

Relato de Experiência sobre o Uso de Scrum no Desenvolvimento de um Sistema de Gestão para Academias

Gustavo G. Berti¹, Laryssa L. M. R. Bino¹, Luan G. M. Marques¹,
Thalles S. M. Barros¹, Vinicius de O. Rebussi¹,
Frank W. C. de Oliveira¹, Marcelo F. Terenciani¹

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

{20230006414, 20230006460, 20230006174, 20230006423,
20230006399}@estudantes.ifpr.edu.br
{frank.willian, marcelo.terenciani}@ifpr.edu.br

Inserido no contexto acadêmico do Instituto Federal do Paraná (IFPR) – Campus Paranavaí, este trabalho foi desenvolvido como parte de uma proposta interdisciplinar, integrando as disciplinas de Construção de Software e Projeto Integrador. A iniciativa resultou no desenvolvimento do projeto *Stilofit*, conduzido por uma equipe de alunos do 3º ano do curso de Engenharia de Software, com orientação docente durante o ano letivo de 2025. A iniciativa tem como proposta a implementação de um sistema *web* como parte dos requisitos para conclusão das disciplinas, possibilitando a aplicação de conhecimentos relacionados a metodologias ágeis [Beck et al. 2001], levantamento e análise de requisitos, arquitetura de sistemas, versionamento de código, integração entre o ambiente de interação do usuário (*front-end*) e a estrutura funcional (*back-end*).

A definição do escopo do sistema surgiu da necessidade apresentada pela Academia Corpo Stilo de otimizar seus custos operacionais, desenvolvendo uma solução própria, em substituição aos serviços terceirizados. Isso visa superar dificuldades comuns enfrentadas pelos gestores, como administração de contratos, turmas, controle financeiro, comunicação com clientes e integração de processos, tudo por meio de uma ferramenta centralizada.

Diante desse contexto, o principal objetivo do projeto consiste na criação de um sistema *web* integrado para gestão de academias, capaz de automatizar rotinas administrativas, operacionais e comerciais. O desenvolvimento foi sendo conduzido com base nos requisitos levantados junto à academia Corpo Stilo, mas sua concepção foi pensada de forma genérica, possibilitando a aplicação em outras academias que adotem fluxos de trabalho semelhantes. Além disso, buscou-se proporcionar aos estudantes uma experiência prática de aplicação das metodologias ágeis, do uso de tecnologias modernas e do desenvolvimento de competências como trabalho em equipe, organização e resolução de problemas.

A metodologia escolhida foi o Scrum [Schwaber and Sutherland 2020], devidamente adaptada ao contexto acadêmico. A equipe foi organizada conforme os papéis previstos no *framework*: um estudante assumiu o papel de *Product Owner* (PO), responsável por intermediar a comunicação com o cliente e priorizar o *backlog*¹; outro o de *Scrum Master* (SM), encarregado de facilitar as cerimônias, garantir a aplicação da

¹*Backlog* é uma lista ordenada de requisitos, funcionalidades, melhorias e correções que representam tudo o que é necessário para o desenvolvimento de um produto. No Scrum, existe o *Product Backlog*, que

metodologia e apoiar a equipe no processo de melhoria contínua; e, três desempenharam o papel de desenvolvedores, dedicando-se à implementação das funcionalidades planejadas. Entre as práticas adotadas estiveram a divisão de tarefas em *Sprints*², a realização de *Sprint Plannings*, e, *Sprint Reviews* e *Retrospectives* ao final de cada ciclo, como forma de avaliar os resultados obtidos e promover melhorias contínuas no processo de desenvolvimento, além de uma interação constante com o cliente. Essa abordagem proporcionou ao grupo maior clareza sobre responsabilidades, prazos e prioridades, além de promover organização e transparência no processo de trabalho.

Para o desenvolvimento do sistema, foi utilizado React com Vite para o desenvolvimento do *front-end*, Java com *Spring Boot* para o *back-end*, Figma para o projeto das interfaces de usuário e Visual Studio Code como ambiente de desenvolvimento integrado (IDE). Essas ferramentas foram escolhidas devido à ampla adoção no mercado, facilidade de aprendizado e compatibilidade com os objetivos acadêmicos do projeto.

A gestão do código e das tarefas foi realizada por meio de uma organização no GitHub³, onde foram criados repositórios separados para *front-end* e *back-end*, com permissões controladas para garantir segurança e colaboração eficiente. Cada história de usuário tinha uma ramificação (*branch*) própria e as alterações eram submetidas via *Pull Request* (PR), aprovadas após revisão de ao menos um outro membro da equipe. O quadro Kanban [Anderson 2015] do GitHub Projects foi estruturado em colunas como “*Product Backlog*”, “*Sprint Backlog*”, “*In Progress*”, “*In Review*” e “*Done*”, permitindo o acompanhamento visual do progresso das *Sprints* e garantindo maior transparência entre os integrantes.

Na primeira *Sprint* a equipe ficou responsável por desenvolver uma página institucional do projeto, com o objetivo de se organizar, familiarizar-se com as ferramentas, disponibilizar tempo para que o PO levantasse as histórias de usuário (HUs) iniciais e produzisse protótipos para o próximo ciclo. Para a segunda e terceira *Sprints*, iniciou-se a implementação das interfaces de usuário principais do sistema no *front-end*, com áreas destinadas de gestão de clientes, contratos e turmas, utilizando de dados persistidos em armazenamento local. A quarta *Sprint* teve um foco maior na integração da funcionalidade do cliente com o *back-end*, além da criação nova tela de funcionário e a resolução de pendências das outras *Sprints*.

O levantamento de requisitos foi realizado por meio de reuniões com o cliente, prototipação das interfaces de usuário e com uma breve experiência do funcionamento do negócio. No começo do projeto, um dos desafios enfrentados foi a participação do cliente, inicialmente limitada. Isso impactou o detalhamento dos requisitos, o que dificultava a obtenção de *feedback* detalhado. Esse cenário demandou explicações adicionais por parte do PO sobre o funcionamento do processo ágil, ressaltando a importância de validações constantes. No decorrer do projeto a participação do cliente se tornou mais efetiva, o que contribuiu para maior clareza e alinhamento das funcionalidades.

contém todas as demandas do produto, e o *Sprint Backlog*, que reúne os itens selecionados para serem desenvolvidos durante uma *Sprint*.

²Sprint é um ciclo de tempo fixo no *framework* Scrum (geralmente entre 1 e 4 semanas), durante o qual uma equipe trabalha para desenvolver um conjunto definido de funcionalidades ou melhorias. Cada *Sprint* inclui planejamento, execução, revisão e retrospectiva.

³<https://github.com/CS-PI-2025-Thread>

Além disso, a elaboração dos primeiros critérios de aceite foi um desafio, por isso exigiram ajustes até atingir o nível de clareza necessário. Outro desafio foi a curva de aprendizado com React e a integração entre *front-end* e *back-end*, superados com estudos complementares, uso de protótipos elaborados no Figma e apoio mútuo entre os integrantes. Ademais, outro desafio encontrado foi a comunicação entre os integrantes da equipe fora do ambiente de sala de aula, exigindo recorrentes conversas para alinhamento.

Com o objetivo de facilitar o acompanhamento contínuo por parte do cliente, a aplicação foi implantada na plataforma Vercel⁴, que oferece hospedagem gratuita, integração com repositórios Git e *deploy* automatizado. Essa abordagem permitiu que o cliente acompanhasse em tempo real a evolução do sistema ao longo das entregas parciais, promovendo maior alinhamento com as suas expectativas e contribuindo para decisões mais ágeis durante o desenvolvimento.

Os resultados obtidos até o momento incluem a entrega de protótipos validados, a implementação inicial das telas principais e a definição de arquitetura consistente para evolução do sistema. A avaliação inicial, feita por professores, apontou pontos positivos quanto à organização do projeto e ao uso de metodologias ágeis, ao mesmo tempo que sugeriu melhorias na componentização do *front-end* e na clareza das regras de negócio. Esse *feedback* foi fundamental para ajustes nas *Sprints*, incluindo a elaboração de protótipos mais detalhados e critérios de aceitação mais específicos para *back-end*.

Do ponto de vista de aprendizado, a experiência permitiu aos estudantes aprofundar conhecimentos em React, Spring Boot e integração de sistemas, além de desenvolver competências interpessoais como colaboração, comunicação e gestão de tempo. A prática de metodologias ágeis também reforçou a importância do planejamento incremental e da adaptabilidade. O exercício das funções de *Product Owner*, *Scrum Master* e desenvolvedor ampliou a visão sobre a importância da priorização, da transparência com o cliente e da flexibilidade diante de imprevistos.

Como trabalhos futuros, prevê-se a expansão do sistema com funcionalidades avançadas, como bloqueio automático de inadimplentes e integração com o Gympass. Além disso, espera-se aprimorar a documentação, os testes e a escalabilidade da aplicação, garantindo sua evolução contínua.

Referências

- Anderson, D. J. (2015). *Kanban: sucesso evolutivo com o método ágil*. Casa do Código, São Paulo.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). Manifesto para Desenvolvimento Ágil de Software. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 19 set. 2025.
- Schwaber, K. and Sutherland, J. V. (2020). *O Guia do Scrum: O guia definitivo para o Scrum: As regras do jogo*. Scrum.org.

⁴A aplicação está disponível em: <https://stilofit-front-end.vercel.app>