

Relato de Experiência Sobre o Processo de Entrega Contínua de um Sistema de Controle de Estoque

Márcio J. de Carvalho, Arthur T. Sperandio, Pablo Henrique L. Domiciano, Yago dos S. Macinelli, Frank W. C. de Oliveira, Marcelo Figueiredo Terenciani

¹Campus Paranavaí - Instituto Federal do Paraná (IFPR)
Paranavaí– PR – Brasil

{marciojosedecarvalho, arthur.toyokawa,

pablo.domiciano, yagom.ym}@gmail.com,

{frank.willian, marcelo.terenciani}@ifpr.edu.br

As disciplinas de Projeto Integrador e Construção de Software fazem parte do currículo do curso de Engenharia de Software e têm como objetivo aplicar os conhecimentos adquiridos na criação de *software*.

O Projeto Integrador combina conceitos de diferentes disciplinas, enquanto a Construção de Software se concentra em aplicar esses conhecimentos por meio de metodologias avançadas. Essas disciplinas buscam unir teoria e prática, proporcionando aos estudantes uma base sólida em Engenharia de Software.

Com esse intuito foi proposto aos discentes a integração de ambas disciplinas para o desenvolvimento de um sistema, o qual foi dado continuação de um projeto que teve início no ano de 2022, na disciplina de Projeto Integrador I; definindo a aplicação de técnicas da metodologia *Scrum*[Schwaber and Sutherland 2020], na qual a mesma foi adaptada para abranger melhor a carga horária e limitações de comunicação entre a equipe, sendo a maior delas o *Daily Scrum*, adaptado para ser realizado apenas nos dias de aulas em que ocorreria o desenvolvimento do sistema.

O *Scrum* baseia-se no empirismo e no *lean thinking* - traduzido como ”pensamento enxuto”. Empirismo que parte da premissa que conhecimento é adquirido a partir de experiências e de tomadas de decisões baseando-se no que se observa. *Lean thinking* firmasse na redução dos desperdícios e foca-se no essencial a ser realizado.

No *Scrum* temos o *Scrum Team* que é composto pelo *Product Owner*, responsável pelo valor do trabalho; *Scrum Master*, encarregado da metodologia; e vários *Developers*, que geram o produto. Essa equipe colabora com os *stakeholders*, incluindo clientes e usuários, no desenvolvimento do produto.

O *Scrum* é realizado em *Sprints*, períodos de tempo em que o *Product Backlog* é transformado em tarefas a serem entregues. Esse processo envolve vários eventos, incluindo o *Sprint Planning*, onde o que será feito na *Sprint* é planejado; o *Daily Scrum*, uma reunião diária em que os *Developers* avaliam o progresso e fazem adaptações se necessário; a *Sprint Review*, em que os resultados da *Sprint* são apresentados aos *stakeholders*; e a *Sprint Retrospective*, onde o *Scrum Team* analisa a *Sprint* anterior e identifica melhorias para a próxima.

Com base nessas definições, os papéis foram atribuídos aos membros da equipe. O *Product Owner* foi escolhido devido ao seu amplo conhecimento do sistema desejado,

o *Scrum Master* devido às suas habilidades de gerenciamento, e os *Developers* foram distribuídos de acordo com suas habilidades anteriores para trabalhar em diferentes partes do programa. No decorrer do ano letivo houve a saída de dois membros, o que impactou diretamente no tamanho do incremento entregue, como resultado o *Sprint Backlog* precisou ser adaptado e itens foram passados para *Sprints* posteriores.

As tecnologias estabelecidas para o desenvolvimento do projeto foram a linguagem Java¹ com o *framework* - biblioteca de códigos pré-arquitetados - Spring Boot², ambas aplicadas ao desenvolvimento do *back-end*³ do sistema, já para o *front-end*⁴ foi definida a linguagem Typescript⁵ agregada ao *framework* de interface baseado em componentes: React⁶.

Para o gerenciamento do banco de dados, estabelecemos o modelo relacional e para tal o *SGDB* - Sistema Gerenciador de Banco de Dados - MySQL⁷. Para o *deploy* - implantação do *software* em um ambiente de produção - do sistema foi usado os serviços da Oracle Cloud⁸. Enquanto para todo o *versionamento* - gerenciamento de versões - determinou-se o uso da plataforma Github.

Essas definições ocorreram pelo próprio *Scrum Team*, sendo estabelecidas por terem afinidade com as mesmas, e por possuírem aulas que usam essas mesmas tecnologias, agregando o conhecimento daqueles já são familiarizados e sendo essencial para aqueles que não tiveram contato com as mesmas.

Depois de escolher as tecnologias a serem usadas, a próxima etapa é definir o que o sistema deve incluir e como deve funcionar. Seguindo a metodologia *Scrum*, o *Product Owner* elaborou uma lista completa de recursos e funcionalidades para o *Product Backlog*. Essas decisões foram baseadas nas necessidades do projeto, que envolve o gerenciamento de estoque e caixa de uma loja de roupas.

Para auxiliar no controle e nas métricas de desenvolvimento foi aplicado o uso da ferramenta Jira Software, esta ferramenta possui potentes auxiliares para desenvolvimento baseado no *Scrum*, dispendo de painéis ágeis que dividem projetos em partes de trabalho gerenciáveis, para haver lançamentos mais rápidos, facilitando na automatização de um fluxo de trabalho ágil. Por meio dela foi criado *issues* - problemas ou questões a serem resolvidos no *software*, tarefas elencadas a cada desenvolvedor de acordo com suas especificidades e afinidades para resolver o problema demandado.

Após a finalização dos lançamentos do *Product Backlog* na ferramenta Jira, iniciou-se a codificação do sistema utilizando a IDE - Ambiente de Desenvolvimento Integrado - IntelliJ, ferramenta para desenvolvimentos em Java, e em minoria o editor Visual Studio Code.

Os *Developers*, 4 alunos no total, foram divididos em dois times, baseado nas es-

¹<https://docs.oracle.com/javase/8/docs/>

²<https://spring.io/projects/spring-boot>

³Back-end é parte do *software* que lida com a lógica do servidor

⁴Front-end é a interface do usuário e interações visíveis de um sistema

⁵<https://www.typescriptlang.org>

⁶<https://react.dev>

⁷<https://www.mysql.com>

⁸<https://www.oracle.com/cloud/>

pecificidades de cada membro, um grupo atuaria como *front-end developers* e o outro como *back-end developers*, essa divisão ocorreu para maximizar a progressão do sistema dentro do tempo limite estabelecido nas *Sprints*, que eram definidas em torno de 5 semanas, totalizando um total de 10 horas mínimas trabalhadas.

Então o time de desenvolvimento do *back-end* focou-se na criação de toda a estrutura do banco de dados e suas respectivas entidades, em seguida encetou as funcionalidades de persistência de cada entidade, manipulando os dados presentes no banco de dados.

Em paralelo, o time responsável pelo *front-end* ficou empenhado pela criação das interfaces do sistema e dos formulários, integrando com APIs - Interface de Programação de Aplicativos - para obtenção de dados gerenciados por um serviço específico, como, por exemplo, ViaCEP⁹, API que gerencia dados de CEP e IBGE do Brasil.

Após finalizar uma *issue* através de uma *branch* - recurso fornecido pela ferramenta Git¹⁰ - que se trata de uma ramificação do “ramo” principal onde fica estabelecido o código revisado e pleno; o desenvolvedor abre um *pull request*, ação de solicitar uma revisão que valida as implementações realizadas a serem integradas à *branch* principal. Em caso de incoerências ou necessidades de ajustes no código, o *pull request* é negado e o desenvolvedor retorna à *issue* para realizar os ajustes necessários.

Ao decorrer do desenvolvimento houve conversas entre o *Product Owner* e os *stakeholders*, onde lhes foi apresentado os estados atuais do mesmo e questionado suas opiniões sobre todo o andamento do processo, essas conversas aconteciam por meio de e-mails e mensagens pelo WhatsApp ao decorrer da *Sprint*.

Este desenvolvimento nos ajudou a compreender a maior parte dos conhecimentos adquiridos ao tempo que os colocávamos no código, as metodologias estudadas fizeram mais sentido ao passo que vimos como sem elas a equipe fica um pouco dispersa, tendo definições claras de funções e objetivos todo o fluxo de trabalho caminha corretamente. A junção de ambas as disciplinas se deu com excelência por toda essa experiência que obtemos no decorrer do ano letivo.

Esse desenvolvimento nos ajudou a aplicar nosso conhecimento no código, tornando as metodologias mais compreensíveis. Percebemos que, sem essas metodologias, a equipe fica menos organizada. A combinação das disciplinas foi excelente devido à experiência adquirida durante o ano letivo.

Nesse contexto, fica evidente a importância de desenvolver projetos que tenham aplicações no mundo real, permitindo a identificação de requisitos com base em problemas reais ou solicitações do cliente. Nesse sentido, a metodologia *Scrum* se destaca como um excelente fluxo de trabalho para aprimorar a experiência acadêmica, preparando os alunos para integrarem equipes de desenvolvimento no mundo profissional. Além disso, ao trabalharem com o *Scrum* durante seus estudos, os alunos adquirem uma experiência que os ajudará a enfrentar desafios profissionais e a desenvolver suas habilidades ao longo do ciclo de vida do *software*.

⁹<https://viacep.com.br>

¹⁰<https://git-scm.com>

References

Schwaber, K. and Sutherland, J. (2020). *O Guia do Scrum*. Scrum.org.