

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL GOIANO – CAMPUS IPORÁ
NOME DO CURSO

RELATÓRIO DE ESTÁGIO CURRICULAR

**RELATÓRIO DE ESTÁGIO BACK-END COM
JAVASCRIPT E TYPESCRIPT**

HIGOR KOAKOVSKI PEREIRA

IPORÁ, GO

2024

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL GOIANO – CAMPUS IPORÁ
NOME DO CURSO

RELATÓRIO DE ESTÁGIO BACK-END COM
JAVASCRIPT E TYPESCRIPT

HIGOR KOAKOVSKI PEREIRA

DR. CLEON XAVIER PEREIRA JÚNIOR
Orientador

Relatório de Estágio Curricular
apresentado ao Instituto Federal Goiano
– *Campus Iporá*, como requisito parcial
para conclusão do Curso de Tecnologia
em Análise e Desenvolvimento de
Sistemas.

IPORÁ, GO

Janeiro/2024



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

Identificação da Produção Técnico-Científica (assinale com X)

- ☐ Tese
- ☐ Dissertação
- ☐ Monografia – Especialização
- ☐ Artigo - Especialização
- ☒ TCC - Graduação
- ☐ Artigo Científico
- ☐ Capítulo de Livro
- ☐ Livro
- ☐ Trabalho Apresentado em Evento
- ☐ Produção técnica. Qual: _____

Nome Completo do Autor: **Higor Koakovski Pereir**

Matrícula: **2019105210430105**

Título do Trabalho: **RELATÓRIO DE ESTÁGIO BACK-END COM JAVASCRIPT E TYPESCRIPT**

Restrições de Acesso ao Documento [Preenchimento obrigatório]

Documento confidencial: ☒ Não ☐ Sim, justifique: _____

Informe a data que poderá ser disponibilizado no RIIF Goiano: **27/10/2024**.

O documento está sujeito a registro de patente? ☐ Sim ☒ Não

O documento pode vir a ser publicado como livro? ☐ Sim ☒ Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

1. O documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
2. Obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
3. Cumpru quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Iporá, 27 de novembro de 2024

Higor Koakovski Pereir

Assinado eletronicamente pelo o Autor e/ou Detentor dos Direitos Autorais

Ciente e de acordo:

Cleon Xavier Pereira Junior

Assinatura eletrônica do(a) orientador(a)

Documento assinado eletronicamente por:

- Cleon Xavier Pereira Junior, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 27/11/2024 16:22:29.
- Higor Koakovski Pereira, 2019105210430105 - Discente, em 27/11/2024 16:34:22.

Este documento foi emitido pelo SUAP em 27/11/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 655778
Código de Autenticação: df986f06e8



INSTITUTO FEDERAL GOIANO

Campus Iporá

Av. Oeste, Parque União, 350, Parque União, IPORÁ / GO, CEP 76.200-000

(64) 3674-0400



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Ata nº 92/2024 - GE-IP/CMPIPR/IFGOIANO

ATA DA SESSÃO DE JULGAMENTO DO TRABALHO DE CURSO
DE HIGOR KOAKOVSKI PEREIRA

Aos vinte e sete dias do mês de novembro de dois mil e vinte e quatro, às quinze horas e quatro minutos, por videoconferência no Google Meet, reuniu-se, em sessão pública, a banca examinadora designada na forma regimental pela Coordenação do Curso para julgar o trabalho de curso intitulado “**RELATÓRIO DE ESTÁGIO BACK-END COM JAVASCRIPT E TYPESCRIPT**”, apresentado pelo acadêmico **Higor Koakovski Pereira** como parte dos requisitos necessários à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas. O professor Cleon Xavier Pereira Junior, orientador, não pôde participar devido ao seu afastamento para pós-graduação. Dessa forma, com a aprovação do colegiado, conforme §1º do Art. 19 do Regulamento de Trabalho de Curso, conforme registrado na Ata nº 88/2024 - GE-IP/CMPIPR/IFGOIANO, a coordenação de Trabalho de Curso indicou o Professor Dr. Thamer Horbylon Nascimento para presidir a banca. Sendo assim, a banca examinadora foi presidida Professor Doutor Thamer Horbylon Nascimento, tendo como membros o Professor Doutor Marcos Alves Vieira e o Professor Mestre Wesley Flávio de Miranda. Aberta a sessão, o acadêmico expôs seu trabalho. Em seguida, foi arguido pelos membros da banca e:

(**X**) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **aprovação** do acadêmico, sem restrições.

() tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **aprovação** do acadêmico, **condicionada a satisfazer as exigências** listadas na Folha de Modificação de Trabalho de Curso anexa à presente ata, no prazo máximo de 60 dias, a contar da presente data, ficando o professor orientador responsável por atestar o cumprimento dessas exigências.

() não tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **reprovação** do acadêmico.

Conforme avaliação individual de cada membro da banca, será atribuída a nota **7,0 (sete)** para fins de registro em histórico acadêmico.

Os trabalhos foram encerrados às quinze horas e quarenta e nove minutos. Nos termos do Regulamento do Trabalho de Curso do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal Goiano – Campus Iporá, lavrou-se a presente ata que, lida e julgada conforme, segue assinada pelos membros da banca examinadora.

(Assinado Eletronicamente)
Prof. Dr. Thamer Horbylon Nascimento

(Assinado Eletronicamente)
Prof. Dr. Marcos Alves Vieira

(Assinado Eletronicamente)
Prof. Me. Wesley Flávio de Miranda

Documento assinado eletronicamente por:

- Thamer Horbylon Nascimento, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 27/11/2024 15:50:36.
- Wesley Flavio de Miranda, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 27/11/2024 15:53:07.
- Marcos Alves Vieira, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 27/11/2024 16:43:40.

Este documento foi emitido pelo SUAP em 27/11/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 655738
Código de Autenticação: 6133f96ea3



INSTITUTO FEDERAL GOIANO

Campus Iporá

Av. Oeste, Parque União, 350, Parque União, IPORA / GO, CEP 76.200-000

(64) 3674-0400

SUMÁRIO

1. INTRODUÇÃO.....	4
2. CARACTERIZAÇÃO DA EMPRESA E DADOS DO ESTÁGIO	6
3. ATIVIDADES DESENVOLVIDAS.....	7
4. CONSIDERAÇÕES FINAIS	14
5. REFERÊNCIAS BIBLIOGRÁFICAS	15

1 - INTRODUÇÃO

Este relatório de estágio tem como propósito apresentar as experiências e atividades desenvolvidas durante o estágio realizado no programa LabLuby JS, da empresa Luby Software. O estágio, com uma carga horária total de 1128 horas, foi conduzido entre junho de 2022 e fevereiro de 2023, no qual foram disponibilizados cursos e provas para o aprofundamento nos conhecimentos de desenvolvimento web. De acordo com Celestino Júnior e Patricio (2015) a *World Wide Web* (www ou web) é uma coleção de recursos vinculados, ou seja, arquivos, documentos, texto, imagens entre outros que estão conectados e disponíveis através da internet, os autores reiteram que a *Web*¹ não funciona por mágica, e é útil ter um amplo conhecimento sobre as tecnologias relacionados para compreender melhor seu funcionamento.

Segundo Warren (2022) e Estrella (2023), o início do desenvolvimento web data da década de 1990, quando Tim Berners-Lee, pesquisador da Organização Europeia de Pesquisas Nucleares (Cern), e sua equipe, desenvolveram a linguagem de marcação HTML (HyperText Markup Language), como uma forma para que seus colegas pudessem redigir e publicar com facilidade suas pesquisas na *World Wide Web*. Segundo os mesmos autores, até o momento de sua criação, a comunicação era limitada a engenheiros, pesquisadores e funcionários do governo, o sistema se tornou tão popular dentro do Cern que foi difundido entre o público externo e o sucesso foi tamanho que em 1997 havia mais de 200 mil sites.

Souto (2023) afirma que dentro do desenvolvimento web é possível dividir as áreas de atuação em duas principais vertentes distintas, porém complementares, o *front-end* e o *back-end*. O *front-end* é responsável pelo desenvolvimento da parte visual com a qual podemos interagir em uma aplicação, enquanto o *back-end* se refere ao o que fica por detrás desta aplicação, fazendo a ponte entre os dados recebidos e enviados da aplicação *front-end* para um banco de dados, impondo as devidas validações e regras de negócio. Em seus primórdios, o *front-end* era composto de HTML², CSS³ e *JavaScript*. Sendo o *JavaScript* uma linguagem de programação criada por Brendan Eich, programador e cofundador da Mozilla e

¹ Nome pelo qual é conhecida a rede mundial de computadores.

² O *HyperText Markup Language* (Linguagem de Marcação de Hipertexto) é uma linguagem de computador para estruturação de páginas na internet.

³ A *Cascading Style Sheets* (Folhas de Estilo em Cascatas) é uma linguagem de computador para definir a apresentação dos dados no navegador.

do *Firefox*⁴, a pedido da *Netscape*⁵ como uma forma de tornar sites mais dinâmicos, foi utilizado principalmente para validação de formulários.

De acordo com o site Remessa Online (2022), o *JavaScript* era assim utilizado até que em 2009 o pesquisador Ryan Dahl, percebeu a ineficiência em como a linguagem utilizava os recursos da máquina, consumindo mais processamento que o necessário. Tendo isso em mente, Dahl iniciou o desenvolvimento do que seria o *Node.js*, um ambiente de execução *JavaScript* para solucionar o travamento de servidores quando haviam diversas requisições, que é tratada no *Node.js* através de uma solução baseada em eventos de alto desempenho. Além de trazer maior aproveitamento de processamento para aplicações *JavaScript* e permitir que seja possível escrever aplicações *server-side*⁶ utilizando *JavaScript*.

O estágio foi realizado com foco no uso das tecnologias supracitadas, organizando-se numa plataforma online, própria da empresa, de cursos e provas a serem realizadas sobre o desenvolvimento web, para aprimoramento dos conhecimentos com o objetivo de participação em um projeto real da empresa.

⁴ Navegador web gratuito desenvolvido pela Mozilla Foundation.

⁵ Empresa americana de serviços de informática.

⁶ Termo utilizado para operação realizado no servidor.

2 – CARACTERIZAÇÃO DA EMPRESA E DADOS DO ESTÁGIO

2.1 – Caracterização da empresa

O estágio foi realizado na empresa Luby Tecnologia S.A.⁷, com sede na capital do estado de São Paulo, Rua Galeno de Almeida, número 188, bairro Pinheiros.

Sua principal atividade é o desenvolvimento de softwares. A organização possui personalidade jurídica privada e global sendo caracterizada ainda como de porte médio devido a quantidade de colaboradores, cerca de 300 pessoas.

Seus principais produtos são o desenvolvimento de software no geral e utilização de tendências tecnológicas como solução para diversos tipos de mercado, como elaboração de programas para gerenciamento de pessoas/recursos humanos, bem como manutenção para ambientes legados.

Boa parte dos seus colaboradores atuam em um regime remoto por todo o país, os principais meios de comunicação são aplicativos de mensagens e voz disponíveis pela internet.

2.2 – Dados do estágio

Caracterizado Programa LabLuby JS, o estágio ocorreu entre o período de 06/2022 até 02/2023, a carga horária foi dividida em 6 horas de estudo diárias, fechando 30 horas semanais, onde tive a oportunidade de aprender e exercer conhecimentos variados voltados para o desenvolvimento web. A supervisão do estágio na empresa foi realizada Alon Lubieniecki e no Instituto Federal Goiano Campus Iporá GO pelo Professor Dr. Cleon Xavier Pereira Júnior.

⁷ Luby Software - Transformação Digital. Disponível em: <https://luby.com.br/>. Acesso em: 22 de Maio de 2023.

3 – ATIVIDADES DESENVOLVIDAS

As atividades desenvolvidas durante o estágio podem-se dividir em três principais: a realização do processo seletivo, período de cursos e provas para apreensão de conhecimento e atuação como desenvolvedor dentro de uma equipe.

3.1 – Processo seletivo

Para ser aceito no programa de estágio, foi necessária a realização de uma prova e diversos testes de lógica de programação com o intuito de comprovar o conhecimento mínimo em desenvolvimento para ser aceito na turma. Na realização da prova era possível escolher o *back-end* como área, que consistia no desenvolvimento de uma API⁸ para um sistema de gerenciamento de compras de veículos, ou escolher o *front-end*, que consistia no desenvolvimento de um site para o sistema de gerenciamento de compras de veículos utilizando *React*⁹. Das opções, foi escolhido o *back-end*.

3.2 – Cursos e provas

Após aprovação no processo seletivo fui introduzido ao sistema da Luby para estagiários: o LabLuby, no qual havia diversos cursos visando aprofundar os conhecimentos na área do *back-end* e que eram seguidos de provas para aplicar o conhecimento adquirido, dentre eles os citados a seguir.

3.2.1 – JavaScript

De acordo com Flanagan (2013) *JavaScript* é a linguagem de programação da Web, sendo a linguagem de programação mais onipresente da história, desta maneira constituir uma base sólida de conhecimento sobre *JavaScript* é de extrema importância para qualquer desenvolvedor *Web*.

⁸ Application Programming Interface (Interface de Programação de Aplicação) conjunto de ferramentas, definições e protocolos que permitem que aplicações de software se comuniquem entre si.

⁹ Biblioteca *front-end JavaScript* de código aberto com foco em criar interfaces de usuário em páginas web.

O curso de *JavaScript* tem por objetivo revisar e aprofundar os conhecimentos na linguagem, retomando funcionalidades básicas e aplicando-as através de pequenas atividades, tais como DOM (*Document Object Model* ou Modelo de Objeto de Documentos), eventos, manipulação de objetos, manipulação de *arrays*, *loops*, temporizadores e integração com APIs externas. O curso contou com uma série de aulas sobre os temas supracitados e foi finalizado desenvolvendo a tela de um sistema de registro de apostas. O sistema simulava o consumo de dados de uma API externa através de um arquivo em formato *JSON*¹⁰ para renderização da tela, além da criação de funcionalidades de preenchimento automático de apostas, limpeza da cartela e carrinho nos quais as apostas eram armazenadas antes de serem submetidas.

3.2.2 – *Live code*

O *live code* foi uma atividade realizada ao vivo com o tutor da turma de *back-end*. Na atividade eram sorteadas questões de lógica de programação que deveriam ser desenvolvidas sem consulta utilizando os conhecimentos adquiridos durante o curso de *JavaScript*.

3.2.3 – *Typescript*

O *TypeScript* é uma linguagem de programação baseada em *JavaScript* desenvolvida e mantida pela Microsoft desde 2012, um dos principais motivos para o seu uso é a adição de tipagem estática ao *JavaScript*, garantindo que uma vez definido um tipo para uma variável esse tipo não pode ser alterado, evitando possíveis erros no sistema.¹¹

O curso de *TypeScript* foi introdutório e apresentou informações básicas sobre a linguagem e seus conceitos, bem como as formas de uso em variáveis, funções, classes e objetos.

Kikuc (2024) atesta que o a utilização do *TypeScript* melhora a legibilidade do código facilitando o entendimento do sistema especialmente quando é desenvolvido por uma

¹⁰ *JavaScript Object Notation* é um formato padrão baseado em texto para armazenar e trocar dados entre computadores.

¹¹ Informações obtidas da documentação oficial do *TypeScript*.

equipe; diminui a quantidade de *bugs*¹², comprovando-se em pesquisas que a utilização de *TypeScript* evita cerca de 15% dos erros mais comuns durante o desenvolvimento; saber esta linguagem abre portas para o mercado de trabalho visto que é uma habilidade em demanda. A partir destas características a empresa elencou o curso como indispensável aos estagiários.

3.2.4 – Introdução a API REST

Campos (2013) afirma que um dos principais desafios no mercado de TI¹³ é prover acesso a um número maior de clientes através de aplicações desenvolvidas, e visando solucionar essa barreira surge o conceito de API, com a capacidade de disponibilizar uma interface bem definida dos dados disponíveis em uma aplicação na internet.

Contudo, é necessária uma padronização para a construção de APIs, a mais utilizado é o padrão REST (*Representational State Transfer* ou Transferência Representacional de Estado) proposto por Roy Fielding em 2000, como descrito por Massé (2011).

O curso foi de suma importância visto que ter o conhecimento para estruturação de uma API REST é fundamental para um desenvolvedor *Web*, tanto para a construção de uma boa API enquanto programador *back-end* quanto para o seu consumo enquanto programador *front-end*.

3.2.5 – Containerização com Docker

De acordo com Mouat (2015), a containerização, isto é, a encapsulação de uma aplicação e suas dependências, tem mudado a forma como se desenvolve, distribui e executa-se código. Esse conceito garante que se possa construir código localmente sabendo que ele será executado da mesma maneira independente do ambiente onde o sistema será hospedado. Segundo o mesmo autor, contêineres são uma tecnologia antiga utilizada por sistema UNIX que vem sendo aprimorada desde sua concepção, em 2013 a criação do *Docker* simplificou e popularizou a containerização. O *Docker* é composto por dois principais componentes, o *Docker Engine* que é responsável por prover uma interface rápida e conveniente para criação

¹² Erro no código de um programa que faz com que ele se comporte de forma inesperada ou falhe ao executar uma função.

¹³ Tecnologia da informação.

e execução de contêineres, e o *Docker Hub* que consistem em um sistema em nuvem para disponibilização de contêineres.

Desse modo, a empresa disponibilizou o curso de Docker, apresentando conceito e como escrever e executar contêineres como parte da formação.

3.2.6 – Frameworks com AdonisJS

Segundo a documentação oficial¹⁴ *AdonisJS* é um *framework*¹⁵ Web baseado em *Typescript* para desenvolvimento de aplicativos webs e APIs com ferramentas modernas, suporte para testes, e uma biblioteca oficial de pacotes.

Tendo passados pelos módulos de *JavaScript*, *TypeScript* e API REST, foi proporcionado um curso desenvolvido pela empresa acerca de *AdonisJS* como introdução a frameworks, comumente utilizados para facilitar a construção de aplicativos visto que proporcionam diversas ferramentas prontas para desenvolver uma aplicação. Como exemplo, após o curso foi realizada uma prova para a construção de uma API para o um sistema de apostas, utilizando das ferramentas disponíveis no *AdonisJS* para o desenvolvimento da mesma, como a criação de *migrations*¹⁶, autenticação, validadores de dados e rotas para interagir com o sistema.

3.2.7 – Mensageria com Apache Kafka

Segundo Korab (2017) a mensageria envolve um acordo entre dois sistemas quanto ao formato das mensagens trocadas entre eles, e a partir disso eles se tornam desacoplados um do outro através de uma interface, permitindo que sistemas desenvolvidos de maneira diferente possam se comunicar entre si. O mesmo autor explica que um sistema de mensageria comumente envolve um terceiro responsável por desacoplar o emissor do receptor ou receptores.

¹⁴ Disponível em: <https://adonisjs.com>.

¹⁵ Conjunto de ferramentas, bibliotecas, estruturas e diretrizes que facilitam o desenvolvimento de software, aplicativos e sistemas.

¹⁶ Migração de dados que consiste na transferência de informações digitais de um sistema de armazenamento para outro.

Um exemplo de sistema para mensageria é o *Apache Kafka*. Foi disponibilizado um *workshop* realizado por um membro da Luby S.A., apresentado seu sistema usado para coletar, processar e armazenar dados de eventos de streaming, o *Apache Kafka* utiliza *producers*, que são produtores de mensagens e *consumers*, os consumidores dessas mensagens. O *Apache Kafka* é popularmente utilizado em microsserviços para notificar ações dentro do sistema para seus diversos serviços, como quando um novo usuário é cadastrado ou uma venda é realizada no sistema.

3.2.8 – Microsserviços

Microsserviços é um estilo de arquitetura, no qual um grande e complexo aplicativo de software é dividido em um ou mais serviços, permitindo que esses serviços sejam implementados sem uma acoplagem forte entre eles visto que o objetivo de cada serviço é realizar uma única tarefa de maneira satisfatória (Daya, *et al.*, 2015).

O curso de microsserviços apresentou seu conceito, vantagens e desvantagens, comparação com sistemas monolíticos¹⁷ e como aplicar em um sistema essa arquitetura. Após o curso foi realizada uma prova cujo objetivo era aplicar no sistema desenvolvido durante a prova de *AdonisJS* um microsserviço de envio de e-mails para novos usuários cadastrados e novas apostas realizadas.

3.2.8 – API com *NestJS* e *GraphQL*

Segundo a documentação oficial¹⁸ *NestJS* é um *framework* para construção de aplicações *server-side* de maneira eficiente e escalável com *Node.js*. Já o *GraphQL*, de acordo com seu site oficial¹⁹, é uma linguagem de consulta que disponibiliza de forma clara os dados presentes em uma API e permite ao cliente solicitar exatamente os dados que precisa, evitando *Over-Fetching*²⁰ e *Under-Fetching*²¹ de informação.

¹⁷ Sistema monolítico é um modelo de desenvolvimento de software que reúne todos os componentes de uma aplicação em um único programa.

¹⁸ Disponível em: <https://nestjs.com>.

¹⁹ Disponível em: <https://graphql.org>.

²⁰ Cenário em que o cliente recebe mais informação que necessita, aumentando o consumo de dados e o tempo de espera pelos dados solicitados.

O curso de *NestJS* com *GraphQL* abordou a criação de API com o *framework* aplicando *GraphQL*. A prova consistiu na criação de dois microsserviços, um de cadastro de usuários e outro de aprovação de créditos para o usuário cadastrado utilizando a integração do *NestJS* com *GraphQL* permitindo uma aplicação prática dos conhecimentos adquiridos.

3.2.8 – *Clean architecture* e *Serverless*

Este curso foi dividido em duas partes, a primeira abordando e aplicando design de software *clean architecture*, uma proposta para desenvolver softwares de maneira limpa abstraindo lógica interna de sistemas externos, mantendo as partes do mesmo facilmente alteráveis independentemente de algum fator externo como um serviço ou uma biblioteca (Martin, 2017), seguindo os princípios SOLID²² e uso de testes para o desenvolvimento de um sistema de enquetes.

A segunda parte do curso tratou do *framework Serverless*, utilizado para facilitar a implementação de aplicações em sistemas que utilizam servidores nuvem baseados em eventos, evitando que seja necessária uma máquina virtual 100% dedicada para uma aplicação.

3.2.9 – *Live code* final

A seção de cursos e provas do estágio foi finalizado com mais um *live code* com o tutor da turma de *back-end*. Este teve duração de duas horas e foi realizada um CRUD²³ de usuários utilizando os conhecimentos adquiridos no curso de *clean architecture*.

²¹ Cenário em que o cliente não recebe informação suficiente para complete uma tarefa.

²² SOLID são cinco princípios da programação orientada a objetos que facilitam o desenvolvimento de softwares, tornando-os fáceis de manter. São eles: *Single Responsibility Principle* (Princípio da responsabilidade única), *Open-Closed Principle* (Princípio Aberto-Fechado), *Liskov Substitution Principle* (Princípio da substituição de Liskov), *Interface Segregation Principle* (Princípio da Segregação da Interface) e *Dependency Inversion Principle* (Princípio da inversão da dependência).

²³ CRUD é a sigla para as quatro operações básicas de armazenamento de dados: Create (criar), Read (ler), Update (atualizar) e Delete (deletar).

3.3 – Atuação em equipe

Após aprovação nos cursos fui incluído em uma equipe de desenvolvedores que atuavam em projetos reais dentro da empresa, essa parte do estágio durou três meses. Nele apliquei todos os conhecimentos adquiridos durante os cursos, aprendi como funcionava o desenvolvimento com equipe na prática, tive reuniões diárias de alinhamentos e seguindo padrão SCRUM ²⁴tivemos reuniões de revisão e planejamento de Sprints.

²⁴ SCRUM é um *framework* de gestão de projetos que utiliza metodologias ágeis.

4 – CONSIDERAÇÕES FINAIS

Ter uma boa base em conceitos, arquiteturas e novas tecnologias é essencial para ser um bom desenvolvedor de software. Através dos cursos e das orientações e apoio fornecidos pelo tutor do LabLuby e os cursos bem atualizados e de boa qualidade fornecidos durante o estágio pude construir uma base sólida de conhecimentos para poder atuar profissionalmente na área da programação. Por meio do período trabalhando em equipe pude adquirir experiência quanto ao desenvolvimento de software em grupos, as plataformas, padrões e organizações utilizadas para estabelecer um cronograma de desenvolvimento e realizar a entrega de software. Concluo que no decorrer do estágio adquiri uma nova concepção da programação, tanto como parte da minha vida, quanto em concepção profissional. Nesse sentido, o estágio me abriu novas portas e me permitiu pensar na carreira de programador como uma profissão, de tal maneira que fui contratado como desenvolvedor trainee na Luby após a conclusão das etapas.

5 – REFERÊNCIAS BIBLIOGRÁFICAS

LIVROS

CELESTINO JÚNIOR, Joaquim; PATRÍCIO, Robério Gomes. **Desenvolvimento para web**. 2ª ed. Fortaleza: EdUECE, 2015.

DAYA, Shahir *et al.* **Microservices from Theory to Practice**: Creating Applications in IBM Bluemix Using the Microservices Approach. Vervanté, 2015.

FLANAGAN, David. **JavaScript**: O guia definitivo. Porto Alegre: Bookman, 2013.

KORAB, Jakub. **Understanding Message Brokers**: Learn the Mechanics of Messaging through ActiveMQ and Kafka. Sebastopol: O'Reilly Media, 2017.

MARTIN, Robert C. **Clean Architecture**: A Craftsman's Guide to Software Structure and Design. Hoboken: Prentice Hall, 2018.

MASSÉ, Mark. **Rest API**: Design Rulebook. Sebastopol: O'Reilly Media, 2011.

LIVROS EM MEIO ELETRÔNICO

KUKIC, Goran. **Typescript fundamentals**. Novi Sad: 2024. Disponível em: <https://github.com/GoranKukic/typescript-fundamentals>. Acesso em: dezembro de 2024

MOUAT, Adrian. **Using Docker**. Sebastopol: O'Reilly Media, 2015. Disponível em: <http://178.140.10.58:8083/read/5175/pdf>. Acesso em: janeiro de 2024

DISSERTAÇÕES, TESES, TRABALHOS DE CONCLUSÃO

CAMPOS, Marcos Vinícius Souza. **Desenvolvimento de APIs baseadas em REST para integração e construção de aplicações**. 2013. Trabalho de Conclusão de Curso (Bacharelado em Análise de Sistemas e Tecnologia da Informação) – Faculdade de Tecnologia de Americana, Americana, 2013.

HOME PAGES

ADONISJS. **AdonisJS Framework**. Disponível em: <https://adonisjs.com/>. Acesso em: dezembro de 2024.

ANGELO LUZ. Criando a aplicação do zero | NestJS com GraphQL #2. Youtube, 2020. Vídeo (1h 15min). Disponível em: <https://www.youtube.com/watch?v=tVQwV-c19RU>.

ANGELO LUZ. Entenda o GraphQL | NestJS com GraphQL #1. Youtube, 2020. Vídeo (31 min). Disponível em: <https://www.youtube.com/watch?v=Yr4GszcuLHI>.

A HISTÓRIA por trás do www: conheça Tim Berners-Lee. **Warren Magazine**, 2022. Disponível em: <https://warren.com.br/magazine/tim-berners-lee>. Acesso em: 03 de Abril de 2023.

DACIUK, Fernando. Curso Java Script Ninja. **Udemy**, 2021. Disponível em: <https://www.udemy.com/course/curso-javascript-ninja>. Acesso em: Junho de 2022.

ESTRELLA, Carlos. O Que é a Rede Mundial de Computadores. **Hostinger Tutoriais**, 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/rede-mundial-de-computadores>. Acesso em: 03 de Abril de 2023.

GRAPHQL. **GraphQL - A Query Language for APIs**. Disponível em: <https://graphql.org/>. Acesso em: dezembro de 2024.

LABLUBY | Cursos. Disponível em: <https://lab.luby.com.br/>. Acesso em: Maio 2022. RODRIGO BRANAS. Introdução ao REST // REST API #1. Youtube, 2020. Vídeo (28 min). Disponível em: <https://www.youtube.com/watch?v=yZifRUvxdAk&t>.

LUBY. Integração de Microsserviços com Kafka - Henrique Lopes. Youtube, 2021. Vídeo (53 min). Disponível em: <https://www.youtube.com/watch?v=DhRRBP6eaOk>.

LUBY. Testes automatizados com JAPA no AdonisJS. Youtube, 2022. Vídeo (39 min). Disponível em: <https://www.youtube.com/watch?v=lkHfgX6mxLw>.

LUBY SOFTWARE - Transformação Digital. Disponível em: <https://luby.com.br/>. Acesso em: 22 de Maio de 2023.

MANGUINHO, Rodrigo. NodeJs, Typescript, TDD, DDD, Clean Architecture e SOLID. **Udemy**, 2021. Disponível em: <https://www.udemy.com/course/tdd-com-mango>. Acesso em: agosto de 2022.

NESTJS. **NestJS - A Progressive Node.js Framework**. Disponível em: <https://nestjs.com/>. Acesso em: dezembro de 2024.

Node.js: descubra tudo sobre a linguagem e suas aplicações. **Remessa Online**, 2022. Disponível em: <https://www.remessaonline.com.br/blog/node-js-descubra-tudo-sobre-a-linguagem-e-suas-aplicacoes>. Acesso em: 04 de abril de 2023.

ROVEDA, Ugo. Desenvolvimento Web: O que é e como ser um desenvolvedor web. **Kenzie**, 2020. Disponível em: <https://kenzie.com.br/blog/desenvolvimento-web>. Acesso em: 03 de abril de 2023.

SOUTO, Mario. Front-end, Back-end e Full Stack. **Alura**, 2023. Disponível em: <https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>. Acesso em: 04 de abril de 2023.

TYPESCRIPT. **TypeScript Documentation.** Disponível em:
<https://www.typescriptlang.org/docs/>. Acesso em: dezembro de 2024.