



LUCAS SILVA SILVÉRIO

**DESENVOLVIMENTO DE SOFTWARE *FRONT-END* NA
EMPRESA NEXOOS/AME**

LAVRAS – MG

2023

LUCAS SILVA SILVÉRIO

DESENVOLVIMENTO DE SOFTWARE *FRONT-END* NA EMPRESA NEXOOS/AME

Trabalho de conclusão do curso apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Sistemas de Informação, para a obtenção do título de Bacharel.

Prof. Dr. Bruno de Abreu Silva

Orientador

LAVRAS – MG

2023

LUCAS SILVA SILVÉRIO

DESENVOLVIMENTO DE SOFTWARE *FRONT-END* NA EMPRESA NEXOOS/AME

Trabalho de conclusão do curso apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Sistemas de Informação, para a obtenção do título de Bacharel.

APROVADA em 18 de Julho de 2023.

| | |
|--|--------|
| Prof. Dr. Bruno de Abreu Silva | UFLA |
| Prof. Dr. Paulo Afonso Parreira Junior | UFLA |
| Ubirajara Chagas Pelli | Nexoos |

Prof. Dr. Bruno de Abreu Silva
Orientador

**LAVRAS – MG
2023**

Dedico principalmente a Deus, à minha família, aos meus pais Nádia Silvério e Wellington Silvério e meu irmão Matheus Silvério.

AGRADECIMENTOS

Primeiramente eu gostaria de agradecer a Deus, que sempre esteve ao meu lado me ajudando, dando forças, coragem e segurança em todos os dias em que estive na universidade, aos meus pais e irmão que foram meus pilares na minha trajetória e sempre me deram condições, encorajaram a ir atrás dos meus sonhos, independente da situação sempre estiveram do meu lado.

Também tenho de agradecer a todos envolvidos que estiveram neste processo direta e indiretamente: à toda minha família que sempre esteve comigo me dando suporte em todos os momentos, bons ou ruins, aos professores do DCC/UFLA e funcionários da universidade que me deram suporte nesta caminhada e a Zetta, pelos primeiros ensinamentos, e a Nexoos que me deram uma oportunidade de aprendizado muito preciosa.

Um agradecimento especial aos amigos que estiveram comigo durante o tempo de curso, em especial Ruan Lourençoni, Victor Marcondes, Pedro Freitas e Pedro Menezes que sempre estiveram comigo e tornaram a faculdade mais divertida. Por fim, ao professor Bruno de Abreu Silva pela orientação com tanta excelência e paciência durante o desenvolvimento deste relatório.

*“Eu quero ser maior que essas muralhas que eles construíram ao meu redor”
(BK’)*

RESUMO

Buscando ser uma *fintech* de referência no mercado e garantir seu espaço no mercado financeiro, a Nexoos investe seus esforços para conseguir uma base tecnológica adequada, com um sistema eficiente, intuitivo e seguro. Desta forma, este relatório de estágio supervisionado tem por fim relatar o processo de desenvolvimento do sistema interno da Nexoos/AME tanto mobile quanto a página WEB. O sistema é separado em algumas frentes diferentes como, por exemplo, a área de investidores, solicitação de crédito, gerenciamento de propostas, análise de proposta e parceiros, entre outras. Cada uma delas atende a uma demanda diferente do sistema financeiro, pois os fundos de solicitações ou os modelos de investimentos podem ser diversos. Durante o estágio foram realizadas atividades em vários projetos, porém só foi autorizada a abordagem de um deles. O sistema em questão foi criado, corrigido ou refatorado durante o período de estágio na Nexoos. Para o desenvolvimento foram utilizadas, principalmente, as tecnologias de Ruby on Rails, React e JavaScript, e o Framework Scrum para organização de atividades. Assim, foi realizada a participação nos processos da empresa, com o resultado da criação do novo Fluxo de Solicitação da Nexoos.

Palavras-chave: Fintech. Banco. Mercado Financeiro. Investimento. Empréstimo. Framework Scrum. React. Ruby on Rails. Postgresql. Front-end. HTML. CSS. TypeScript. JavaScript

ABSTRACT

Seeking to be a reference fintech in the market and guarantee its space in the financial market, Nexoos invests its efforts to achieve an adequate technological base, with an efficient, intuitive and secure system. In this way, this supervised internship report aims to report the development process of the Nexoos/AME internal system, both mobile and the WEB page. The system is separated into a few different fronts such as, for example, the investor area, credit request, proposal management, proposal analysis and partners, among others. Each one of them meets a different demand from the financial system, as request funds or investment models can be different. During the internship, activities were carried out in several projects, but only one of them was authorized to approach. The system in question was created, corrected or refactored during the internship at Nexoos. For development, Ruby on Rails, React and JavaScript technologies were used, and the Scrum Framework for organizing activities. Thus, participation in the company's processes was carried out, with the result of the creation of the new Nexoos Request Flow.

Keywords: Fintech. Bank. Financial market. Investment. Loan. Scrum Framework. React. Ruby on Rails. Postgresql. Front-end. HTML. CSS. TypeScript. JavaScript

LISTA DE FIGURAS

| | |
|--|----|
| Figura 2.1 – Funcionamento do Scrum | 15 |
| Figura 2.2 – Demonstração de JavaScript | 16 |
| Figura 2.3 – Demonstração de Typescript | 17 |
| Figura 2.4 – Exemplo de HTML | 18 |
| Figura 2.5 – Exemplo de código em CSS | 19 |
| Figura 2.6 – Demonstração de React | 20 |
| Figura 3.1 – Fluxo de solicitação legado | 25 |
| Figura 3.2 – Fluxo de solicitação refatorado | 26 |
| Figura 3.3 – Fluxo antigo - Inserção de CNPJ | 27 |
| Figura 3.4 – Fluxo antigo - Inserção de Faturamento | 27 |
| Figura 3.5 – Fluxo antigo - Tela de dados de pessoa física | 28 |
| Figura 3.6 – Nexoos - Solicitação de crédito - inserção de valor | 29 |
| Figura 3.7 – Nexoos - Solicitação de crédito - valor inserido | 30 |
| Figura 3.8 – Trecho do código-fonte do projeto | 31 |
| Figura 3.9 – Nexoos - Solicitação de crédito - Inserção de Dados | 32 |
| Figura 3.10 – Nexoos - Solicitação de crédito - Tela de carregamento | 33 |
| Figura 3.11 – Trecho do código-fonte do projeto | 34 |
| Figura 3.12 – Nexoos - Solicitação de crédito - Tela de aceite | 35 |
| Figura 3.13 – Nexoos - Solicitação de crédito - Tela de Recusa | 36 |
| Figura 3.14 – Nexoos - Solicitação de crédito - Tela de Erro | 37 |

SUMÁRIO

| | | |
|----------------|--|-----------|
| 1 | INTRODUÇÃO | 9 |
| 1.1 | Empresa | 9 |
| 1.2 | Objetivos | 10 |
| 1.3 | Estrutura do restante deste documento | 11 |
| 2 | Conceitos e Tecnologias utilizadas | 12 |
| 2.1 | Metodologias Ágeis | 12 |
| 2.1.1 | Framework Scrum | 12 |
| 2.2 | Tecnologias Utilizadas | 15 |
| 2.2.1 | Desenvolvimento <i>Front-end</i> | 15 |
| 2.2.1.1 | JavaScript | 16 |
| 2.2.1.2 | TypeScript | 17 |
| 2.2.1.3 | HTML (HyperText Markup Language) | 17 |
| 2.2.1.4 | CSS | 18 |
| 2.2.1.5 | React | 19 |
| 2.2.2 | Versionamento de código | 20 |
| 2.2.2.1 | Git | 20 |
| 2.2.2.2 | GitHub | 21 |
| 2.2.3 | Gerenciamento de Projeto | 22 |
| 2.2.3.1 | Jira | 22 |
| 3 | Atividades Desenvolvidas | 23 |
| 3.1 | Treinamento e adaptação | 23 |
| 3.2 | Fluxo de Solicitação | 24 |
| 3.2.1 | Explicação do Sistema Legado | 24 |
| 3.2.2 | Inserção do valor | 28 |
| 3.2.3 | Inserção dos Dados | 31 |
| 3.2.4 | Tela de carregamento | 32 |
| 3.2.5 | Resultado da Análise | 34 |
| 4 | CONCLUSÃO | 38 |
| | REFERÊNCIAS | 40 |

1 INTRODUÇÃO

Para concluir e obter o título de Bacharel em Sistemas de Informação da UFLA, uma das exigências é a aprovação do trabalho de conclusão de curso. Este trabalho pode ser apresentação por meio de diferentes modelos, um dos quais é o relatório de estágio, desta forma, os alunos realizam estágios em organizações. A prática profissional é uma das possibilidades de aplicação do conhecimento obtido durante o estudo. Este relatório apresenta a atividade aplicada das empresas de tecnologia e o conhecimento adquirido no processo de seleção ao longo de um ano, o período de estágio. A organização oferece oportunidades que nos dão uma primeira exposição a um ambiente profissional e oportunidades de aprendizado que só podem ser obtidas através da prática em um ambiente organizacional real. Este documento tem por objetivo relatar o trabalho de conclusão de curso do autor, realizado na modalidade Estágio em uma Organização. Sendo assim, este capítulo traz uma contextualização do estágio supervisionado e uma apresentação da empresa Nexoos, onde o estágio foi efetuado. Além disso, serão apresentados os objetivos do trabalho e a estrutura do restante do documento.

1.1 Empresa

O estágio foi feito de agosto de 2021 e continua no atual momento em que o relatório está sendo escrito com contrato até o mês de outubro de 2023. O estágio foi realizado na Nexoos¹, uma empresa do grupo AME e visa democratizar o sistema financeiro, fomentando melhores oportunidades de negócios para investidores em potencial e pequenas e médias empresas que desejam um valor por empréstimo. Uma das formas de ligação entre o investidor e uma empresa que solicitou uma quantia em dinheiro, por exemplo, é chamado de *peer-to-peer*. A empresa também fornece API (Interface de Programação de Aplicação) para parceiros que desejam utilizar a plataforma desenvolvida pela Nexoos.

Idealizada em Londres em 2014, a Nexoos ganhou forma após o fundador e CEO Daniel Gomes vencer o concurso de startups *Bright Ideas Awards*². A Fintech nasceu da vontade de trazer mudanças para as pequenas e médias empresas (PMEs), democratizar o sistema financeiro e impulsionar a economia.

¹ <<https://www.nexoos.com.br/>>

² <www.brightideasinhealth.org.uk>

Em setembro de 2016, a Nexoos lançou seu primeiro negócio de crédito no Brasil após a conclusão de um projeto-piloto no Paraguai. Em 2019, tornou-se a primeira fintech de crédito a operar como uma sociedade de empréstimos entre pessoas (SEP), regulada pelo banco central.

Atualmente, a Nexoos presta serviços financeiros a empreendedores com taxas competitivas e a mais de 65.000 investidores, pessoas físicas e jurídicas, que buscam rentabilidade acima da média do mercado. O diferencial é dar agilidade e segurança aos clientes por meio de processos desburocratizados e online. Hoje a companhia tem em torno de 80 funcionários, divididos em quatro setores diferentes Tecnologia, Financeiro, Recursos Humanos e Compliance e ouvidoria. Tem como sua base o escritório em São Paulo e todos os funcionários trabalham em regime *home-office*, exceto o setor de crédito que faz o sistema híbrido.

A equipe de tecnologia da Nexoos, onde o estágio está sendo realizado, trabalha com Metodologia de Desenvolvimento Ágil a partir do *Framework* Scrum. Sendo assim, toda a estrutura de tecnologia da empresa é realizada segundo os princípios do Scrum, portanto, os funcionários são divididos em times onde tem um *Product Owner* e um time de desenvolvimento. Atualmente, os times de desenvolvimento são divididos em quatro times diferentes, três de *back-end* e um específico para o *front-end*, onde foi realizada a maioria do estágio. Os times de *back-end* tem um foco específico e é responsável por um produto. O time de investidores, o de sistema interno, o de parceiros são alguns exemplos de separação. Já a equipe de *front-end* é apenas uma que vai atendendo as demandas criadas pelos *Product Owners* do time de *back-end*, podendo estar dividido em até dois grupos diferentes.

1.2 Objetivos

Considerando a organização em questão (Nexoos) que trabalha no mercado financeiro, o estágio teve como objetivo fazer a manutenção e melhoria de todo o sistema, tanto interno, o que é utilizado pelos próprios funcionários, quanto externo, que são as páginas abertas que atendem os clientes. Pois o sistema existente é o mesmo desde a criação da empresa, portanto necessitava de novos recursos e alguns até mesmo uma refatoração. Por conta disto, já não estavam mais cumprindo sua finalidade ou estavam com suas tecnologias muito desatualizadas, o que dificultava sua manutenção.

No estágio em questão, as atividades envolviam tanto o desenvolvimento de novas funcionalidades, quanto a manutenção das funcionalidades já presentes. Dentre elas, a melhora do fluxo de análise e desembolso, pois grande parte das análises das empresas solicitantes eram

feitas manualmente, portanto a criação de uma análise digital foi necessária, um tempo depois foi necessário fazer uma refatoração completa criando outro sistema que substitui o sistema em questão. Após essa primeira refatoração foi decidido pela diretoria que todos os sistemas internos (que não eram de empresas parceiras) seriam refatorados para receber uma renovação e uma nova identidade visual da empresa e uma atualização das tecnologias utilizadas. Assim, o autor do trabalho participou da criação do projeto de solicitação.

Do ponto de vista da melhoria das habilidades do estagiário, o estágio teve como intenção:

- Introduzir o estagiário no mercado de trabalho, com vivências reais de como é o dia-a-dia na empresa;
- Amadurecer os conhecimentos técnicos e *soft-skills* adquiridos durante a formação acadêmica, estando em contato com problemas mais complexos;
- Contribuir com a empresa e, juntos, buscar um crescimento em um ambiente colaborativo.

Durante o estágio, a empresa possibilitou ao estagiário procurar a função onde melhor se adaptasse, como poder mudar de *back-end* para *front-end* ou de algum desses para a função de *Product Owner*, entre outras possibilidades. Isso fez com que ele pudesse se encontrar no lugar onde mais se encaixava, podendo decidir seguir sua carreira. Com isso, durante o tempo de estágio o autor esteve em todas as três possibilidades citadas, começou no desenvolvimento *back-end* durante quatro meses, logo após passou para o *front-end* e, por fim, teve uma pequena passagem pela área de *Product Owner* de um mês e meio. Mas em sua maior parte, o estágio foi feito com a equipe de *front-end*, onde continua no momento em que este relatório está sendo feito.

1.3 Estrutura do restante deste documento

O restante do trabalho está organizado da seguinte forma: (ii) o Capítulo 2 apresenta as tecnologias e ferramentas utilizadas durante o estágio; (iii) no Capítulo 3 são apresentadas as principais atividades desenvolvidas durante o projeto; e (iv) por fim, no Capítulo 4 encontram-se as considerações finais deste trabalho.

2 CONCEITOS E TECNOLOGIAS UTILIZADAS

Neste capítulo serão apresentadas os conceitos necessários para entender esse relatório de estágio. Dentro desses conceitos serão apresentadas as tecnologias, ferramentas e *frameworks* utilizados durante o desenvolvimento do sistema.

O projeto realizado utilizou como tecnologias o React, que tem dentro da sua construção JavaScript, HTML5 e CSS3, incorporando o TypeScript, componentização de elementos utilizando a biblioteca de componentes MaterialUI. Como metodologia ágil para o desenvolvimento, foi utilizado o *Scrum Framework* e o Jira para gerência de atividades. E como ferramenta de versionamento de código o Git com o GitHub. Cada um dos conceitos serão abordados no decorrer do capítulo.

2.1 Metodologias Ágeis

As praticas ágeis surgiram em 2001 quando um grupo de 17 pessoas se reuniram para debater sobre abordagens da gestão de projetos, e a partir desta reunião foi desenvolvido o Manifesto para Desenvolvimento Ágil ¹ de *software* onde determinaram a existência das metodologias e seus princípios característicos. Conforme o livro (SOMMERVILLE, 2007), os métodos ágeis contam com uma abordagem iterativa para especificação, desenvolvimento e entrega de software, sendo criados principalmente para apoiar o desenvolvimento de aplicações de negócios nas quais os requisitos de sistemas mudam rapidamente durante o processo de desenvolvimento. Desta forma, temos como um dos métodos ágeis mais utilizados atualmente o *Framework Scrum*, que será apresentado a seguir.

2.1.1 Framework Scrum

Scrum (SCRUM.ORG, 2021) é uma estrutura de gerenciamento de projetos focada em entregar valor ao cliente de forma ágil e colaborativa. Foi originalmente desenvolvido para dar suporte ao desenvolvimento de *software*, mas pode ser aplicado a outros tipos de projetos. O documento oficial desse framework é o *The Scrum Guide* (SCRUM.ORG, 2020), que traz o que é as definições das partes do *Scrum* e um guia de como deve ser feito.

Trazendo uma introdução para a entender o funcionamento do *Scrum*, é necessário entender o que é uma *Sprint*. Uma *Sprint* é um tempo fixo, que, conforme o *The Scrum Guide*, pre-

¹ <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>

cisa ter entre uma ou duas semanas, no qual o time *Scrum* trabalha para completar um conjunto de funcionalidades ou tarefas definidas previamente entregando o máximo de valor possível.

O *framework* é composto por eventos que ditam o andamento do trabalho. Tais eventos são:

- *Planejamento da Sprint*: Ocorre no início de cada sprint, é quando a equipe se reúne para planejar o trabalho a ser realizado durante a próxima sprint. Durante este rito, a equipe define quais itens do *backlog* serão priorizados e como serão completados.
- *Daily*: É uma reunião diária de curta duração na qual cada membro da equipe informa o que realizou no dia anterior, o que planeja fazer hoje e se precisa de ajuda em algum obstáculo.
- *Revisão da Sprint*: Ocorre no final de cada sprint e é uma oportunidade para a equipe demonstrar o trabalho realizado ao longo do sprint e receber feedback do cliente ou do time.
- *Retrospectiva da Sprint*: É uma reunião na qual a equipe reflexiona sobre a sprint anterior, identifica pontos fortes e fracos e planeja ações para melhorar o processo na próxima sprint.

Estes ritos são realizados em um ciclo contínuo e permitem à equipe se adaptar rapidamente a mudanças e entregar valor ao cliente constantemente.

Em relação aos atores envolvidos nos processos do *Scrum*, temos:

- *Scrum Master*: É um papel importante em equipes que utilizam o framework de desenvolvimento de *software Scrum*. Ele é responsável por garantir que o processo de desenvolvimento de *software* seja realizado conforme as práticas e valores do *Scrum*.

O *Scrum Master* é o guardião da filosofia *Scrum* e trabalha para auxiliar a equipe a implementar as práticas do *Scrum* de maneira eficaz. Ele ajuda a equipe a entender e aplicar os conceitos do *Scrum*, incluindo a realização de *sprints*, revisões, planejamentos, *daily meetings*, entre outros.

Além disso, o *Scrum Master* é responsável por auxiliar a equipe a remover obstáculos e garantir que eles possam trabalhar eficientemente. Ele também é responsável por garantir que as regras do *Scrum* sejam seguidas e que o processo de desenvolvimento seja ágil e adaptável a mudanças.

- *Product Owner*: É uma pessoa responsável pela definição e gerenciamento do escopo e dos objetivos de um produto ou projeto de *software*. Ele é o principal representante do negócio ou do usuário final, com a autoridade final para tomar decisões sobre o que deve ser incluído ou excluído no produto.

O *Product Owner* trabalha em estreita colaboração com a equipe de desenvolvimento e garante que o produto atenda às expectativas do usuário final, atendendo aos requisitos e prioridades de negócios. Ele também é responsável por garantir que o produto esteja alinhado com a estratégia de negócios e esteja avaliando continuamente o valor e o impacto do produto.

- *Time de Desenvolvimento*: É uma equipe formada por membros com habilidades e conhecimentos diversos, que trabalham juntos para desenvolver um produto ou solução. O time de desenvolvimento é altamente autônomo e responsável por suas próprias decisões. Eles se comprometem em trabalhar juntos para alcançar as metas do projeto, mantendo um alto nível de transparência e colaboração.

Por fim, temos os artefatos do *Scrum*, usados para detalhar o produto que está sendo desenvolvido, são eles:

- *Backlog do Produto*: É uma lista de itens pendentes, priorizados pelo *product owner*, que devem ser entregues ao cliente.
- *Backlog da Sprint*: O *Backlog da Sprint* é um conjunto de itens do *Backlog do Produto* que foram especialmente selecionados pelo *Product Owner* juntamente com o Time de Desenvolvimento para a *Sprint*.

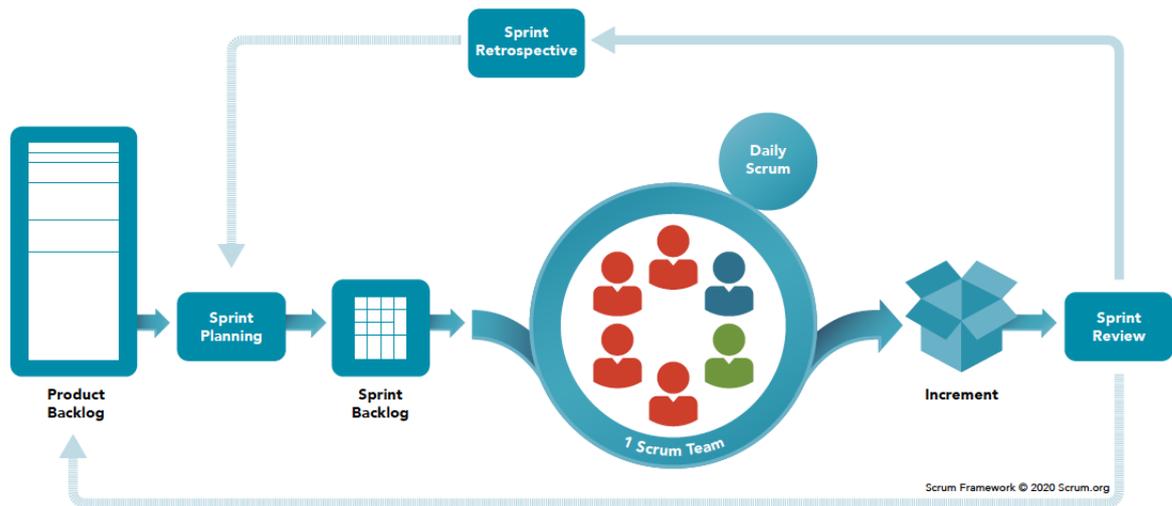
O autor deste relatório atuou no *Scrum* em um dos Times de desenvolvimento, criando novas funcionalidades ou realizando correções conforme o que foi determinado no *Backlog da Sprint* do projeto que será apresentado posteriormente no relatório.

Diferente do que é determinado pela documentação do *Scrum*, a Nexoos realizou pequenas alterações para o framework atender as necessidades da organização. Como, por exemplo, o papel do *Scrum Master* é realizado pelo *Product Owner*, logo, não são necessárias duas pessoas gerenciar um time. Ou

A seguir, na Figura 2.1, uma visualização do ciclo completo de uma *Sprint* do *Scrum*, onde de início temos o *backlog* do produto, onde temos todas as atividades do projeto. Logo

após passar pelo planejamento da *sprint* é criado o *backlog* da *sprint* e assim a *sprint* é iniciada. Enquanto a *sprint* acontece, temos a *daily* sendo realizada diariamente para os desenvolvedores reportarem suas atividades realizadas durante o dia e assim, no final, obtemos a entrega de valor ou incremento do projeto. E por fim são realizadas a revisão e retrospectiva da *sprint*.

Figura 2.1 – Funcionamento do Scrum



Fonte: Scrum.org (2021)

2.2 Tecnologias Utilizadas

Esta seção irá apresentar as tecnologias utilizadas durante a realização do estágio e desenvolvimento do projeto. Essas tecnologias abordam o que foi utilizado para a estrutura do *front-end*, as ferramentas de versionamento de código e gerenciamento de projeto.

2.2.1 Desenvolvimento *Front-end*

O *front-end* é a parte de um projeto web que lida com o que se vê e interage em um site no seu navegador. Isso inclui o design, organização do conteúdo e criação do layout. A construção desta página é realizada por meio de linguagens de programação, como HTML, CSS e JavaScript.

2.2.1.1 JavaScript

*JavaScript*² é uma linguagem de programação interpretada de alto nível. Foi criado em 1995 para tornar as páginas web mais interativas e dinâmicas. Hoje, é amplamente utilizado não apenas no desenvolvimento de páginas da Web, mas também em aplicativos de desktop, servidor, jogos e IoT (Internet das Coisas em inglês). Os recursos do *JavaScript* incluem orientação a objetos, suporte para programação funcional e processual, criação dinâmica de variáveis, alta expressividade e capacidade de manipular HTML e CSS.

JavaScript é uma marca registrada da Oracle, mas obviamente não é derivada de Java. O *JavaScript* é executado diretamente no navegador ou em um ambiente de tempo de execução como o Node.js. Isso permite que os desenvolvedores criem aplicativos da Web ricos e interativos sem a necessidade de *plug-ins* adicionais.

A figura 2.2 apresenta um exemplo de código em JavaScript onde é apresentada uma função na linguagem que teria o comportamento de atualização de um nome. Primeiramente é selecionado um arquivo e a partir disso capturado um *click* na função de atualizar nome. E assim é esperado que o usuário insira um novo nome.

Figura 2.2 – Demonstração de JavaScript

```
const para = document.querySelector('p');

para.addEventListener('click', atualizarNome);

function atualizarNome() {
  let nome = prompt('Informe um novo nome:');
  para.textContent = 'Jogador 1: ' + nome;
}
```

Fonte: Docs (2023c)

² <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>

2.2.1.2 TypeScript

*TypeScript*³ (ORG., 2023) é uma linguagem de programação *open-source* desenvolvida pela Microsoft. É uma *superset* de *JavaScript* que adiciona tipagem estática e outras funcionalidades ao *JavaScript*. *TypeScript* permite que os desenvolvedores escrevam código com menos erros e mais claramente, facilitando a manutenção e o desenvolvimento de aplicativos de grande escala.

Conforme o (ALBANO, 2023) apresenta, a principal diferença do TypeScript é ser uma linguagem tipada. Desta forma, ele apresenta os erros enquanto o desenvolvimento da aplicação é realizado, diferente do JavaScript.

A figura a seguir 2.3, apresenta um exemplo da criação de uma função utilizando o TypeScript onde ele utiliza de funções para transformar uma frase para letras maiúsculas.

Figura 2.3 – Demonstração de Typescript

```
function printName(obj: { first: string; last?: string }) {  
  // Error - might crash if 'obj.last' wasn't provided!  
  console.log(obj.last.toUpperCase());  
  
  'obj.last' is possibly 'undefined'.  
  
  if (obj.last !== undefined) {  
    // OK  
    console.log(obj.last.toUpperCase());  
  }  
  
  // A safe alternative using modern JavaScript syntax:  
  console.log(obj.last?.toUpperCase());  
}
```

Fonte: Albano (2023)

2.2.1.3 HTML (HyperText Markup Language)

HTML (*HyperText Markup Language*) (DOCS, 2023b) é a linguagem de marcação padrão da *World Wide Web*. Ele é usado para criar e estruturar conteúdo na Internet, como páginas web, documentos eletrônicos e aplicativos da web. HTML permite que os desenvolvedores especifiquem o conteúdo de uma página web e como ele deve ser exibido em um navegador da web.

³ <<https://www.typescriptlang.org/>>

O HTML é composto por elementos, sendo marcas ou tags, e atributos, que fornecem informações adicionais sobre os elementos. Por exemplo, o elemento “p” é usado para criar um parágrafo e o atributo *align* pode ser usado para alinhar o texto do parágrafo.

Além de permitir que os desenvolvedores criem conteúdo estruturado, o HTML também oferece suporte a recursos como *links*, imagens, tabelas, formulários e em sua versão mais atual, HTML5, já oferece suporte a vídeo e áudio, permitindo que os usuários interajam com o conteúdo da página.

A Figura 2.4 apresenta uma exemplificação de um trecho de código de HTML apresentando uma estrutura básica de documento HTML.

Figura 2.4 – Exemplo de HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>Minha página de teste</title>
  </head>
  <body>
    
  </body>
</html>
```

Fonte: (DOCS, 2023b)

2.2.1.4 CSS

CSS (*Cascading Style Sheets*) (DOCS, 2022) é uma linguagem de folhas de estilo usada para descrever a aparência e formatação de documentos HTML. É usado para separar o conteúdo da página web da sua aparência, permitindo que os desenvolvedores mantenham o layout e o estilo de uma página em um único arquivo CSS.

CSS permite que os desenvolvedores apliquem estilos, como cor de fundo, cor de texto, tipografia e muito mais, a elementos HTML específicos. Os estilos podem ser aplicados a elementos individuais, como parágrafos ou cabeçalhos, ou a grupos de elementos, como todas as hiperligações na página.

CSS também oferece recursos avançados, como *media queries*, que permitem que os desenvolvedores ajustem o estilo de uma página com base nas configurações do dispositivo do usuário e animações, que permitem que os desenvolvedores criem interações animadas com o

conteúdo da página. Como na figura 2.5 onde são adicionados estilos a alguns textos e títulos definidos em outro arquivo em HTML. Esses estilos são relacionados a estilizações de texto, como tamanho e alinhamento do texto.

Figura 2.5 – Exemplo de código em CSS

```
h1 {  
  font-size: 60px;  
  text-align: center;  
}  
  
p, li {  
  font-size: 16px;  
  line-height: 2;  
  letter-spacing: 1px;  
}
```

Fonte: Docs (2022)

2.2.1.5 React

React (REACTJS, 2021) é uma biblioteca JavaScript para criar interfaces de usuário. Ele permite construir aplicativos web de maneira mais eficiente e simples, ajudando a gerenciar e renderizar componentes eficientemente. React utiliza uma abordagem de programação baseada em componentes, onde a interface é dividida em pequenos componentes reutilizáveis, tornando o desenvolvimento mais fácil e escalável. Além disso, React tem uma comunidade ativa e uma ampla gama de ferramentas e recursos disponíveis, tornando-o uma escolha popular para a criação de aplicativos web modernos.

Para a realização do projeto o React foi utilizado juntamente com o *TypeScript* ao invés do *JavaScript*, para que além das suas utilidades do React de juntar o *JavaScript*, HTML e CSS, ele ainda conta com a vantagem do *TypeScript* de ser uma linguagem tipada, apontando os erros durante o desenvolvimento, excluindo a necessidade de executar o projeto para descobrir erros.

A figura a seguir 2.6 apresenta um código em React, com a criação de um cabeçalho com uma imagem e alguns textos apresentando uma aplicação React e um link que redireciona o usuário para o site oficial do React.

Figura 2.6 – Demonstração de React

```
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}
```

Fonte: Docs (2023a)

2.2.2 Versionamento de código

As ferramentas de versionamento são utilizadas para gerenciar as alterações em um código. À medida que os ambientes de desenvolvimento aceleram, os sistemas de controle de versão auxiliam as equipes a trabalharem com mais rapidez e inteligência.

2.2.2.1 Git

O Git⁴(GIT, 2023) é um sistema de controle de versão de código-fonte distribuído e sua distribuição é feita gratuitamente. Ele foi criado por Linus Torvalds, que também criou o sistema operacional Linux, lançado em 2005. Com ele, os desenvolvedores podem rastrear e

⁴ <<https://git-scm.com/>>

gerenciar modificações em seu código, reverter para versões anteriores e trabalhar em equipe em projetos. Git usa uma estrutura de árvore para rastrear as alterações no código, o que significa que cada alteração é registrada como um "*commit*" em uma linha de tempo. Esta linha de tempo é compartilhada com outros desenvolvedores e mantida em vários repositórios.

Este recurso é amplamente utilizada em todo mundo por desenvolvedores e equipes de desenvolvimento, suportada por uma gama de plataformas, incluindo Linux, Windows e MacOS.

Esta ferramenta é parte essencial no trabalho em equipe, logo, durante o projeto abordado, todo seu gerenciamento é realizado utilizando a ferramenta para o controle do que está sendo realizado e adicionado no repositório no GitHub.

2.2.2.2 GitHub

O GitHub⁵ (GITHUB, 2023) é uma plataforma de hospedagem de código-fonte baseada na web que usa o Git como sistema de controle de versão. Foi criada em 2008 por Chris Wansrath, J. Hyett, Tom Preston-Werner e Scott Chacon. Além de permitir que os desenvolvedores armazenem e gerenciem seu código em um repositório remoto, o GitHub oferece uma ampla gama de recursos adicionais para auxiliar a equipe a colaborar e desenvolver *software* de forma mais eficiente, como ferramentas de gerenciamento de projetos.

Alguns dos recursos incluem:

- *commit*: os *commits* são uma forma de separar o trabalho realizado em versões onde o usuário deixa uma mensagem caracterizando o que foi adicionado, alterado ou removido a partir desta versão.
- *Pull requests*: permite que os usuários proponham e discutam alterações no código.
- *Wiki*: permite aos usuários criar documentação para seu projeto.
- *Gist*: permite aos usuários compartilhar trechos de código rapidamente.
- *Integrações*: o GitHub possui integrações com uma ampla gama de outras ferramentas de desenvolvimento, como *TravisCI*, *Heroku* e muito mais.

⁵ <<https://docs.github.com/pt>>

O GitHub é usado por milhões de desenvolvedores e equipes de tecnologia em todo o mundo, e é uma plataforma crítica para a colaboração em projetos de *software* de código aberto e privado. É amplamente considerado a maior plataforma de hospedagem de código-fonte do mundo.

2.2.3 Gerenciamento de Projeto

O gerenciamento de projeto é necessário para ser feita uma organização do que será feito e para auxiliar o controle dos rituais e artefatos do *Framework Scrum*.

2.2.3.1 Jira

O Jira⁶ é uma plataforma de gerenciamento de projetos, desenvolvida pela Atlassian, amplamente utilizado por equipes de desenvolvimento de software e outros campos para planejar, rastrear e organizar o trabalho. Como explica (ATLASSIAN, 2023) “É a fonte única de informações de todo o ciclo de vida de desenvolvimento”. Fornece recursos para criar e gerenciar tarefas, atribuir proprietários, acompanhar o progresso, definir prazos, priorizar atividades e colaborar em equipe. Também permite criar fluxos de trabalho personalizados para atender às necessidades específicas de uma equipe ou projeto. O sistema é baseado em um modelo de *issue*, onde cada tarefa ou pendência é representada por um item individual com informações detalhadas, como descrição, *status*, prioridade, histórico de alterações e comentários. Isso permite que os membros da equipe rastreiem facilmente as tarefas, colaborem e rastreiem as atividades concluídas. E foi utilizado durante o projeto para atribuir as atividades de cada membro do time de desenvolvimento e fazer toda a organização do projeto do Novo fluxo de solicitação.

⁶ <<https://www.atlassian.com/software/jira>>

3 ATIVIDADES DESENVOLVIDAS

Este capítulo apresenta as atividades realizadas durante o estágio e o desenvolvimento de um dos projetos da Nexoos durante este período. O desenvolvimento deste projeto contribuiu com os objetivos propostos, principalmente em aplicar os conhecimentos adquiridos durante a graduação e adaptabilidade de novos desafios e dificuldades que foram aparecendo durante o período de estágio.

3.1 Treinamento e adaptação

Primeiramente, foi passado por um período de treinamento, onde foram passados alguns conceitos básicos necessários para o estagiário participar das atividades da empresa. Desta forma, no treinamento foram passados alguns cursos, como *Ruby on Rails*, *Git*, *Scrum*, *PostgreSQL* entre outras tecnologias. Outros temas de cursos passados eram os necessários para poder trabalhar em uma instituição financeira, como o de regras da LGPD (Lei Geral de Proteção de Dados Pessoais) e segurança cibernética, por serem necessários para garantir todo o cuidado com os dados dos clientes.

Após o treinamento foi passado um pequeno projeto de introdução para contextualização e testes dos conhecimentos adquiridos durante os cursos, onde o estagiário poderia desenvolver qualquer projeto utilizando as tecnologias apresentadas nas aulas. Nesse projeto foi solicitado que fosse realizada a criação de um blog, onde os usuários poderia criar *posts* utilizando de texto e imagens e permitindo que outros usuários pudessem adicionar comentários em cada uma das postagens. Por fim, depois de um mês o projeto foi avaliado pelo coordenador, a partir disso o estagiário foi alocado em uma equipe para começar o trabalho de fato.

A princípio era para ser um estágio em *full-stack*, onde se trabalha tanto com o *back-end* quanto com o *front-end* com tudo sendo desenvolvido em *Ruby on Rails*. Assim, a equipe era composta de um *Product Owner*, cinco desenvolvedores, contando com o estagiário, sendo um deles o líder técnico e todos trabalhando como *full-stack*.

A empresa possibilita a troca de função por conta do estagiário para poder encontrar a função que mais se adapta. Logo, foi proveniente, em certo momento, focar somente na parte de *front-end* e desenvolver suas habilidades no desenvolvimento de telas e um tempo depois houve a possibilidade de atuar como *Product Owner* de um *squad* apenas de desenvolvedores *front-end*, porém essa passagem durou poucos meses. Assim, para cada função, foi elaborado um PDI (Plano de Desenvolvimento Individual) baseado na nova função e seus objetivos, além

de trabalhar as *soft-skills* (Habilidades pessoais não técnicas) que fazem o auxílio no dia a dia de trabalho.

O desenvolvimento deste projeto foi realizado com uma junção de times, design, *back-end* e *font-end*. O entendimento dos requisitos foi feito pelo *Product Owner* do time e pelo time de design, que também foi responsável pelo desenho das telas e validado com os *stakeholders*. Todas essas atividades seguiram os ritos do *Framework Scrum*. Foram feitas as *plannings* a cada início de uma nova *sprint*, que durava duas semanas; as *daily*s ocorriam pela manhã, logo no início do expediente, e as *retrospectives* e *reviews* após o período determinado para a duração da *sprint*, reiniciando o ciclo.

Apesar de todo o treinamento ter sido realizado para o trabalho de *full-stack* em *Ruby on Rails*, o estagiário utilizou de conhecimentos adquiridos anteriormente em outro estágio realizado por ele em *React-Native* e com os colegas de trabalho e mentores para adquirir experiência suficiente para a realização do projeto.

Na próxima seção será apresentado um projeto realizado durante o estágio e todas as atividades realizadas são de *front-end*.

3.2 Fluxo de Solicitação

O projeto abordado foi uma refatoração de um projeto legado que estava com problemas de manutenibilidade, desta forma, foi feita uma nova versão do *front-end* e *back-end* do sistema, completamente nova utilizando como base o funcionamento da anterior.

Se trata do fluxo de solicitação de crédito para novos clientes. Esse fluxo tem como princípio o usuário inserir o valor que deseja pegar como crédito, inserir suas informações pessoais e informações da empresa, e assim ele é enviado para uma breve análise para uma rejeição ou uma pré-aprovação. Se rejeitado ainda recebe algumas sugestões de algumas outras *fintechs* parceiras onde a pessoa pode tentar conseguir o crédito.

3.2.1 Explicação do Sistema Legado

O sistema legado, era completamente feito em *Ruby on Rails*, tanto o *back-end* quanto o *front-end*. Por ser um sistema legado, seu código-fonte era muito complexo e tinha uma grande dificuldade de manter. O design já não seguia padrões atuais e sua tecnologia já estava defasada. Desta forma, optaram por uma refatoração, repensando o seguimento do fluxo e adequando aos modelos mais atuais com tecnologias adequadas.

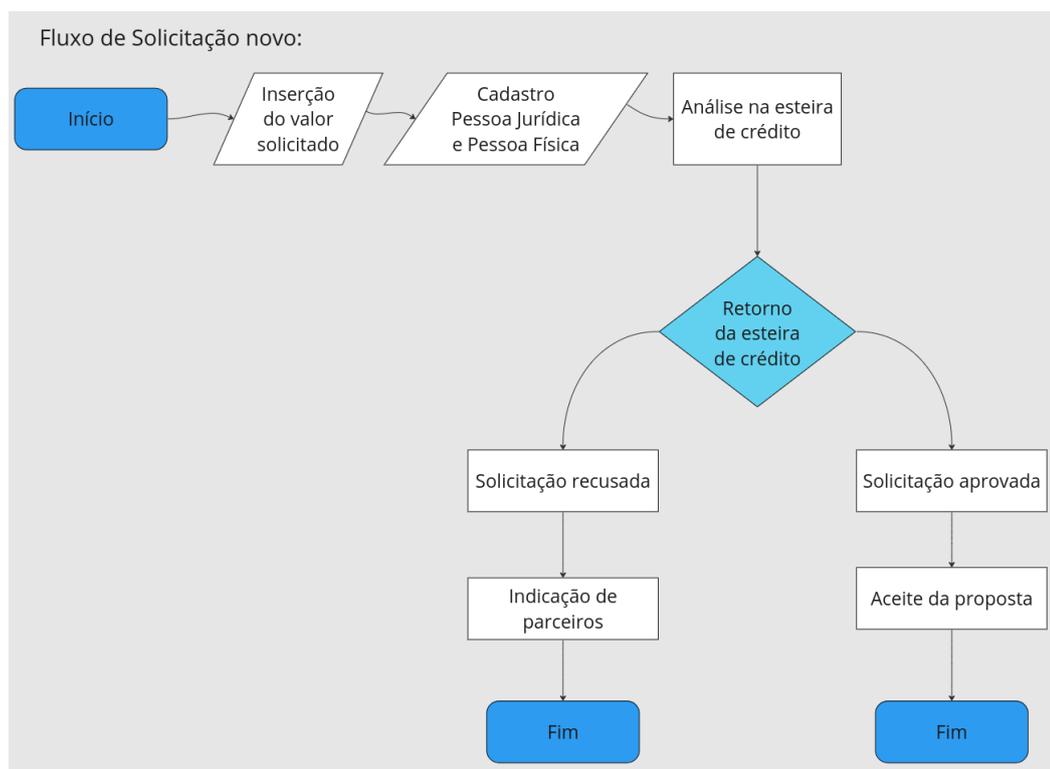
A figura 3.1 apresenta o fluxo anterior, passo-a-passo, e em comparação com o fluxo novo 3.2 pode-se verificar que além de refeito, ele também foi simplificado para que esse processo se torne mais fácil e faça com que o solicitante não acabe desistindo durante o processo.

Figura 3.1 – Fluxo de solicitação legado



Fonte: Imagem construída pelo autor

Figura 3.2 – Fluxo de solicitação refactorado



Fonte: Imagem construída pelo autor

O fluxo antigo necessitava de mais etapas e para ser realizado, fazendo que em alguns casos o solicitante desistisse da solicitação por conta da dificuldade de realizá-la. Pois era necessário inserir um dado por vez, o valor com a informação de quanto deste valor poderia ser considerado lucro e somente assim partir para a pré-análise. Este fluxo também tinha a particularidade do sistema legado, a inserção do valor solicitado era efetuada na tela principal, inicial, do site da Nexoos¹ e os dados de pessoa física só são solicitados após a análise da proposta.

O fluxo legado é exemplificado a partir da 3.3 até a 3.5 passando por cada uma das etapas até o resultado da análise, A seguir, serão apresentadas algumas telas de inserção de dados do antigo fluxo.

¹ <<https://www.nexoos.com.br/>>

Figura 3.3 – Fluxo antigo - Inserção de CNPJ

Qual o CNPJ da sua empresa?

Queremos saber algumas informações sobre sua empresa. Vamos começar com o CNPJ.

31.169.553/0001-04

Próximo

NEXOOS

f y in ig w

Sobre a Nexoos Para sua empresa Para investidores Legal

Fonte: Imagem do sistema

Figura 3.4 – Fluxo antigo - Inserção de Faturamento

Qual seu faturamento mensal?

Informe o faturamento bruto, sem descontar as despesas mensais, como folha salarial, luz, etc.

50.000,00

Deste faturamento bruto, quanto mais ou menos você considera que vai de fato para a sua conta PJ?

70%

Próximo

Fonte: Imagem do sistema

Figura 3.5 – Fluxo antigo - Tela de dados de pessoa física

Conte um pouco sobre você

Preencha os dados abaixo para efetuar o cadastro e dar continuidade na solicitação do seu financiamento.

carlos gizbert

465.589.208-08

(13) 99107-0808

.....

A senha deve conter pelo menos 8 caracteres.

Li e aceito os [termos e condições do solicitante](#) e as [políticas de privacidade](#).

Autorizo ser consultado no [Sistema de Informações de Crédito \(SCR\)](#) do Banco Central.

Enviar

Privacidade - Termos

Fonte: Imagem do sistema

3.2.2 Inserção do valor

Nesta atividade, era necessário criar uma interface simples onde haveria um campo de inserção de valor, um *steper* para mostrar ao usuário em qual etapa da solicitação ele se encontra e um botão para continuar para a próxima fase da solicitação. O campo de inserção tinha algumas regras de negócio que o valor inserido tinha que estar entre a quantia de 50 mil reais e 550 mil reais, portanto foi necessário criar essa validação para se adequar às regras. A maior dificuldade em realizar esta atividade, foi a de componentizar toda a estrutura da tela para poder ser reutilizada em outras partes do código e criar uma máscara no campo do valor para que assim que for inserindo ele possa se adequar a estrutura monetária. Como apresentada na figura 3.6 :

Figura 3.6 – Nexoos - Solicitação de crédito - inserção de valor

nexoos

Etapa 1 de 2

Quanto você precisa?

R\$ 0,00

Solicite um valor entre R\$ 10.000,00 e R\$ 550.000,00

CONTINUAR

© 2023 Nexoos do Brasil Tecnologia e Serviços Ltda.

Fonte: Imagem retirada diretamente da tela do sistema

Desta forma, após a inserção do valor a tela ficaria como apresentado na figura 3.7, com o botão habilitado após encontrar um valor válido:

Figura 3.7 – Nexoos - Solicitação de crédito - valor inserido



The screenshot shows the Nexoos logo at the top left. Below it, a progress indicator shows 'Etapa 1 de 2' with a blue line. The main question is 'Quanto você precisa?'. To the right, a light blue box contains the text 'R\$ 10.000,00'. Below this box, a smaller line of text reads 'Solicite um valor entre R\$ 10.000,00 e R\$ 550.000,00'. At the bottom right, there is a blue button labeled 'CONTINUAR'.

© 2023 Nexoos do Brasil Tecnologia e Serviços Ltda.

Fonte: Imagem retirada diretamente da tela do sistema

A imagem 3.8 apresenta um exemplo do código que solucionou o problema de ter uma máscara no campo formatando o valor de uma forma correta e adicionando a moeda antes dele. Foram utilizados códigos em Regex e algumas funções como a “*NumberFormat*” que faz a formatação para o formato brasileiro.

Figura 3.8 – Trecho do código-fonte do projeto

```
export class CurrencyMask implements FieldMaskValidation {
  constructor(readonly field: string) {}

  mask(value: string): string {
    const noMaskValue = parseFloat(value.replace(/\D/g, '')) / 100

    return new Intl.NumberFormat('pt-BR', {
      style: 'currency',
      currency: 'BRL',
    }).format(noMaskValue)
  }
}
```

Fonte: Imagem do código do sistema

3.2.3 Inserção dos Dados

Após a etapa de Inserção de valores, passamos para a tela de inserção de dados, onde o usuário irá preencher os dados solicitados para o sistema poder realizar uma rápida avaliação, tanto dele próprio quanto da empresa. Os dados solicitados são CNPJ da empresa, Faturamento mensal, nome completo do solicitante, CPF, Celular e *E-mail*, todos os dados são obrigatórios e essenciais para a realização da análise. No fim da figura 3.9, também temos os *checkboxes* dos termos de uso, política de privacidade e certificação de consulta ao BACEN (Banco central do Brasil). As dificuldades desta tela tem semelhanças com as da tela anterior em relação às validações e máscaras de campo, por exemplo, o campo *e-mail* exige que o texto tenha uma arroba (@) e um domínio (".com", ".br", ".org"), o telefone deve ter oito ou nove dígitos além do DDD (Discagem Direta à Distância) e o CPF precisa ter onze dígitos, entre outras validações. Outra regra de negócio que pertence a esta tela é a que para o usuário poder continuar a solicitação é necessário marcar todos os *checkboxes* e só assim, caso realizado, o botão "Solicitar" ficará habilitado para uso e ele poderá prosseguir para a próxima etapa.

Além disso, os componentes do rodapé, o botão "solicitar" e do *steper* foram reutilizados da tela anterior para a composição desta tela.

Figura 3.9 – Nexoos - Solicitação de crédito - Inserção de Dados

nexoos

Etapa 2 de 2

Informações sobre você e a empresa

CNPJ da Empresa *

Faturamento mensal *

Nome Completo *

CPF *

Celular *

E-mail *

Li e concordo com os [termos de uso](#).

Li e concordo com a [política de privacidade](#).

Autorizo a consulta das informações presentes no [Sistema de Informações de Crédito \(SCR\)](#), bem como da minha agenda de recebíveis em qualquer sistema gerido por registradoras autorizadas pelo Banco Central do Brasil.

SOLICITAR

VOLTAR

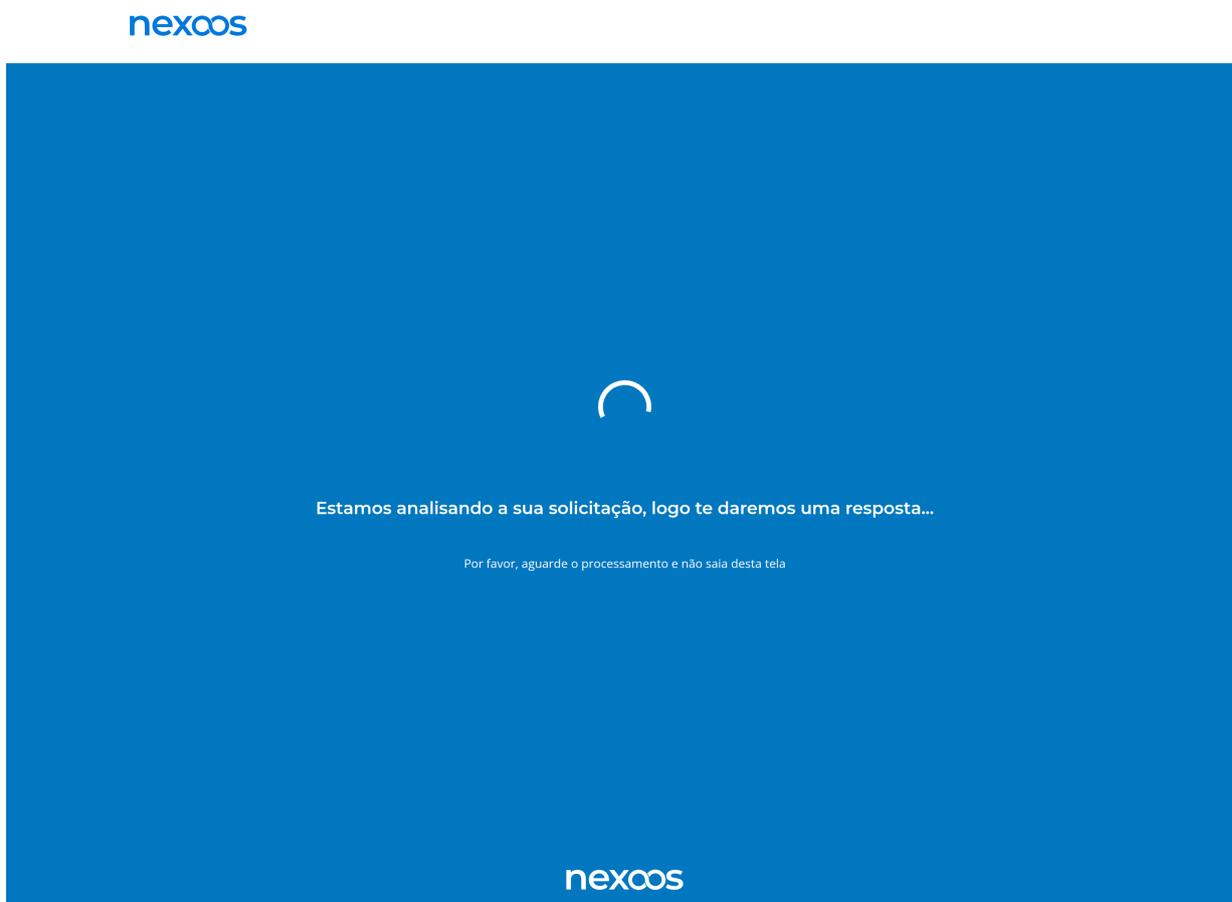
Fonte: Imagem retirada diretamente da tela do sistema

3.2.4 Tela de carregamento

O objetivo desta tela é dar um *feedback* visual para o usuário enquanto o sistema envia os dados para a análise feita no *back-end* se os dados do usuário são válidos e uma consulta do CNPJ da empresa no sistema do Banco Central, e retorna o resultado. Aparentemente a composição da tela é simples, apenas textos e o componente de carregamento que apresenta a porcentagem com um *GIF* que circula essa porcentagem.

A maior dificuldade desta tela foi realizar a criação do componente e estilização do “GIF” de carregamento, pois, ele é um componente pronto da biblioteca do MaterialUI, mas sua estilização é um pouco complicada de ser feita. E também o tempo de carregamento foi limitado em 1 minuto, para que caso o processo de análise leve mais tempo que o esperado, ele cancele e retorne um erro para o solicitante.

Figura 3.10 – Nexoos - Solicitação de crédito - Tela de carregamento



Fonte: Imagem retirada diretamente da tela do sistema

A figura 3.11 mostra a resolução para a dificuldade de limitar o tempo em que a tela ficara carregando. Na função do *useEffect*, que está delimitado em 60.000 milissegundos, logo, um minuto, caso não houver nenhum retorno do *WebSocket*, se retornar com sucesso, será redirecionada para a 3.12 e caso contrário para a 3.13.

Figura 3.11 – Trecho do código-fonte do projeto

```
useEffect(() => {  
  if (!loanContext.loan_id) {  
    return  
  }  
  const timer1 = setTimeout(() => {  
    websocket.closeChannel(`loan_${loanContext.loan_id}_status`)  
    toProposalStatusPage(appRoutes.proposalAnalysis.success)  
  }, 60000)  
}
```

Fonte: Imagem do código do sistema

3.2.5 Resultado da Análise

Após a tela de carregamento e a análise feita, o sistema redireciona o usuário para a tela apresentando o resultado, podendo ser aprovado (pré-aprovado, pois ainda teria que passar por uma validação mais afundo por analistas da empresa até enfim conseguir o crédito) ou recusado, onde receberia algumas alternativas de empresas parceiras da Nexoos que também são facilitadoras de crédito, com critérios diferentes do que foi utilizado nesta.

A dificuldade dessa tela foi transformar em componente tudo o que era necessário para compor ambas as telas, pois elas só tinham em comum os campos como título, imagem e texto auxiliar, então foram passados para ela, a partir do resultado, cada um dos componentes conforme a resposta enviada do *back-end* e assim era composta com a aparência da resposta devida. Porém, a tela de recusa ainda necessitava dos *cards* dos parceiros que precisavam entrar, então, também foi necessário criar o componente do card e enviá-los como elementos filhos da tela principal, esses *cards*.

As imagens a seguir representam os possíveis resultados que a solicitação pode ter. Após a análise o solicitante tem a possibilidade de receber as opções de aceite, recusa e erro. A tela de aceite significa que sua solicitação passou na verificação e ele poderá prosseguir com a solicitação. A tela de recusa apresenta que a solicitação não foi na análise e propõem alguns parceiros onde o solicitante pode tentar conseguir o crédito. E a tela de erro, apresenta que o solicitante já tem um cadastro feito no sistema, logo, não precisaria passar pelo fluxo de solicitação novamente.

A figura 3.12 apresenta o caso de sucesso da análise:

Figura 3.12 – Nexoos - Solicitação de crédito - Tela de aceite

nexoos



Sua solicitação foi concluída com sucesso!

Entraremos em contato após a análise da sua solicitação em até 2 dias úteis.

© 2023 Nexoos do Brasil Tecnologia e Serviços Ltda.

Fonte: Imagem retirada diretamente da tela do sistema

Como a versão de recusa da solicitação, apresentada na 3.13, onde o solicitante é avisado do resultado e parceiros da companhia, serão sugeridos:

Figura 3.13 – Nexoos - Solicitação de crédito - Tela de Recusa

nexoos



Poxa! Ainda não temos uma proposta de empréstimo para sua empresa.

Confira outras linhas de crédito que podem ajudar seu negócio

| GERU | provu | rapidoo <small>empresas ONE 7</small> |
|---|---|--|
| Empréstimo sem garantia para PF | Empréstimo sem garantia para PF | Crédito com Antecipação de Recebíveis |
| <ul style="list-style-type: none"> ✓ Valores entre R\$ 1 mil e R\$ 50 mil ✓ Taxas de juros entre de 1,6% à 5.9% a.m ✓ Custo efetivo total de 1.9% a.m. até 6.3% a.m. ✓ Parcelas em até 36 vezes ✓ Não concede crédito para negativados | <ul style="list-style-type: none"> ✓ Até R\$ 100 mil em crédito para empresas. ✓ Antecipação de boletos de até R\$ 50 mil ✓ Taxa a partir de 3,5% a.m. ✓ Empresas com protestos ou restrições | <ul style="list-style-type: none"> ✓ Valores entre R\$ 1 mil e R\$ 50 mil ✓ Taxas de juros de 1,99% a 10,95% ao mês ✓ Pagamento em até 36 meses ✓ Dinheiro na conta no mesmo dia |
| SAIBA MAIS | SAIBA MAIS | SAIBA MAIS |

© 2023 Nexoos do Brasil Tecnologia e Serviços Ltda.

Fonte: Imagem retirada diretamente da tela do sistema

E por fim, a tela de erro 3.14

Figura 3.14 – Nexoos - Solicitação de crédito - Tela de Erro



Poxa! Este email já possui uma solicitação em nosso sistema

Para continuar faça o seu login e solicite um novo crédito, ou tente novamente com um novo email

FAÇA O LOGIN E SOLICITE

VOLTAR PARA SOLICITAÇÃO

Caso tenha dúvidas entre em contato pelo contato@nexoos.com.br.

© 2023 Nexoos do Brasil Tecnologia e Serviços Ltda.

Fonte: Imagem retirada diretamente da tela do sistema

Também foi criada uma nova API em *Ruby on rails* em um novo projeto, por um dos times de *back-end* e a partir disso, foram integradas pelo time de *front-end* nas telas criadas em React. Desta forma, os projetos foram criados separadamente para a utilização de tecnologias mais adequadas, assim, facilitando a manutenção, tanto da API quanto das telas.

4 CONCLUSÃO

Este relatório de estágio abordou as atividades desenvolvidas pelo estagiário na empresa Nexoos, onde o mesmo atuou principalmente como desenvolvedor *front-end*. As principais atividades envolveram a implementação de uma nova versão de um sistema legado da empresa, em especial no que se refere ao fluxo de solicitação de crédito.

O novo Fluxo de Solicitação, até o momento que este relatório está sendo escrito, ainda não foi disponibilizado ao público, pois está passando pelo período de testes. Sendo a principal página para a empresa, precisa estar completamente funcional e livre de erros para poder ir para produção e assim ser utilizada pelo público.

O estágio foi essencial para mostrar como funciona o mercado de trabalho, como lidar com as dificuldades diárias que aparecem no dia-a-dia em uma empresa e para o estagiário adquirir uma senioridade e para poder conseguir um contrato definitivo.

As trocas de experiências que aconteceram com o estagiário, seus companheiros e mentores foram importantes para o estagiário conseguir cumprir suas atividades e adquirir a maioria do aprendizado que recebeu durante este período. O ambiente que a Nexoos proporciona para o estagiário, onde ele pode se encontrar no lugar em que mais se encaixa e todo o suporte que eles dão para que este tempo seja o mais simplificado e enriquecedor possível foram essenciais para o sucesso do trabalho.

Do ponto de vista do estagiário, poderiam ter ocorrido mais capacitações após o período de treinamento. O PDI (Plano de Desenvolvimento Pessoal) desenvolvido semestralmente pelo estagiário e seu supervisor foi de grande importância para essa capacitação, porém não tinha grande profundidade e quem tinha que procurar essas capacitações era o próprio estagiário, por ficar um bom tempo sem uma supervisão adequada, portanto nem sempre era completamente realizado. Desta forma, os aprendizados foram mais com a “mão na massa” do que realmente em capacitação.

Os conceitos e boas práticas vistas durante o trajeto acadêmico, principalmente nas disciplinas de: introdução aos Algoritmos, Estruturas de Dados, Paradigmas de Linguagens de Programação e Práticas de Programação Orientada a Objetos, serviram como uma base e contribuíram para o raciocínio lógico e capacidade de propor soluções efetivas para problemas. Além dessas, tiveram outras disciplinas que auxiliaram no processo de desenvolvimento de software e trouxeram conceitos importantes para tal, como: Engenharia de Software, Interação Humano-Computador, Sistemas Distribuídos e Programação Web. As matérias de conhecimento de ne-

gócio, como, Gerencia de Processos de *Software* e a eletiva de Planejamento empresarial, deram uma noção de negócio necessária para entender o motivo pelo qual se estava realizando cada uma das atividades.

Além disso, considerando que não só o conhecimento técnico faz um bom profissional, é fundamental desenvolver habilidades sociais, emocionais e psicológicas que foram amplamente exploradas durante o estágio. Essas habilidades incluem trabalho em equipe, lidar com prazos, entender e desenvolver as próprias características, incluindo ritmo de trabalho, áreas de interesse, comunicação. Esses pontos da vida profissional são tão importantes quanto os aspectos técnicos, pois além de fazer com que o estagiário seja um bom profissional, traz um crescimento pessoal muito grande. E definitivamente, o estágio foi uma experiência onde todos esses aspectos foram vividos.

Em suma, o estágio supervisionado proporciona muitas vantagens, como colocar em prática os conhecimentos adquiridos durante a graduação, aproximar-se do mercado de trabalho, descobrir áreas de interesse, trabalhar em equipe, aprimorar a experiência profissional e pessoal e possibilita a liberdade financeira na vida do estagiário. Considerando o exposto, o estágio relatado neste trabalho contribui significativamente para o crescimento do autor da obra sob diversos ângulos.

REFERÊNCIAS

ALBANO, C. **Conheça a diferença entre JavaScript e TypeScript**. 2023. Disponível em: <<https://coodesh.com/blog/candidates/conheca-a-diferenca-entre-javascript-e-typescript>>.

ATLASSIAN. **Jira Documentation**. 2023. Disponível em: <<https://www.atlassian.com/br/software/jira/guides/getting-started/introduction#what-is-jira-software>>.

DOCS, M. W. **CSS**. 2022. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>.

DOCS, M. W. **Começando com React**. 2023. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started>.

DOCS, M. W. **HTML: Linguagem de Marcação de Hipertexto**. 2023. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>.

DOCS, M. W. **O que é JavaScript?** 2023. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript>.

GIT. **Git Documentation**. 2023. Disponível em: <<https://git-scm.com/>>.

GITHUB. **GitHub Documentation**. 2023. Disponível em: <<https://docs.github.com/pt>>.

ORG., T. **TypeScript**. 2023. Disponível em: <<https://www.typescriptlang.org/>>.

REACTJS. **React - Uma biblioteca JavaScript para criar interfaces de usuário**. 2021. Disponível em: <<https://pt-br.reactjs.org/>>.

SCRUM.ORG. **The Scrum Guide**. 2020. Disponível em: <<https://www.scrum.org/resources/scrum-guide>>.

SCRUM.ORG. **WHAT IS SCRUM?** 2021. Disponível em: <<https://www.scrum.org/resources/what-is-scrum>>.

SOMMERVILLE, I. **Engenharia de Software**. [S.l.]: Pearson Addison-Wesley, 2007.