Foi realizado um trabalho em quatro ambientes diferentes do aplicativo Linkest, a primeira área foi a infraestrutura, foi feita alterações para suportar o aplicativo com suas atualizações; a segunda área foi a API (Interface de programação de aplicações) do aplicativo ,foi criado e atualizado de acordo com a prioridade do cliente; a terceira área foi o plugin (programa que fornece uma função ou melhoria a um programa maior) para o browser do google chrome, foi criado e melhorado com suas atualizações; e a última área foi o chatbot (programa que auxilia na comunicação do aplicativo com o usuário) do aplicativo para o Facebook.

3.1 Primeiro ambiente

Na infraestrutura foi utilizada a arquitetura de três camadas no software, porém foram necessárias duas mudanças essenciais para poder evoluir o aplicativo para um modelo B2B. As mudanças necessárias foram:

* Adição e implementação da linguagem Docker na camada mais baixa da arquitetura com o intuito de tornar mais escalável o aplicativo e deixar mais fácil a correção, manutenção do código e adição de novas funcionalidades no sistema final.
* Outra mudança foi a troca de servidor Heroku para o servidor Azure, facilitando um aumento de recebimento de dados e tornando mais seguro a aplicação; além disso, foi um pedido do cliente a utilização do Azure.

Na implementação da linguagem Docker foi utilizado um Framework chamado Terraform, que permite a criação de uma infraestrutura total a partir de um único script. Neste script informa-se qual o sistema operacional (S.O.) que será utilizado na máquina virtual (um arquivo de computador que se comporta como um computador de verdade), qual o servidor final será utilizado (transformando a infraestrutura compatível com esse servidor), além poder informar o tipo e quantidade de conexões que o sistema possa atender.

O produto final da utilização do Framework é um pacote de arquivos que serão enviados ao servidor proporcionando a infraestrutura do sistema e a criação de um arquivo Docker, onde o programador define e especifica containers que serão utilizados na programação posterior.

Foi mantido a utilização do CodeShip (empresa que fornece serviço de integração contínua e com suporte para Git, Bitbucket, etc.) para a integração contínua e do Cloudflare (empresa que fornece uma rede de distribuição de conteúdo, serviços de segurança da Internet e serviços de servidor de nome de domínio distribuído) para o registro do domínio.

3.2 Segundo ambiente

Na parte da API foi necessária uma reestruturação com mudança de linguagem de programação, a retirada de algumas e adição de novas funcionalidades requeridas pelo cliente para que fosse possível a criação de um MVP (produto mínimo viável) funcional.

Anteriormente, o sistema foi implementado com a linguagem PHP (Hypertext Preprocessor) e com a utilização do Framework CakePHP, porém esta utilização tornava a API muito dependente do Front-End do sistema e com dificuldade de manipular grandes quantidades de dados. A escolha da nova linguagem foi decidida para resolver os problemas presentes na antiga versão, por isso a linguagem escolhida foi Python e com a utilização do Framework Eve, pelo fato dele permitir de criar e implantar facilmente serviços Web RESTful personalizáveis com recursos essenciais e de tornar mais fácil a manipulação de dados com o suporte nativo com o banco de dados MongoDB, permitindo a utilização do MongoEngine (Mapeador de documentos de objeto para MongoDB), tornando mais simples e funcional o trabalho com os dados.

No sistema Linkest foi necessária a instalação dos pré-requisitos e copiar os arquivos criados de um novo projeto utilizando Eve na máquina virtual (arquivo de computador, normalmente chamado de imagem, que se comporta como um computador) criada no Docker, para ser possível a criação de um container direcionado para ser a API. Também foi necessária a criação de uma novo container para receber o MongoDB. Logo após, essa etapa foi possível

a implementação das funcionalidades na API do sistema. A utilização CodeShip proporcionou uma facilidade de atualização do sistema com os updates presentes no Bitbucket (Sistema de controle de versão utilizado na implementação do projeto).

3.3 Terceiro ambiente

Este ambiente foi direcionado para o Front- End do sistema, anteriormente o Linkest tinha uma página Web, mas a partir de pedido do cliente, foi criado um plugin do navegador Google Chrome que realizaria a função de Front-End do sistema. O plugin foi criado com a linguagem Javascript com a utilização da biblioteca React, permitindo a criação de uma interface simples, funcional e intuitivo. A interface criado com React é baseado por componentes que são administrados pelos próprios estados que permite a manipulação de dados complexos no contexto daquele componente. O plugin demandava as informações a API através de promessas feitas utilizando XMLHttpRequest (objetos que interagem com os servidores recuperando dados de um URL, Uniform Resource Locator, sem ter que fazer uma atualização de página inteira, permitindo que uma página da Web atualize apenas parte de uma página sem interromper o que o usuário está fazendo) e recebia os dados em formato JSON (JavaScript Object Notation), que era tratada por funções em Javascript e exibida de forma intuitiva ao usuário com o React.

Ao passar do tempo, a interface foi modificada várias vezes para ficar mais “amigável” e a pedido do usuário. A implementação e teste da interface foi realizada de forma local e quando ela se tornou mais próxima do produto final descrito pelo cliente, foi feito o upload ao Chrome Web Store (Página de plugins da Google Chrome) para poder ser disponibilizada ao público.

3.4 Quarto ambiente

O último ambiente trabalhado foi o Chatbot direcionado ao Facebook e ao Slack, no Linkest não existia este ambiente, portanto ele foi criado a pedido do usuário. Ao início, o chatbot iria funcionar no Facebook e no Slack, porém com o passar do tempo o usuário pediu a implementação somente do Facebook.

Para a criação do chatbot, foi necessário criar uma aplicação que servia de intermediário entre a API e entre o Facebook, que manipulava as informações recebida da API e mandava ao Facebook, por sua vez direcionava a mensagem ao usuário, a aplicação teve que ser adicionada a um novo container para permitir sua funcionalidade. A aplicação foi criada na linguagem Javascript e foi feita seguindo os tutoriais disponibilizados pelo próprio Facebook, no site de Facebook for developers.

Foi necessário a criação de uma página no Facebook para o chatbot, para permitir que o chatbot pudesse relacionar com o usuário através de mensagens da página. O usuário com o passar do tempo pediu melhorias e mudanças no formato das mensagens do chatbot, que todas foram atendidas.

Construção do Sistema

A construção do sistema foi realizada em passos, o primeiro passo foi construir o primeiro ambiente e o terceiro ambiente em paralelo, pois o primeiro ambiente seria a base para todo o sistema e o terceiro ambiente tem muito haver com a parte gráfica e front-end do sistema e não necessitava tanto da base para iniciar; Em seguida, foi realizada a construção do segundo ambiente e seus respectivo testes unitários; Logo após, foi realizada a integração do segundo ambiente, o primeiro ambiente e o terceiro ambiente e foi realizado testes com o usuário; Em seguida, foi construído o quarto ambiente e feito seus testes unitários; Prosseguindo, foi feita a integração do quarto ambiente no sistema e foi disponibilizado ao usuário para testes; E terminando, foi realizada às melhoria e consertos de bugs no sistema em geral, ouvindo os feedbacks dos usuários e do cliente.

A implementação do sistema foi realizada usando os conhecimentos aprendidos no curso de TADS, seguindo tutoriais, dicas e materiais das documentações das tecnologias utilizadas disponíveis na internet.