Introdução

Desenvolver um sistema do zero parece assustador no começo. São muitas ideias, tarefas, ajustes e decisões acontecendo ao mesmo tempo. Foi justamente por isso que, desde o início do nosso projeto, entendemos que não dava pra seguir sem organização e uma boa estratégia de desenvolvimento. Foi aí que entrou o Ciclo de Vida de Desenvolvimento de Software, que nada mais é do que um jeito de planejar, construir, testar e entregar um sistema de forma estruturada.

Pra facilitar essa jornada e tornar tudo mais leve e eficiente, escolhemos trabalhar com metodologias ágeis, mais especificamente o Scrum. Essa escolha fez toda a diferença, principalmente porque nosso time era formado por estudantes com pouco tempo livre, mas muita vontade de aprender e entregar algo funcional. Ao invés de tentar fazer tudo de uma vez, dividimos o trabalho em etapas menores e focamos em entregar valor aos poucos, sem sobrecarregar ninguém.

O ciclo de vida do software é o conjunto de fases que um sistema percorre desde sua concepção até seu fim, passando por etapas como levantamento de requisitos, modelagem, desenvolvimento, testes e manutenção (SOMMERVILLE, 2019). A forma como essas etapas são organizadas pode variar bastante dependendo do modelo escolhido. No nosso caso, optamos por um modelo ágil, que foca em flexibilidade, entregas frequentes e colaboração constante com a equipe.

O Scrum, segundo Schwaber e Sutherland (2020), é um dos frameworks ágeis mais usados no mundo todo, justamente por sua simplicidade e eficácia. Ele divide o projeto em ciclos curtos chamados sprints, geralmente de uma a quatro semanas, onde ao final de cada um existe uma entrega concreta e funcional do sistema. Isso permite receber feedback rápido e corrigir problemas antes que eles se tornem grandes demais. E foi exatamente isso que buscamos aplicar no nosso projeto.Desde o começo, organizamos o desenvolvimento em sprints semanais, onde cada integrante tinha tarefas bem definidas. No início de cada sprint, fazíamos reuniões de planejamento para decidir o que seria entregue naquela etapa, e ao final fazíamos uma retrospectiva para avaliar os acertos e pontos a melhorar. Esse ciclo contínuo nos ajudou a manter o foco, evoluir o projeto aos poucos e, principalmente, aprender muito com os erros e acertos.

As ferramentas que facilitaram a nossa vida

Durante o projeto, usamos várias ferramentas que foram essenciais para manter a organização, o controle e a colaboração entre o grupo:

Figma: Antes de sair programando, prototipamos as telas no Figma. Isso nos deu uma visão clara de como o sistema deveria funcionar e nos ajudou a alinhar as ideias entre todos os membros, principalmente nas partes visuais e de navegação.

Kanban: Utilizamos um quadro Kanban (digital) com colunas como “A Fazer”, “Fazendo” e “Feito”. Assim, conseguimos visualizar o andamento das tarefas, distribuir o trabalho de forma equilibrada e evitar que alguém ficasse sobrecarregado.

GitHub: Foi nosso repositório oficial de código. Usamos branches, pull requests e commits para manter tudo versionado e colaborativo. Isso garantiu segurança no desenvolvimento e um histórico completo das alterações.

Scrum: Mais do que uma metodologia, foi a base do nosso trabalho. Tivemos dailies curtas, reuniões de planejamento e retrospectiva, e um Product Backlog com todas as funcionalidades previstas. A estrutura enxuta do Scrum nos manteve produtivos, mesmo com a correria do dia a dia.

A escolha por trabalhar com Scrum e essas ferramentas foi estratégica: com tempo limitado e um grupo pequeno, precisávamos de uma forma de desenvolvimento que fosse prática, adaptável e eficiente. Os métodos ágeis oferecem justamente isso — entregas rápidas, possibilidade de mudança ao longo do caminho e foco no trabalho em equipe (PRESSMAN, 2016).

Essa organização nos permitiu:

* Trabalhar bem, mesmo à distância;
* Manter a equipe engajada e bem distribuída;
* Corrigir problemas rapidamente, sem retrabalho desnecessário;
* E principalmente: ver resultados concretos a cada semana.

Seguir um ciclo de vida ágil com Scrum foi uma das melhores decisões do projeto. Isso nos permitiu desenvolver com mais tranquilidade, manter a equipe alinhada e, ao mesmo tempo, ter entregas constantes. Essa experiência prática mostrou que, mesmo em projetos pequenos, aplicar boas práticas de engenharia de software e organização faz toda a diferença. Com certeza, esse aprendizado vai ser útil não só em trabalhos futuros na faculdade, mas também na vida profissional.

Referências

* PRESSMAN, Roger S. *Engenharia de Software: uma abordagem profissional*. 7. ed. Porto Alegre: AMGH, 2016.
* SOMMERVILLE, Ian. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019.
* SCHWABER, Ken; SUTHERLAND, Jeff. *The Scrum Guide – O Guia Definitivo do Scrum*. 2020.

Se quiser, posso te ajudar também com os próximos tópicos do trabalho, como a documentação técnica do projeto, a descrição das sprints, ou até mesmo a conclusão final. Deseja continuar com isso?