**Tribo Operações**

**Data da Entrevista: 21/12/2021**

**Superbid: Bemerson e Tiago Melin**

**Lsinn: AQ e TN**

**Sistema de agendamento de visitas:**

Anteriormente, os dados eram buscados via API para acesso direto a fonte de dados no Banco Oracle com arquitetura monolítica, com retorno dos dados para tratamento, atualmente, não existe mais a API para buscar as informações direto na fonte, o front-end possui um BFF com integrações por API, com conceito de microsserviços para busca de dados.

Houve a alteração da arquitetura para microsserviços, com a concepção de uma camada intermediária (BFF) que chama o terceiro responsável por acessar o banco de dados (accounts, por exemplo), com a busca de um dado genérico que serve para diferentes fontes, com tratamento dos dados e devolução para o front-end.

Para a concepção da solução, os desafios estão presentes devido a necessidade de escalabilidade da solução, que precisar funcionar em paralelo. Além disso, o erro em um microsserviço não pode impactar na solução como um todo. Para sanar o desafio, foi desenvolvido o BFF (intermediário) e um novo processo para atender ao banco de dados.

Desafio consistiu em como homologar esse sistema, para autenticação, além da redundância de dados, devido a necessidade de diversos banco de dados para atender a cada microsserviço, por conta da API genérica buscar todos os dados que precisam ser tratados. Para sanar o desafio, está sendo estudada para ser realizada a homologação de eventos com o Kafka combinado aos microsserviços para que sejam reportados os dados com necessidade de atualização ao BFF. Estão sendo realizados testes para homologação e modelagem.

**Arquitetura de Microfontends para Plataforma:**

Desenvolvimento de uma nova plataforma pela equipe da Superbid, para sanar o problema identificado nos leilões no qual os clientes (vendedores) não conseguiam acompanhar os leilões em tempo real, além de o comprador não ter a visão após conseguir comprar o produto, sem saber o passo a passo da negociação, a partir da concepção de um novo front-end desenvolvida por microfrontends em React, devido a possibilidade do desenvolvimento de uma solução em módulos, minimizando os problemas causados em um sistema monolito, além disso foi utilizado Kclock.

Além disso, está sendo aplicado conceitos de design system e documentação do UX (Storybook), possibilitando que cada usuário consiga enxergar os dados.

Devido a nova arquitetura, desafios foram encontrados para o desenvolvimento da nova plataforma, devido ao baixo conhecimento da equipe com a nova solução. Testes unitários estão sendo realizados para melhoria da experiência e mitigação de bugs para quando for realizado o deploy da aplicação.

Além disso, foi identificado que no momento do deploy dos módulos do servidor existe a necessidade de atualizar o módulo principal, com as atualizações dos módulos de forma individual, torna preciso a atualização do caminho para chamar o módulo, foi identificado que se caso dois módulos estivessem fazendo atualização ao mesmo tempo, acabava apagando o caminho e as informações anteriores. Devido a falta de know how da equipe, não foi possível sanar o desafio identificado, fazendo necessário o apontamento da identificação do novo serviço manualmente no módulo principal (container), com a utilização do Github tornando possível que os commits sejam atualizados antes de serem realizados os desenvolvimentos, possibilitando assim realizar o push sem comprometer a atualização.

\*Storybook – documentação dos componentes de UX