultura da empresa, promovendo um ambiente de trabalho onde a troca de ideias e a resolução de desafios ocorreram de forma eficaz e produtiva. Em resumo, o estágio em uma empresa de grande renome, com a aplicação de metodologias ágeis, enfatizando a aprendizagem contínua e promovendo uma cultura de colaboração, contribuiu significativamente para a minha formação profissional e me preparou para os desafios que enfrentarei no futuro. Estou profundamente grato por essa experiência enriquecedora e pelas oportunidades que ela proporcionou.

Para aprimorar o curso de Sistemas de Informação, sugere-se a inclusão de disciplinas adicionais focadas em tecnologias específicas, reduzindo proporcionalmente a ênfase em teorias computacionais. Essa abordagem prática proporcionaria aos estudantes uma experiência mais alinhada com as demandas do mercado, preparando-os de maneira mais eficaz para os desafios da indústria de tecnologia.

REFERÊNCIAS

ANGULAR. **ANGULAR**. 2023. Disponível em: <<https://angular.io/>>.

DTI. **Site DTI Digital**. 2023. Disponível em: <<https://www.dtidigital.com.br/>>.

MICROSOFT. **Azure**. 2023. Disponível em: <<https://azure.microsoft.com/pt-br/>>.

MICROSOFT. **DotNet**. 2023. Disponível em: <<https://dotnet.microsoft.com/pt-br/>>.

MICROSOFT. **O que é o Azure DevOps?** 2023. Disponível em: <<https://learn.microsoft.com/pt-br/azure/devops/user-guide/what-is-azure-devops?view=azure-devops>>.

POSTMAN. **What is Postman??** 2023. Disponível em: <<https://www.postman.com/product/what-is-postman/>>.

SCHWABER, K. **The scrum guide**. 2020. Disponível em: <<https://www.scrum.org/>>.

VALENTE, M. T. **Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade**. 2020. Disponível em: <<https://engsoftmoderna.info/>>.