



A Visionnaire validou o código existente e estruturou a evolução modular para concluir o desenvolvimento de um CRM

Em meados de 2020, a Britânia (também controladora da marca Philco no Brasil) estava desenvolvendo com outro fornecedor um novo CRM (do inglês *Customer Relationship Management* - Gestão de Relacionamento com o Cliente) para centralizar o relacionamento com clientes e *leads* (contatos, histórico de interações, oportunidades, propostas, pedidos e suporte), mas o desenvolvimento foi interrompido. A partir disso, a empresa precisava de uma consultoria que avaliasse o que já existia e atestasse a viabilidade de continuar o projeto a partir do código entregue. O desafio acontecia no contexto da pandemia (necessidade de soluções online) e na transição da Britânia para práticas ágeis, exigindo equilíbrio entre velocidade e governança. Além disso, era necessário entender uma ferramenta de referência, levantar requisitos sem “reinventar a roda”, respeitar padrões de desenvolvimento próprios para facilitar manutenção/internalização e definir uma arquitetura robusta e modular, mesmo com escopo ainda pouco detalhado e risco de estimativas imprecisas. Para endereçar isso, a Visionnaire iniciou com um diagnóstico do código-fonte produzido pelo fornecedor anterior sob quatro visões: Arquitetura, UI/UX, *Frameworks* (Node.js/React) e Gestão.

O diagnóstico envolveu reuniões entre as equipes, demonstrações do sistema no estado atual, encontros virtuais e presenciais, visitas às fábricas em Joinville e acompanhamento *in loco* do uso das ferramentas, culminando em apresentações de conclusões. Ao final, ficou constatado que era possível seguir o desenvolvimento a partir do código existente. Com a validação e o trabalho de requisitos/definição técnica, o projeto evoluiu para prototipação e desenvolvimento, com a Visionnaire apoiando a equipe interna da Britânia até que a própria contratante assumisse a continuidade.

Como resultado, a Britânia conseguiu concluir o CRM que estava parado, com economia por reaproveitar o que já havia sido construído e com uma base mais segura para evoluções futuras. O case também destaca ganhos de produtividade e redução de custos via equipe especializada, processos ágeis e baseados em nuvem, com uso de *Scrum* e *sprints* semanais.

Tecnologicamente, o sistema utilizou JavaScript (Node.js), com NestJS, React e Redux, banco Microsoft SQL Server (com usos pontuais de NoSQL), integração com TOTVS e hospedagem nos servidores da Britânia.