**5. ENGENHARIA DE SOFTWARE II**

Nessa etapa do projeto, focamos em aplicar na prática os conhecimentos de Engenharia de Software II, que é basicamente onde a teoria encontra o desenvolvimento real. Aqui, a gente trabalhou com modelagem mais detalhada dos casos de uso, planejamento dos testes e definição de comportamentos esperados para cada funcionalidade do sistema. Essa fase foi muito importante para deixar o Clickdesk mais confiável e preparado para uso de verdade, pensando tanto na parte técnica quanto na experiência do usuário.

Começamos com a elaboração dos modelos de casos de uso, que nada mais são do que representações gráficas das funcionalidades do sistema e da forma como os usuários interagem com ele. Fizemos um diagrama geral, que mostra uma visão macro do sistema, e depois quebramos em diagramas individuais por cenário, pra conseguir detalhar melhor cada funcionalidade — como abrir chamado, responder chamado, consultar FAQ, encaminhar para técnico, encerrar chamado etc.

Para cada caso de uso, escrevemos uma descrição sucinta e objetiva, mas cobrindo tudo que precisava: fluxo principal (o caminho padrão que a ação segue), fluxos alternativos (variações que podem acontecer) e fluxos de exceção (erros ou situações inesperadas, como uma falha na IA ou falta de conexão). Além disso, colocamos as pré-condições (por exemplo, o usuário precisa estar logado pra abrir um chamado) e as pós-condições (como o chamado ficar registrado no sistema com número e status atualizado).

Também usamos os relacionamentos da UML para deixar os diagramas mais organizados e mostrar conexões entre os casos:

«include» foi usado para representar ações que sempre fazem parte de outra, como "Atualizar Status" que é incluído em "Encerrar Chamado".

«extend» foi útil em ações opcionais ou contextuais, como "Autoatendimento com IA", que só acontece se o sistema identificar que pode sugerir uma solução automática.

Generalização entrou em situações em que tínhamos ações parecidas, como "Responder Chamado por Técnico" e "Responder Chamado por IA", que herdam comportamentos do caso mais genérico "Responder Chamado".

Depois da modelagem, partimos pra planilha de testes, onde documentamos os casos de teste de forma organizada. Cada linha da planilha mostra um cenário com ID do teste, descrição da funcionalidade, entradas necessárias, passos a serem seguidos, resultado esperado e o resultado obtido. Isso serviu pra validar tudo que foi planejado: ver se o sistema realmente fazia o que a gente pensava e se os fluxos estavam funcionando bem.

Esse processo ajudou muito a identificar erros antes mesmo de o sistema ser usado “pra valer”. Vários ajustes que fizemos só foram percebidos por conta dos testes, então foi uma etapa essencial. Mais do que só encontrar falhas, os testes ajudaram a melhorar a experiência do usuário final e garantir que o sistema fosse confiável e estável.

No geral, essa parte de Engenharia de Software II foi bem prática e desafiadora, mas também muito útil para enxergar o sistema de uma forma mais madura. É diferente só de pensar nas funções — aqui a gente teve que planejar, prever exceções e validar de verdade. Isso tudo deixou o Clickdesk muito mais preparado para ser uma solução profissional.

Referências

1. PRESSMAN, Roger S. *Engenharia de Software: uma abordagem profissional*. 7. ed. Porto Alegre: AMGH, 2016.
2. SOMMERVILLE, Ian. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019.
3. LARMAN, Craig. *Utilizando UML e Padrões: uma introdução à análise e ao projeto orientado a objetos*. 3. ed. Porto Alegre: Bookman, 2007.
4. BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. *UML: Guia do Usuário*. Rio de Janeiro: Elsevier, 2006.
5. ALENCAR, Felipe; BARBOSA, Bruno. *Requisitos de Software: uma abordagem prática*. São Paulo: Novatec, 2018.