

Hackathon STARK BANK - ITA

Grupo: Kalil e amigos



Membros:

Kalil Georges Balech

Marcus Gabriel de Almeida Nunes

Mathias Cauet Martins de Sousa Damasceno

27 de Maio de 2023

Resumo

O foco do *Hackathon* é a criação de aplicações financeiras que utilizem a estrutura tecnológica do Stark Bank. Inicialmente havíamos decidido apenas pela criação de uma aplicação que informa o usuário da conta do Stark Bank por boletos que estão com a data de vencimento como a data do dia atual, mas que ainda não foram pagos. Esse recurso seria bastante útil para evitar multas de atraso de pagamento, de forma a fornecer maior controle financeiro para os usuários do Stark. Tal aplicação seria implementada utilizando a consulta de boletos por CNPJ, informando o usuário via slack. Devido à falta de recursos via API de pesquisar todos os boletos emitidos para um determinado CNPJ cliente, no entanto, foi-se necessário adicionar novos recursos ao projeto, de modo a torná-lo mais completo e interessante.

Dessa forma, houve a adição desses novos recursos ao projeto. Isso se deve, pois, ao fim, criamos duas aplicações, em que a primeira tem como objetivos tanto a criação de boletos, geração de seu respectivo PDF e envio automático para o e-mail do CNPJ cliente, assim como, a segunda aplicação tem a função de gerar extratos diários de boletos que estão perto de vencer (vencimento no dia atual) e o envio de tal extrato, que está relacionado ao um determinado centro de custo da empresa, via *Slack* para um usuário cadastrado. Tais projetos foram escolhidos com a finalidade de diminuir o tempo gasto para notificar o cliente da criação de um novo boleto em seu CNPJ, (dado que agora os email são enviados automaticamente para o email registrado em seu centro de custo, de forma que a visão computacional da stark rapidamente identifica esse boleto e o adiciona em sua base de dados) e para evitar possíveis atrasos de pagamentos de boletos.

Com a finalização deste projeto, foi possível obter um resultado convincente, validando que os produtos foram desenvolvidos com sucesso para suas finalidades específicas. Durante o processo, demonstrou-se efetivamente a capacidade de enviar PDFs por e-mail e agendar a geração diária dos extratos de boletos. Essas funcionalidades foram implementadas com êxito, demonstrando a viabilidade técnica e funcional das soluções propostas.

1. Introdução

O principal objetivo do *Hackathon* foi promover o desenvolvimento técnico e o espírito empreendedor dos estudantes do ITA, desafiando-os a conceituar e criar produtos de tecnologia. O foco era incentivar os participantes a pensar de forma inovadora para solucionar problemas empresariais, utilizando as soluções fornecidas pelo *Stark Bank*.

Ao longo do projeto, os membros da equipe tinham como objetivo não apenas entregar aplicações funcionais, mas também desenvolver suas habilidades tecnológicas e lógicas (tanto na parte de código, quanto na parte de regras de negócio). A escolha dos membros foi baseada no interesse de trabalhar em conjunto, uma vez que já nos conhecíamos anteriormente e reconhecemos que nossas competências individuais se complementaríamos durante a resolução desse desafio.

2. Descrição do projeto

O objetivo deste projeto foi desenvolver duas aplicações utilizando *Node.js* para o *back-end* e *HTML* com *CSS* puro para o *front-end*. Utilizamos o *Node.js* devido a familiaridade de integrantes do grupo com a tecnologia e não utilizamos nenhuma *framework front-end* para que nossa aplicação se aproxime das aplicações da *Stark Bank*, dado que com tecnologias independentes, existe uma liberdade de desenvolvimento maior, sem se tornar dependente do fornecimento de recursos de terceiros.

Inicialmente, a proposta era criar uma aplicação que enviasse, por meio do *Slack*, todos os boletos com vencimento anterior ou igual à data do dia atual. No entanto, devido à falta de rotas na *API* disponibilizada para pesquisar os boletos a partir do próprio CNPJ do cliente, decidimos desenvolver duas aplicações.

As aplicações foram criadas com propósitos distintos, mas complementares, ambas relacionadas a boletos. A primeira aplicação teve como objetivo o compartilhamento automático de boletos criados a partir de uma conta no *Stark Bank*, por meio de e-mail. Já a segunda aplicação consistiu na criação de um *Slackbot*, que permitia estabelecer uma relação entre o ID do membro no *Slack* e um centro de custo por meio de um cadastro, e assim gerar um relatório de boletos com vencimento no dia atual e também um alerta de saldo baixo na conta.

3. Metodologia

Primeira aplicação

Inicialmente foi consentido entre os membros as linguagens de programação a serem utilizadas para a esta aplicação, a partir dessa escolha, montou-se uma página *web* que continha espaços destinados a receber:

- Valor do boleto a ser gerado
- Nome e informações do endereço do destinatário .
- Cnpj do destinatário.
- Email do destinatário.
- Vencimento do boleto a ser criado.

A partir destas informações e utilizando as *API's* do *Stark Bank*, é possível criar um boleto que fica registrado na base de dados da *sandbox* do *Stark Bank*. Por meio da consulta ao boleto criado, através de seu id, é possível gerar um arquivo *pdf*, novamente chamando as *API's*. Por fim, enviou-se um email com este arquivo para o email do destinatário definido em um dos *inputs* da página *web*. Para este envio, utilizou-se como remetente um email pessoal de um dos integrantes da equipe, porém, para aplicações reais, este email seria um do domínio do *Stark Bank*.

Dessa forma, caso o email do destinatário seja um Mailbox haverá a integração automática do boleto com um centro de custo em específico. Isso é bastante útil, pois apenas com email vinculados a centros de custos dos clientes, poderíamos realizar uma consulta pela série de boletos, e, assim, desenvolver a nossa aplicação inicial, que seria o envio do extrato de boletos que estão perto do vencimento para o cliente, de modo a auxiliá-lo a evitar o pagamento de multas de atraso.

Segunda aplicação

Assim como ocorreu no outra aplicação, os membros escolheram os recursos tecnológicos antes de começar qualquer tipo de criação. Novamente semelhante à aplicação anterior, houve a criação de uma página *web* que recebia os seguintes dados:

- ID do membro no Slack.
- ID do centro de custo.
- Valor para aviso de baixo saldo (opcional)

Para maior segurança, criou-se uma autenticação, a qual envia um token em formato de mensagem no *Slack* do usuário de ID digitado e este token deve ser inserido para autenticação. Esse mecanismo foi desenvolvido como uma verificação se o usuário que interage com a tecnologia é realmente o dono do perfil slack informado.

A partir destes dados, utilizando-se um programa criado localmente foi possível relacionar diretamente o boleto a um centro de custo com um canal no Slack. Posteriormente, utilizando as *API's* do *Stark Bank*, foi possível enviar como mensagem, programada para acontecer diariamente, todos os boletos que vencem na data atual e que estavam no determinado centro de custo para o usuário cadastrado. Neste extrato de boletos, manteve-se aqueles com *status* igual a *denied* para que o usuário possa conferir se a negação do pagamento estava correta ou não. Isso, porém, é facilmente personalizável ao ter acesso ao programa.

Além disso, criou-se uma função, que a cada determinado período de tempo especificado, há a verificação se o saldo da conta está abaixo do limite escolhido. Caso isso ocorra, há o envio de uma mensagem no *Slack* para avisar o usuário.

4. Resultados

Através da primeira aplicação, conseguimos criar, gerar o PDF e compartilhar via email os diversos boletos que a nossa aplicação é capaz de produzir. Esse boleto, após enviado para a MailBox, é registrado na base de dados do *Stark Bank*, e pode ser acessado via a nossa segunda aplicação. Dessa forma, houve a automatização desse processo, que, até então, não era feito pela infraestrutura da *Stark Bank*.

Quanto à segunda aplicação, devido à restrição de tempo para testes, foi feita uma modificação para que as mensagens não fossem enviadas diariamente, mas sim em intervalos específicos durante os testes. Durante esses testes, foram observadas as reações esperadas, como a alteração de status e a inserção de boletos com vencimento na data do teste. Além disso, é importante destacar que o código pode ser facilmente adaptado caso seja necessário enviar um extrato contendo diferentes tipos de pagamentos, como transferências, boletos, contas de concessionárias, entre outros. Assim, a aplicação desenvolvida alerta, de maneira simples, os usuários das contas *Stark Bank* caso ele possua boletos pendentes para o dia de hoje, de modo a evitar o pagamento de multas, assim como caso o saldo da sua conta bancária está abaixo de certo valor pré-definido.

5. Próximos passos

Apesar dos recursos desenvolvidos pelo nosso grupo funcionarem e ambas as aplicações já serem capazes de resolverem problemas reais, claramente, ainda há melhorias a serem feitas para que a nossa estrutura entre em produção. Para cada uma das nossas aplicações há as seguintes melhorias que podem ser realizadas:

- Primeira aplicação:
 - Mudança do email remetente para um email de avisos da empresa.
 - A criação de boleto ocorre internamente no ambiente do *Stark Bank*.
 - Tratamento e validação dos dados de entrada
 - Verificação e garantia que a geração de boletos é feita por humanos
- Segunda aplicação:
 - Adicionar a possibilidade de requisitar o extrato de boletos.
 - Adição da escolha do horário do extrato pelo usuário.
 - Verificação se o dono da conta Slack possui alguma relação com o centro de custos informado
 - Tornar livre o acesso ao *Slack Bot* por vários *workspaces* e usuários.
 - Aumentar a segurança das informações transmitidas, dado que são sensíveis

6. Desafios

Durante o desenvolvimento da aplicação discutida, foram encontrados inúmeros desafios: alguns deles mais simples e outros mais complexos. Entre eles, estão: a dificuldade inicial de consultar todos os boletos que estão relacionados a um mesmo CNPJ cliente - problema superado com o desenvolvimento da nossa aplicação de geração de boletos- e dificuldade em garantir o mínimo de segurança para a aplicação - problema que ainda persiste, mas que foi, de alguma forma tangenciado pelo desenvolvimento da autenticação do usuário da conta slack, via geração de código de confirmação e criptografia, além do armazenamento de informações nocivas via variáveis de ambiente. Dessa forma, nota-se que mesmo aplicações simples podem ser suficientemente desafiadoras quando um grau de excelência mínimo é requerido.

Além dos desafios mencionados acima, outros problemas mais simples foram encontrados. Dentre inúmeros, vale-se mencionar a dificuldade no gerenciamento de permissões de um projeto - dificuldade contornada em conversa por *WhatsApp* com o Dalton, que nos ajudou bastante, fornecendo uma api que não era acessível via *sandbox*, mas que foi utilizada via postman -, dificuldade na integração entre o *front-end* e o *back-end* - desafio vencido através da utilização de bibliotecas simples do *javascript* como *express*, *cors* e *dotenv* - e dificuldades comuns do desenvolvimento de aplicações web, como manipulação de variáveis em objetos

(req, res) de requisições http, utilização da biblioteca de criptografia, entre outros problemas comuns que surgiram pelo caminho.

7. Conclusão

Com base nos dados obtidos dos testes realizados, conclui-se que as aplicações atendem de forma extremamente satisfatória aos objetivos estabelecidos anteriormente. Isso ocorre, pois apesar de ainda podermos implementar melhorias tanto de segurança quanto funcionais ao projeto, os recursos implementados são funcionais e já estão aptos a auxiliar a vida ou rotina financeira de usuários do Stark Bank.

Ao utilizar a primeira aplicação, há uma economia de tempo para o remetente, uma vez que o envio do PDF do boleto é automatizado. Além disso, quando enviado para uma caixa de correio, o destinatário economiza tempo, pois é adicionado automaticamente ao centro de custo. Para uma empresa, essa economia de tempo pode ser direcionada para outras tarefas que atualmente não possuem automação.

No caso da segunda aplicação, os responsáveis financeiros recebem os extratos de pagamento em seus canais do *Slack*, garantindo que nenhum pagamento seja atrasado, o que resultaria em multas e juros. Além disso, essa mesma aplicação previne que um pagamento seja erroneamente negado por algum funcionário, também resultando em multas se não for pago na data correta.

Portanto, concluímos que as aplicações desenvolvidas são funcionais e podem ter um impacto real na rotina diária das empresas. Assim, julga-se que a versão atual da aplicação desenvolvida possa ser incorporada ao marketplace de aplicativos do *Stark Bank* quando tal ambiente de recursos for lançado.