



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS MONTEIRO
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ARTHUR PEREIRA DA SILVA

**RELATO DE EXPERIÊNCIA PROFISSIONAL NO PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE NA EMPRESA IGNIS**

MONTEIRO

2024

ARTHUR PEREIRA DA SILVA

**RELATO DE EXPERIÊNCIA PROFISSIONAL NO PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE NA EMPRESA IGNIS**

Relatório de Estágio apresentado à
Coordenação de Estágio do Instituto
Federal de Educação, Ciência e
Tecnologia da Paraíba, Campus Monteiro,
como requisito parcial para conclusão do
Curso Superior de Tecnologia em Análise
e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Tiago Brasileiro
Araújo.

MONTEIRO

2024

Dados Internacionais de Catalogação na Publicação – CIP
Bibliotecária responsável Porcina Formiga dos Santos Salgado CRB15/204
IFPB Campus Monteiro.

S586r Silva, Arthur Pereira da.

Relato de experiência profissional no processo de desenvolvimento de software na Empresa IGNIS / Arthur Pereira da Silva – Monteiro-PB. 2024.

38fls. : il.

Relatório de estágio (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB campus, Monteiro.

Orientador: Prof. Dr. Tiago Brasileiro Araújo.

1. Arquitetura software 2. Sistemas - desenvolvimento 3. Javascript - linguagem programação 4. Empresa IGNIS – Estágio I. Título .

CDU 004.273

ARTHUR PEREIRA DA SILVA

**RELATO DE EXPERIÊNCIA PROFISSIONAL NO PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE NA EMPRESA IGNIS**

Relatório de Estágio apresentado à
Coordenação de Estágio do Instituto
Federal de Educação, Ciência e
Tecnologia da Paraíba, Campus Monteiro,
como requisito parcial para conclusão do
Curso Superior de Tecnologia em Análise
e Desenvolvimento de Sistemas.

Aprovado em 19 de Setembro de 2024.

BANCA EXAMINADORA

Documento assinado digitalmente



TIAGO BRASILEIRO ARAUJO

Data: 01/10/2024 08:47:15-0300

Verifique em <https://validar.iti.gov.br>

Prof. Dr. Tiago Brasileiro Araújo (Orientador - IFPB)

Prof. Esp. Wagner de Oliveira Santos (Examinador - IFPB)

Documento assinado digitalmente



GILVONALDO ALVES DA SILVA CAVALCANTI

Data: 09/10/2024 09:36:35-0300

Verifique em <https://validar.iti.gov.br>

Prof. Esp. Gilvonaldo Alves da Silva Cavalcanti (Examinador - IFPB)

RESUMO

Neste documento será relatado a experiência profissional que o discente obteve enquanto estagiário como desenvolvedor de software na empresa Ignis. Sendo esse estágio supervisionado na modalidade *home office*, o discente participou de cerimônias do *scrum* (metodologia ágil no processo de desenvolvimento de software) e ficou responsável por realizar atividades práticas como web scraping, implementação de recursos e correções de bugs na plataforma principal e na administrativa, além de realizar a migração e a documentação da plataforma administrativa para uma versão nova. Nessas atividades foi utilizado linguagens de programação que são C#, Javascript e Typescript, além dos frameworks: Hibernate, .Net Framework e React, além da linguagem SQL para as consultas no banco e também de outras bibliotecas cruciais que serão relatadas com mais profundidade no relatório. Outro ponto também são as ferramentas para o desenvolvimento como DBeaver, Visual Studio Code, Microsoft Visual Studio e GitKraken e as de comunicação como Teams, Outlook e o Azure DevOps. Todo essa experiência vivenciada pelo discente durou um período de 1 ano com a carga horária de 30 horas semanais e flexíveis, além de ter sido supervisionado por um desenvolvedor júnior, Leonardo Hummel de Sousa, durante todo o período. O relatório servirá também para mostrar como essa experiência profissional foi importante para o discente fortificar toda a fundamentação teórica que o discente obteve durante o curso de Análise e Desenvolvimento de Sistemas, na prática na Ignis.

Palavras-chave: javascript; arquitetura de software; *scrum*; desenvolvimento de sistemas.

ABSTRACT

In this document, the student's professional experience gained as a software developer intern at Ignis will be reported. This internship was supervised remotely, with student participation in scrum ceremonies (an agile methodology in software development processes) and being responsible for practical activities such as web scraping, implementing features, and fixing bugs in both the main and administrative platforms. Additionally, the student conducted the migration and documentation of the administrative platform to a new version. In these activities were used programming languages such as C#, Javascript and Typescript, along with frameworks including Hibernate, .NET Framework and React. SQL was also employed for database queries and also crucial libraries, which will be detailed further in the document. Another point is also the development tools such as DBeaver, Visual Studio Code, Microsoft Visual Studio, GitKraken, as well as communication tools like Teams, Outlook and Azure DevOps. All this experience lived by the student lasted for a period of 1 year with a workload of 30 flexible hours per week, supervised by a junior developer, Leonardo Hummel de Souza, throughout the period. The report will also demonstrate how this professional experience was vital in solidifying the theoretical foundation acquired during the student's course in Systems Analysis and Development, in practice at Ignis.

Key-words: javascript; software architecture; scrum; systems development.

LISTA DE ILUSTRAÇÕES

Figura 1 - Logomarca da Ignis	11
Figura 2 - Abstração da organização da empresa.	13
Figura 3 - Página inicial da Ignis.	14
Figura 4 - Logomarca da Alvarez & Marsal.	15
Figura 5 - Card no Devops.	16
Figura 6 - Tela inicial do Azure Devops.	19
Figura 7 - Exemplo de Fluxo do GitFlow no GitKraken.	21
Figura 8 - Interface do DBeaver	21
Figura 9 - Interface principal do Storybook	23
Figura 10 - Notebook disponibilizado pela empresa	24
Figura 11 - Exemplo de script SQL para criar a tabela e a chave no NHibernate	28
Figura 12 - Exemplo de mapeamento de classe pelo <i>Entity</i>	29
Figura 13 - Exemplo de organização de pastas de documentação	32

LISTA DE ABREVIATURAS

ADS	<i>Análise e Desenvolvimento de Sistemas</i>
DEV	<i>Desenvolvedor</i>
HTML	<i>Linguagem de Marcação de Hipertexto</i>
IFPB	<i>Instituto Federal da Paraíba</i>
PJ	<i>Pessoa Jurídica</i>
RH	<i>Recursos Humanos</i>
SQL	<i>Structured Query Language</i>

SUMÁRIO

1 INTRODUÇÃO.....	8
2 OBJETIVOS.....	10
2.1 Objetivo geral.....	10
2.2 Objetivos específicos.....	10
3 A EMPRESA.....	11
3.1 Identificação da empresa.....	11
3.1.1 Dados referentes ao estágio.....	11
3.1.2 Perfil da empresa.....	13
3.2 Processos e ambiente.....	15
4 FUNDAMENTAÇÃO TEÓRICA.....	17
4.1 Metodologia Ágil (scrum).....	17
4.2 Azure Devops.....	18
4.3 Teams.....	19
4.4 GitKraken.....	19
4.6 GitFlow.....	20
4.7 DBeaver.....	21
4.8 Ferramentas para o desenvolvimento.....	22
4.8.1 Microsoft Visual Studio.....	22
4.8.2 Visual Studio Code.....	22
4.9 Storybook.....	22
5 RELATO DE ESTÁGIO.....	24
5.1 Atividades desenvolvidas no estágio supervisionado.....	24
5.1.1 Repositórios.....	25
5.1.1.1 GeoPortal.....	25
5.1.1.2 IgnisApi.....	25
5.1.1.3 Ignis-FrontEnd.....	25
5.1.1.4 IgnisAdmin-FrontEnd.....	26
5.1.2 Crawlers.....	26
5.1.2.1 Crawlers pelo NHibernate no GeoPortal.....	27
5.1.2.2 Crawlers pelo Entity Framework no IgnisApi.....	29
5.1.3 Processo de desenvolvimento.....	30
5.1.4 Backend da aplicação principal.....	30
5.1.5 Frontend da aplicação principal.....	31
5.1.6 Backend da aplicação administrativa.....	31
5.1.7 Frontend da aplicação administrativa.....	31
6 CONSIDERAÇÕES FINAIS.....	33
REFERÊNCIAS.....	35
ANEXO A - Detalhes do desafio.....	37

1 INTRODUÇÃO

Durante a sua formação no curso Análise e Desenvolvimento de Sistemas (ADS) no Instituto Federal da Paraíba (IFPB), o discente, Arthur Pereira da Silva, foi apresentado a diversos conceitos voltados a área de programação, como técnicas avançadas na projeção e implementação de soluções de softwares, documentação de sistemas, participação em equipes utilizando metodologias de desenvolvimento (em priori, as metodologias ágeis), testes, dentre outras. Sendo esses assuntos de grande valia para um profissional na área. Entretanto, mesmo sendo notório a importância dessas noções oferecidas pela instituição para o discente, ainda é nítido a importância de experienciar a realidade no mercado de trabalho como desenvolvedor de software.

No livro “Codificador Limpo”, o autor Robert C. Martin, conclui no capítulo 14 afirmando: “[...] A escola pode ensinar a teoria da programação. Mas ela não pode ensinar (e não ensina) a disciplina, prática e habilidade de ser um artesão. Essas são coisas adquiridas ao longo dos anos de ensino e supervisão pessoais [...]”. Conforme o pressuposto anterior, a graduação no curso não carrega consigo uma experiência real na área, sendo esse um fator crucial para o aprimoramento e desenvolvimento de um profissional.

O estágio supervisionado que pode ser entendido como sendo: “um meio que o aluno tem de entrar na área de trabalho que tanto almeja, visitando e participando na prática, de assuntos que ele normalmente acompanhava somente em sala de aula.” - (Priscila, 2021). Seguindo essa lógica, o discente buscou a sua inserção em um estágio, sendo a empresa que proporcionou essa oportunidade de aprimoramento ao discente a Ignis.

A empresa Ignis Inteligência em Energia e Gás Natural LTDA, que está localizada na cidade do Rio de Janeiro–RJ, é uma empresa no setor de energia e gás natural que disponibiliza uma plataforma de gestão estratégica que auxilia empresas no segmento de óleo, gás e outros combustíveis, na tomada de decisões.

Na empresa, o discente atuou como estagiário de desenvolvimento com proficiência em *frontend* e *backend* juntamente com a equipe de desenvolvedores,

possuindo diversas atividades como o desenvolvimento de *features*, correção de bugs e na introdução do processo de documentação do *frontend* na empresa.

Outro ponto a comentar, é que o discente teve uma grande participação no processo da realização de web scraping - raspagem, em tradução livre - também conhecido como extração de dados da web, é o nome dado ao processo de coleta de dados estruturados da web de maneira automatizada (Branco, 2021).

O relatório contará com seis módulos, sendo o primeiro a introdução que é essa seção atual, o segundo sendo os objetivos tanto gerais e específicos desse documento. O terceiro é aonde haverá uma apresentação prévia da empresa como seu local e perfil organizacional, o quarto contará sobre as tecnologias utilizadas e como foram aplicadas durante o estágio, o quinto é sobre as experiências e atividades desenvolvidas pelo discente ao longo do estágio com as tecnologias e metodologias utilizadas. Por fim, a sexta seção sendo a conclusão de toda a experiência vivenciada do discente atuando como desenvolvedor na empresa Ignis e sobre todo esse contato.

2 OBJETIVOS

2.1 Objetivo geral

Esse relatório visa relatar e descrever as experiências e atividades do discente durante todo o processo como estagiário em desenvolvimento de software na empresa Ignis.

2.2 Objetivos específicos

- Introduzir brevemente a empresa Ignis, onde o estágio ocorreu, desde a sua localidade até as suas metodologias;
- Demonstrar as tarefas específicas que o discente teve na empresa, além de explicar as tecnologias utilizadas;
- Refletir sobre a atuação do discente na empresa em ambos os projetos da Ignis, a plataforma principal e a administrativa;
- Descrever a experiência do discente no que diz respeito a aplicação dos seus conhecimentos adquiridos durante o curso nas atividades do estágio.

3 A EMPRESA

Neste tópico será abordado as informações da empresa Ignis, e também as atividades desenvolvidas pelo discente em ordem temporal ao longo do estágio supervisionado.

3.1 Identificação da empresa

O estágio supervisionado aconteceu na empresa Ignis que é uma plataforma de planejamento estratégico para empresas do mercado de óleo e gás. Sua criação ocorreu em 2018 em parceria com a Geopost Energy. Empresa localizada na Avenida Augusto Severo, 804, Rio de Janeiro–RJ, 22210-010. Sendo possível observar a logomarca da empresa Ignis na Figura 1.

Figura 1 - Logomarca da Ignis.



Fonte: LinkedIn.

3.1.1 Dados referentes ao estágio

O discente teve seu primeiro contato com a empresa por meio de uma vaga disponibilizada no grupo de e-mail acadêmico do IFPB-Campus Monteiro, no dia 23 de abril de 2023. Os requisitos para o estágio eram: ser um estudante de tecnologia da informação ou área correlatada, conhecimentos nas linguagens de programação Javascript e C# (sendo o diferencial possuir conhecimentos em .Net Framework,

CSS e Jquery e React), experiência com SQL, inglês intermediário, possuir uma boa lógica de programação e disponibilidade para estagiar 1 ano na empresa.

Para a sua aprovação como estagiário na empresa, o discente foi submetido a um processo interno de seleção, a qual consistia primeiramente na realização de um sistema de um torneio de futebol com Javascript, HTML e CSS, onde os requisitos estão presentes no ANEXO A. Seguindo desse desafio, foi realizada uma entrevista individual com os desenvolvedores júnior da empresa a respeito de conhecimentos e perspectivas dentro na empresa.

Posteriormente, o discente foi selecionado para um período de estágio na Ignis que se iniciou no dia 5 de junho de 2023 até 5 de junho de 2024, durando ao todo 1 ano de estágio supervisionado com uma carga horária semanal de 30 horas semanais, sendo todo período de estágio na modalidade de *home office* (trabalho remoto).

Durante as primeiras semanas com a empresa, o discente iniciou o processo de treinamento interno que visava o familiarizar com as equipes presentes, seus processos e com a metodologia seguida pela empresa. O processo se baseava nas seguintes etapas:

- Reunião geral com os membros da empresa dos setores de desenvolvimento, operação e o RH.
- Explicações sobre os processos de desenvolvimento da empresa com o supervisor.
- Apresentações das equipes de desenvolvimento sobre a arquitetura da plataforma principal, *crawlers* (robôs automatizados utilizados para a coleta dos dados).
- Apresentação do *UX designer* da empresa sobre normas e regras seguidas para o desenvolvimento, além de conceitos aplicados para a plataforma principal.
- Configuração dos ambientes de desenvolvimento e comunicação no computador da empresa.
- *Onboarding* na plataforma principal da empresa sobre a mentoria do setor de operação.

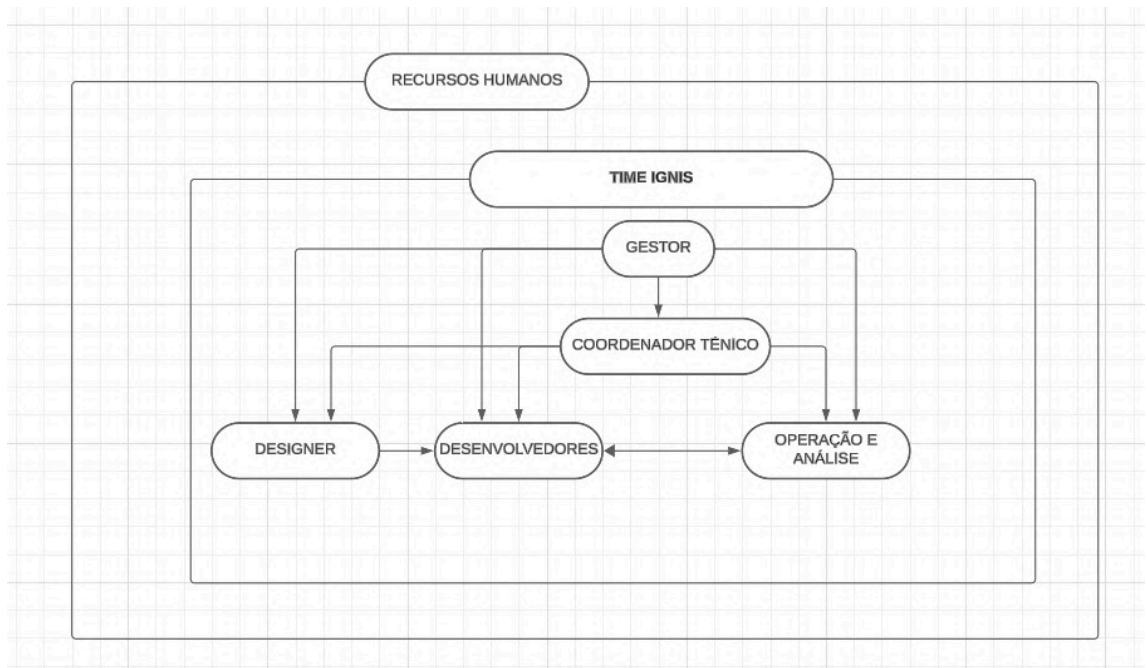
- Orientação sobre o processo de criação de *crawlers* e o *frontend* e *backend* da plataforma principal.

3.1.2 Perfil da empresa

Há 6 anos, a Ignis vem sendo, conforme o CEO da empresa, uma “plataforma de planejamento estratégico para empresas atuantes no mercado de gás natural no Brasil, cuja criação ocorreu em 2018 em parceria com a Geopost Energy, combinando mais de 20 anos de *expertise* no mercado de O&G” - (Rivaldo, s.d.).

Na Ignis, os colaboradores da empresa são divididos em dois times, operação e análise e o time de desenvolvimento, sendo a ponte entre esses times um coordenador técnico e um gestor geral, além do *UX designer*. Cada time é responsável por fazer uma função específica na empresa e sendo o discente pertencente a equipe de desenvolvimento da Ignis.

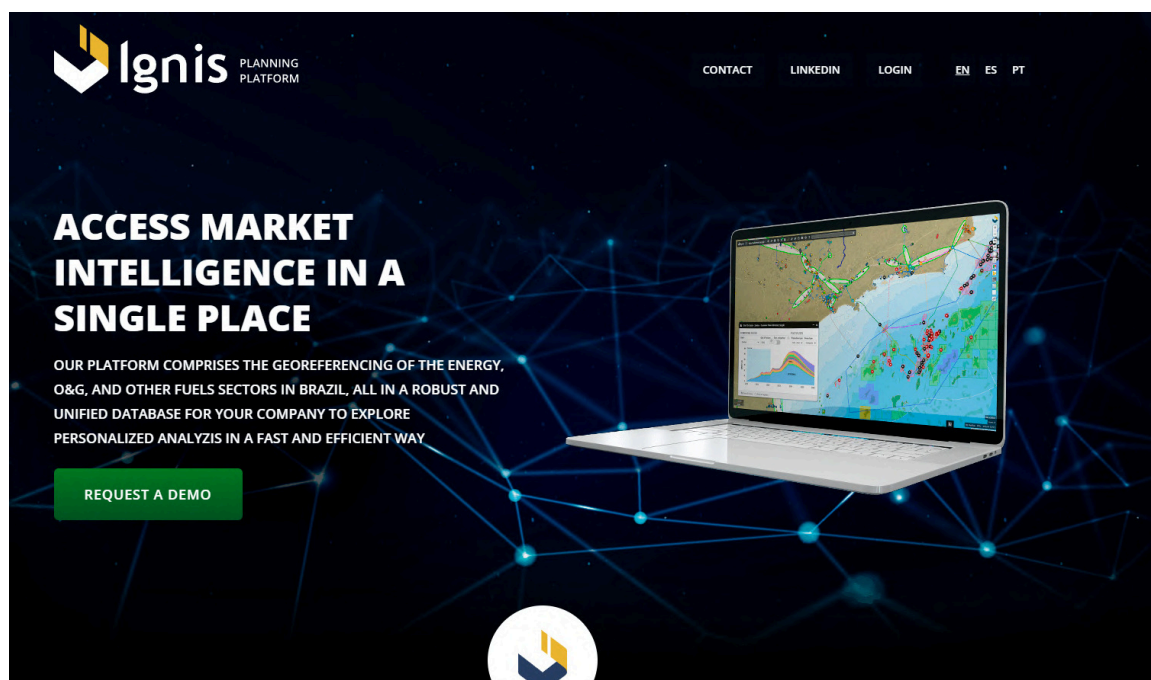
Figura 2 - Abstração da organização da empresa.



Fonte: Própria do autor (2024)

Sendo na empresa a principal fonte de renda os adquirentes que contratam a plataforma da Ignis através das palestras de óleo e gás, LinkedIn, entrando em contato pelas redes sociais ou sendo pela *Landing page* (página da internet responsável por converter visitantes em clientes) da Ignis, como ilustrado a seguir na Figura 3.

Figura 3 - Página inicial da Ignis.



Fonte: Própria do autor (2024)

A empresa que o discente estagiava conhecida no mercado como Ignis passou a pertencer a Alvarez & Marsal, logo da empresa na Figura 4, por questões estratégicas e organizacionais nos últimos meses de contrato de estágio supervisionado do discente, porém ainda mantendo o contrato inicial sobre o CNPJ da Ignis.

Figura 4 - Logomarca da Alvarez & Marsal.



Fonte: LinkedIn.

3.2 Processos e ambiente

A empresa Ignis adota uma metodologia ágil adaptada do *scrum*, metodologia essa que será abordada na seção de fundamentação teórica, para o gerenciamento e organização dos processos internos durante o desenvolvimento com interações de duas semanas. Para seguir essa abordagem é utilizado a ferramenta Azure Devops (ferramenta de gestão empresarial de tarefas), a qual possui um conjunto de serviços de colaboração, organização de processos internos e integração com ferramentas de desenvolvimento que facilitam a transparência para toda equipe.

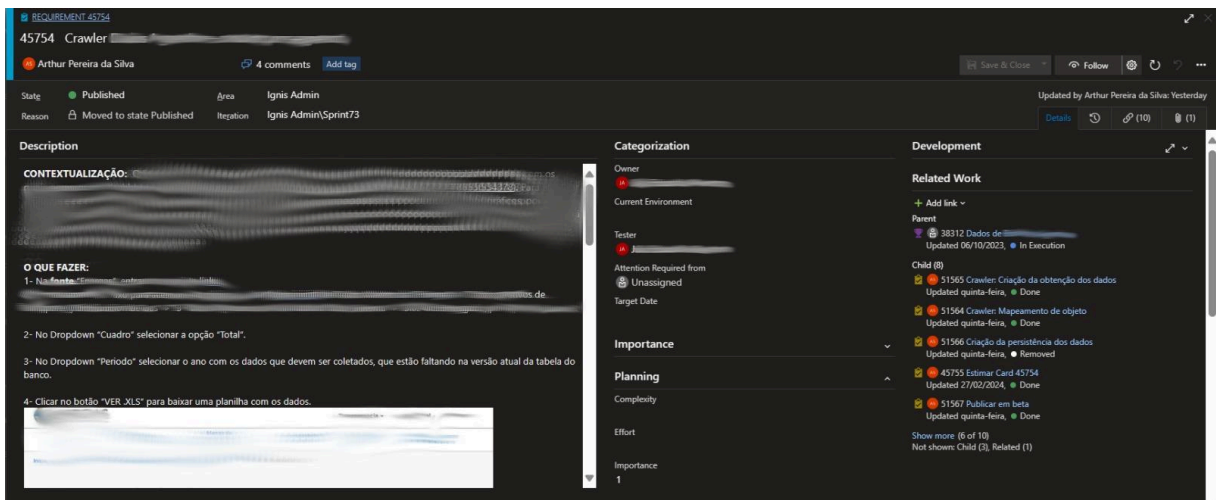
Dentro da empresa, para toda ação necessária, seja uma implementação ou correção de um *bug*, é elaborado um *card* contendo uma contextualização, o que tem que ser feito, para quem, pontos de atenção, definição de concluído e o testador. O *card* segue as seguintes etapas abaixo para o seu desenvolvimento:

1. Criação de uma *task* (tarefa) de elaboração do *card*.
2. Descrição completa do *card* seguindo as orientações citadas anteriormente.
3. Criação de uma *task* para o desenvolvedor estimar o esforço para a realização do *card*.
4. Estimar o *card* criando *tasks* e colocando o tempo que levará para a sua conclusão em ambiente de desenvolvimento.
5. Refino do *card*: processo opcional em que é feito um alinhamento com quem escreveu o *card* e o dev para reavaliar esforços e resolver possíveis dúvidas.

6. Colocar o *card* na *sprint* seguinte caso não seja um *bug* com uma prioridade alta.
7. Enquanto desenvolve as *tasks*, o card passa para o estado de *doing* (a fazer).
8. Após a conclusão das *tasks* por parte do dev as mudanças são colocadas em ambiente beta e é criada uma *task* para o testador avaliar e o card é colocado com o estado de *in test* (em teste).
9. Se o teste for aprovado é feito, as alterações são submetidas a um processo interno do GitKraken, que será abordado mais para frente no documento, antes da sua publicação em produção.
10. Caso o teste seja reprovado, o card volta para o estado de *doing* e é criada uma *task* para o dev aplicar as possíveis correções.

Outro ponto é que nos *cards* tem que ser referenciado a *feature* (funcionalidade do sistema) e colocado como o estado inicial de *to-do* (A fazer). Segue a Figura de como é um *card* no Azure Devops.

Figura 5 - Card no Devops.



Fonte: Própria do autor (2024)

4 FUNDAMENTAÇÃO TEÓRICA

Nesta seção será abordado e detalhado as tecnologias e processos utilizadas durante todo o período como estagiário na Ignis.

4.1 Metodologia Ágil (*scrum*)

Como mencionado, a Ignis utiliza para o desenvolvimento interno das suas tarefas uma versão adaptada da metodologia ágil *scrum* que pode ser definido como “um framework leve e simples criado para gerenciar o desenvolvimento de produtos. Com base no framework *scrum*, você pode adicionar as práticas, técnicas ou ferramentas que julgar mais adequadas de acordo com a sua indústria, seja ela desenvolvimento de software, marketing, recursos humanos, educação, etc.” - (RIBAS, s.d.).

No *scrum* existem as sprints que são um período fixo de tempo a qual a equipe se compromete a realizar um grupo de atividades específicas para entregar um incremento/funcionalidade do produto, sendo esse incremento algum requerimento que conste no *product backlog* (lista dinâmica e priorizada de funcionalidades e correções que precisam ser implementadas no produto).

Em relação às cerimônias que acontecem no *scrum* temos as *daily*s que são reuniões curtas que acontecem diariamente onde cada indivíduo expressa o que fez no dia, se teve algum impedimento e o que falta a ser feito, a *planning* é o planejamento da próxima *sprint*, o que cada indivíduo ou grupo vai buscar implementar e priorizar, e por fim a *retrospective* aonde é debatido os pontos positivos e negativos, sugestões de processo e apontamento sobre priorização no *product backlog*.

Há três papéis importantes para essa metodologia:

1. *Product Owner*: o responsável por representar os interesses e desejos dos *stackholders* sobre o produto;
2. *Scrum Master*: responsável por garantir que toda a metodologia ágil, como a comunicação e o andamento da *sprints*, aconteça da melhor maneira possível;

3. Equipe de desenvolvimento: equipe responsável pela elaboração e entrega do incremento ao decorrer das *sprints*;

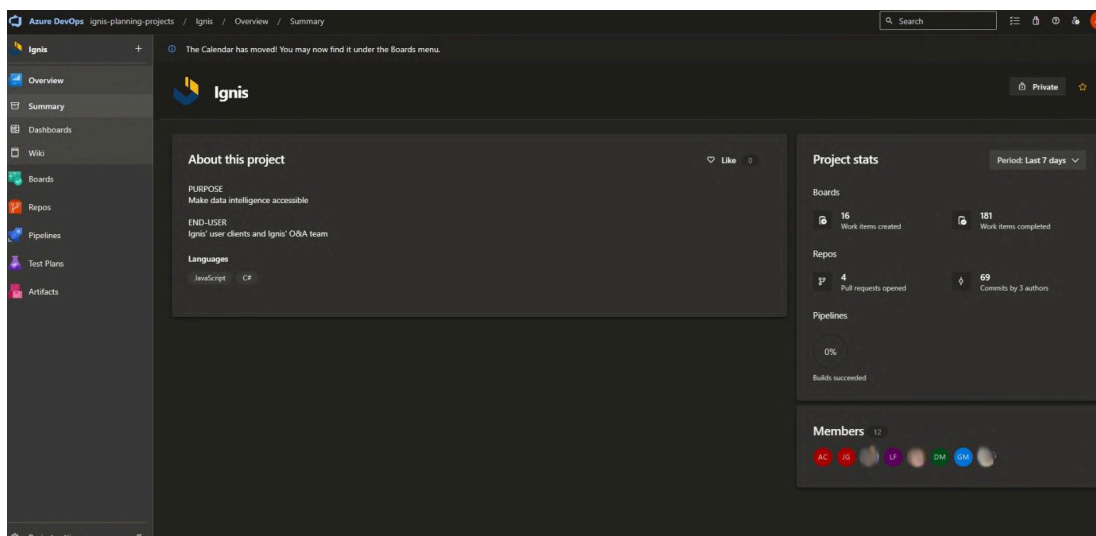
O discente teve contato com o *scrum* durante as disciplinas de Processos de Desenvolvimento, Análise e Projeto, Projeto 1 e Projeto 2, com isso, a adaptação ao processo estabelecido na empresa foi bastante facilitada e agradável.

Na Ignis, além dessas cerimônias do *scrum*, ainda possui outras reuniões de complemento a metodologia ágil, como:

- 101 (*one-on-one*): reunião entre um funcionário e o seu superior, onde era coletado o feedback por parte do funcionário sobre o andamento da *sprint*, possíveis impedimentos no desenvolvimento de suas *tasks*;
- Clube do livro: reunião que reunia todo o time da Ignis, onde era discutido algum tópico ou capítulo de um livro técnico com o intuito de levantar possíveis melhorias sobre o processo que a Ignis adota;
- Clube DEV: reunião contendo apenas a equipe de desenvolvimento da Ignis, sendo uma reunião mais técnica buscando a integração de novas tecnologias, padrões de desenvolvimento, boas práticas e mudanças nos processos internos da equipe;

4.2 Azure Devops

O Azure DevOps, desenvolvido pela Microsoft, é uma ferramenta que permite organizar as tarefas da empresa, registrar e monitorar o andamento dos projetos. O mesmo disponibiliza uma série de recursos personalizáveis. Como a criação de projetos, criação de *cards*, boards contendo o andamento de sprints, integração com repositório Azure para o versionamento dos projetos interno, entre outras formas. Na Figura 5 é mostrado como é a página Inicial do Azure.

Figura 6 - Tela inicial do Azure DevOps.

Fonte: Própria do autor (2024)

4.3 Teams

A Ignis utiliza para comunicação principal o aplicativo também da Microsoft que é o Teams, sendo um aplicativo para fazer a gestão inteligente de equipes dentro da empresa, funcionando como um ambiente de trabalho virtual contendo a possibilidade de fazer chamadas de vídeos e áudios, cronogramas e chats privados e públicos dentro da organização.

4.4 GitKraken

O GitKraken é uma ferramenta para se trabalhar com Git¹ utilizando uma interface gráfica mais amigável ao desenvolvedor, facilitando o versionamento de código, gerenciamento de repositórios, criação de *branches*, *commits*, *merges*, comandos que exerceriam um certo trabalho se fossem feitos utilizando o Git em linha comando e é utilizada por toda equipe de desenvolvimento na Ignis. Outro ponto a comentar que a realização dos commits era seguindo a nomenclatura semântica² para a padronização de processo.

¹ Informações adicionais sobre o Git em: [Git \(git-scm.com\)](https://git-scm.com)

² Informações adicionais sobre o commits semânticos em: [Commits Semânticos](#)

4.6 GitFlow

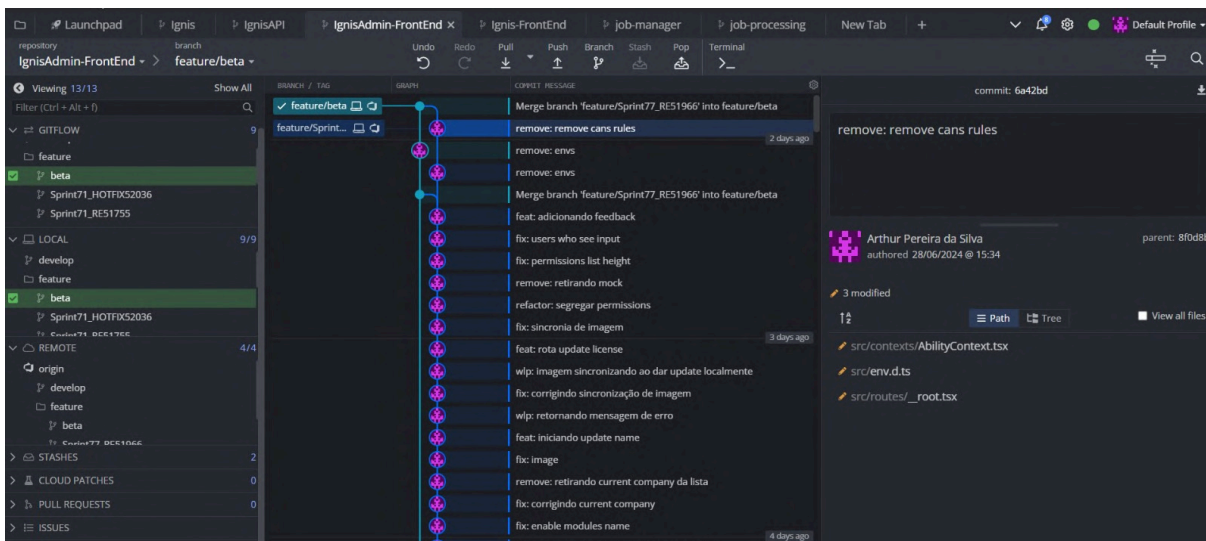
Agora, em relação à organização do processo de versionamento em Git na Ignis, utilizando o GitKraken comentado anteriormente, o GitFlow é “[...] um modelo, uma estratégia ou, ainda, um fluxo de trabalho muito utilizado por equipes de desenvolvimento de software. Ele se destaca por auxiliar na organização do versionamento de códigos.”. - (Pedroso, 2023).

O fluxo seguido era de ter várias ramificações (*branches*) para diferentes frentes no projeto para serem atacadas durante o desenvolvimento, na Ignis para todos os repositórios seguem a ideia do GitFlow, mas com algumas modificações como mostrado a seguir:

- *Master*: código mais estável em produção (no ar para o cliente), sendo considerado a *branch* principal.
- *Develop*: branch de desenvolvimento, contendo o código mais recente sendo desenvolvido pela equipe.
- *Beta*: código que estava entrando para testes para ser analisado pela equipe de Operações e análise.
- *Feature*: *branches* que eram criadas para desenvolver novas funcionalidades ou refatorações. O nome da *branch* é utilizado a sprint atual mais “_RE” e o número do card, como exemplo o “sprint52_RE54598”.
- *Bug*: como o nome sugere, é uma branch que era criada para a realização de alguma correção em alguma feature. Segue o padrão de nomenclatura parecido com o de *feature*, porém ao invés “_RE” é utilizado “_BUG”, a exemplo o “sprint53_BUG54598”.
- *Hotfix*: criada inicialmente para a resolução de *bugs* complexos que precisavam de correção imediatamente sem passar por testes ou problemas simples como nomenclatura de algum campo. Porém, para evitar dúvidas na criação de cards no Azure Devops, foi optado para as *hotfixes* serem apenas problema simples e os complexos ainda continuarem a ser *branches* de *bug*. A nomenclatura da *branch* também não foge muito do padrão, sendo usado o “_HOTFIX” ao invés de “_BUG” ou “RE”, como “sprint71_HOTFIX52036”.

Abaixo, um exemplo do GitFlow no GitKraken é apresentado na Figura 7.

Figura 7 - Exemplo de Fluxo do GitFlow no GitKraken.

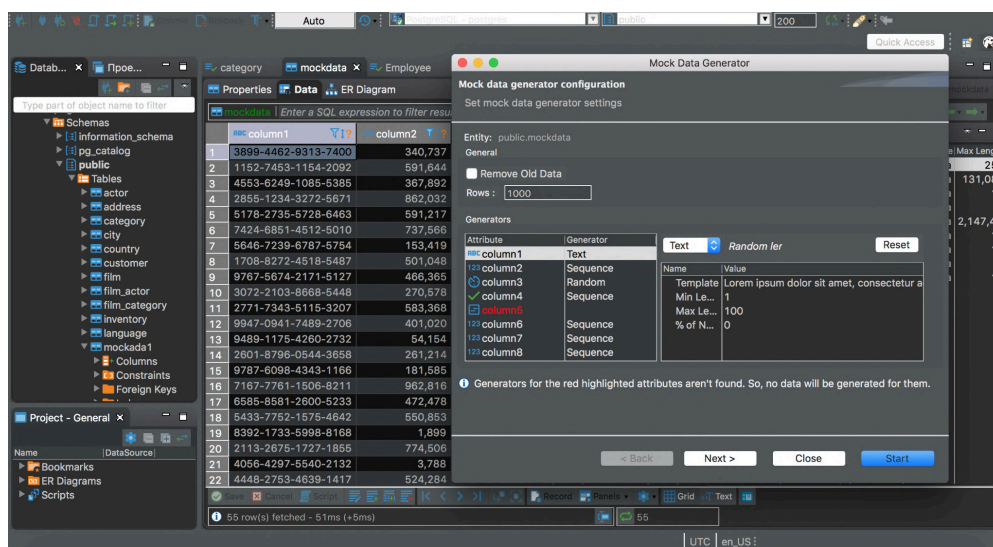


Fonte: Própria do autor (2024)

4.7 DBeaver

Para a gestão do banco de dados, que no caso da Ignis é o Postgres, é utilizado a ferramenta de gestão e administração de banco de dados, o DBeaver, pois possui uma interface intuitiva e de código aberto para os programadores. Segue o exemplo da interface na Figura 8.

Figura 8 - Interface do DBeaver



Fonte: Site oficial do DBeaver: [dbeaver](https://dbeaver.io)

4.8 Ferramentas para o desenvolvimento

Essa seção será para o esclarecimento das ferramentas utilizadas dentro da Ignis.

4.8.1 Microsoft Visual Studio

Para o desenvolvimento das aplicações que envolviam a linguagem C# dentro da Ignis, utilizaremos o Microsoft Visual Studio, uma IDE (Ambiente de Desenvolvimento Integrado) que é uma ferramenta da Microsoft com um dos melhores suportes para as linguagens C, C++ e C# e seus frameworks.

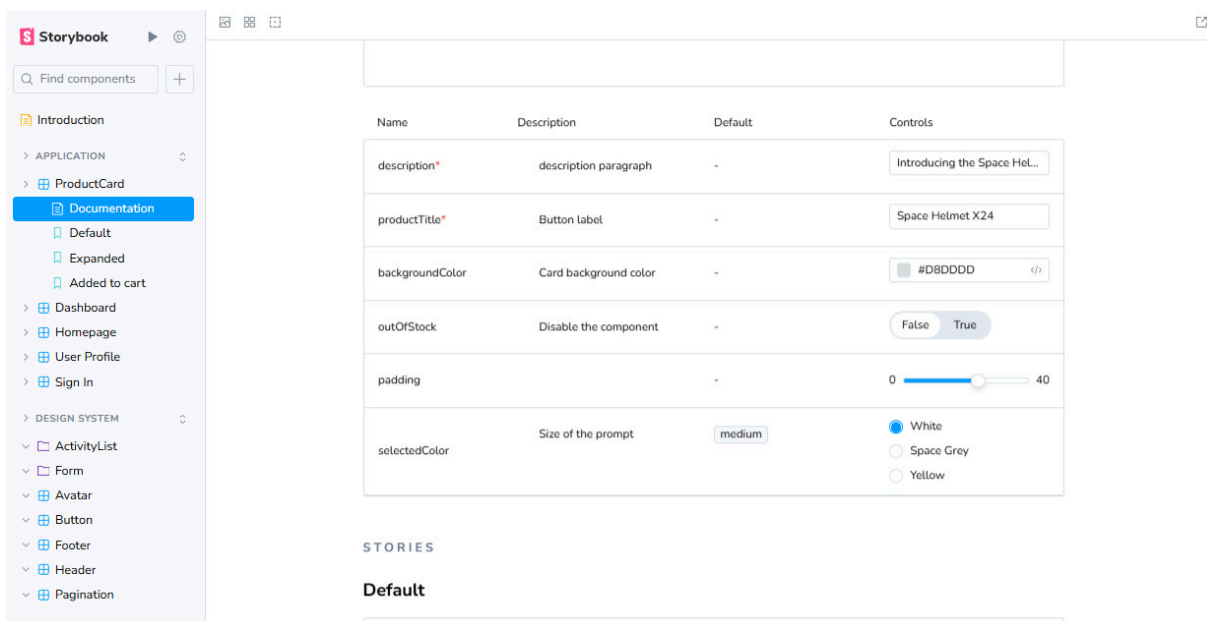
4.8.2 Visual Studio Code

Agora, para o desenvolvimento mais focado no *frontend* com a linguagem de programação Javascript, fora utilizado o Visual Studio Code, que é um editor de texto avançados mais conhecidos e utilizados. Ele possui suporte para várias linguagens no desenvolvimento, terminal integrado e uma vasta gama de extensões disponíveis.

4.9 Storybook

Para a documentação, o discente começou esse processo na Ignis juntamente com a atualização da plataforma administrativa para uma versão mais atual e manutenível (tópico abordado na seção de repositórios).

O Storybook é uma biblioteca *open-source* que procura criar um ambiente que isole seus componentes, de maneira que seja possível documentar os que possuem a capacidade de ser reutilizáveis em outros contextos dentro da aplicação ou que possuem “mutações” (capacidade de mudar a depender de uma entrada) em seu processo. Abaixo a interface principal do Storybook na Figura 9.

Figura 9 - Interface principal do Storybook

Fonte: Site oficial do Storybook: [storybook](https://storybook.js.org)

5 RELATO DE ESTÁGIO

5.1 Atividades desenvolvidas no estágio supervisionado

O discente começou suas atividades presentes neste documento ao longo de um período de 1 ano que esteve como estagiário de desenvolvimento de *software* na Ignis, realizando implementações de novas funcionalidades e manutenção para a plataforma principal e administrativa da empresa, além de participar da documentação do *frontend*.

Na primeira semana como estagiário, o discente teve um *onboarding* (processo de adaptação dentro de uma empresa) com cada *time*, sendo o primeiro com a equipe de operações e análise com o intuito de fazer um *tour* pela plataforma principal, dando ênfase na logística de cada seção e sua finalidade no sistema e por último *onboarding* com a equipe de desenvolvimento onde foi mostrado os processos internos, tecnologias utilizadas, padrões de desenvolvimento e arquitetura sendo todo esse treinamento foi supervisionado pelo desenvolvedor Leonardo Hummel de Sousa, além de configuração do ambiente de desenvolvimento e comunicação na máquina cedida pela empresa como mostrada na Figura 10.

Figura 10 - Notebook disponibilizado pela empresa.



Fonte: Própria do autor (2024).

5.1.1 Repositórios

Para facilitar o entendimento de qual ambiente o discente esteja comunicando nas seções seguintes, essa seção explicará brevemente os repositórios em que o discente teve contato e participação durante o estágio.

5.1.1.1 GeoPortal

Repositório existente criado da união da Ignis com a Geopost no passado, desenvolvido em C#, Vue e JQuery contendo uma solução (conjunto de projetos) com 9 projetos distintos, os que o discente teve participação: plataforma administrativa antiga (tanto a API quanto o *frontend* que também se encontrava nessa solução), *Crawler*, *Web* (API da plataforma principal) e *Models* (projeto contendo as entidades e regras de negócio).

5.1.1.2 IgnisApi

O IgnisApi é o novo repositório para API da Ignis, criado com o intuito de desvincular a logística da Geopost que está enraizada no repositório do GeoPortal da plataforma da Ignis e atualizar os projetos para a versão .Net 7 (e posteriormente .Net 8). Nesse repositório, o discente teve participação em todos os projetos, que no caso são: *Crawler*, API dedicada a nova plataforma administrativa, *Web* (também é a API da plataforma principal) e *Domain* (projeto contendo as entidades e regras de negócio).

5.1.1.3 Ignis-FrontEnd

Nesse repositório, também contando com a participação do discente, contém o *frontend* da aplicação principal da Ignis. Anteriormente fazia parte do repositório GeoPortal comentado anteriormente, integrando em um mesmo ambiente, o *frontend* e o *backend*.

Após esse desvinculo com a Geopost, o *frontend* foi migrado de Vue e JQuery no repositório GeoPortal e foi migrado para um repositório dedicado desenvolvido em React com *Typescript*, dentre outras ferramentas mais recentes de

desenvolvimento no mercado, além do *eslint* e *prettier* que servem para a padronização e formatação de código.

5.1.1.4 IgnisAdmin-FrontEnd

Repositório iniciado pelo discente como proposta da gestão para uma migração da plataforma administrativa para uma versão mais atualizada e manutenível, a qual foi desenvolvida com React com Typescript, Tanstack Query e Router, também utilizando o *eslint* e *prettier*, e outras bibliotecas. Como o nome indica, esse repositório é unicamente para o *frontend* da aplicação.

5.1.2 Crawlers

A primeira participação do discente em um projeto na equipe de desenvolvimento da Ignis foi na criação de um *crawler*³ na *sprint* 51. Sendo essa tarefa responsabilidade do discente e de outro desenvolvedor durante todo o processo de estágio, desde a responsabilidade de criação de um *crawler* até a sua publicação em ambiente de produção na Ignis, além de realizar a manutenção contínua.

Quando o discente chegou na Ignis ainda era utilizado o repositório GeoPortal para o desenvolvimento dos crawlers, e seguindo o que foi orientado por outro estagiário da equipe foi seguido os seguintes passos para o desenvolvimento.

Primeiramente, é entendido o que o *card* necessita e assim é feito o *crawler*, levando em consideração como será feito a coleta e o como será feito. Dessa maneira, é visto se será utilizado tecnologias como o Selenium (biblioteca *open-source* que serve tanto para fazer testes automatizados quanto na automação de navegadores da web), HTML Agility Pack (biblioteca para processamento de documentos HTML em .NET), *regex* (ferramenta presente em quase todas as linguagens para encontrar padrões em strings), EPPlus (biblioteca para se trabalhar com Excel), etc.

³ Informações adicionais sobre Crawlers em: [O que são crawlers.](#)

Agora, para o mapeamento e construção de um crawler é utilizado dois frameworks diferentes para o mapeamento de uma classe (entidade) para uma tabela no banco de dados. Que no caso, são o NHibernate (framework de mapeamento de objeto-relacional) que é um framework presente no repositório do GeoPortal e o entity framework (framework também para o mapeamento de objeto-relacional) que está presente no repositório no IgnisApi. Para elucidação de possíveis dúvidas, teremos duas seções explicando cada etapa da construção dos crawlers em cada repositório.

5.1.2.1 Crawlers pelo NHibernate no GeoPortal

Como já mencionado, o NHibernate é um framework para o mapeamento de objeto-relacional, e sua utilização seguia as seguintes etapas para o seu desenvolvimento no GeoPortal, levando em consideração que todos os detalhes de implementação do crawler já foram elucidados:

1. Começa com a criação de uma *branch* derivada da *branch* de produção, e depois iniciado a construção de uma classe base do objeto com o nome do que queremos coletar com o crawler que herde do *BaseModel* (entidade que define os valores padrões para todos os crawlers) e implementando a interface *ICrawlableObject* que impõem os atributos em relação à data de inserção e data de atualização do valor na tabela;
2. Dando continuidade, é feito o mapeamento da entidade do crawler para a tabela utilizando a classe *ClasseMap* do NHibernate, é criado uma classe chamada *NomeDaEntidadeMap*, onde é feita essa implementação conforme as exigências do *card*;
3. É criado manualmente a tabela no banco e a inserção de uma *key* (chave) que o NHibernate usa de base para saber qual será o próximo índice do elemento a ser inserido naquela tabela, exemplo do *script SQL* na Figura 11.

Figura 11 - Exemplo de script SQL para criar a tabela e a chave no NHibernate

```
CREATE TABLE dados_gas_energy.index_dollar (
  id int8 NOT NULL,
  "year" int4 NULL,
  "month" int4 NULL,
  price numeric NULL,
  insert_datetime timestamp NULL,
  update_datetime timestamp NULL,
  CONSTRAINT index_dollar_pkey PRIMARY KEY (id)
);

insert into nhibernate.table_key (next_value, tipo) values (1, schema.index_dollar)
```

Fonte: Própria do autor (2024)

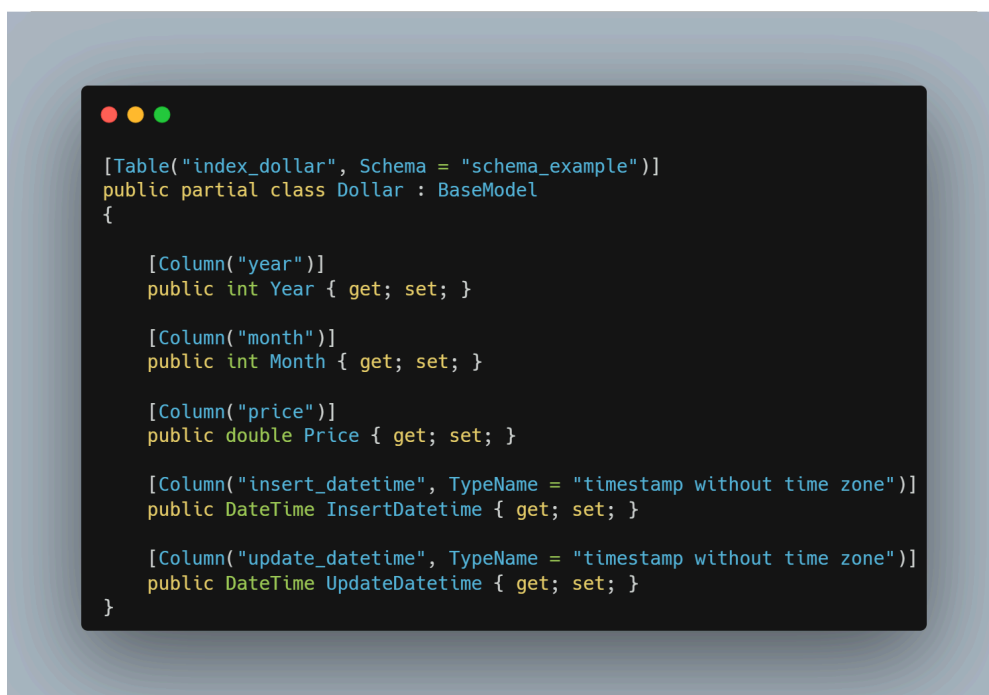
4. Posteriormente, é criado o crawler para coleta desse dado usado a nomenclatura de *CrawlerNomeDoDado*, exemplo: *CrawlerIndexDollar*. Onde é uma classe que herda da classe base de todos os *crawlers*, contendo a persistência (inserir o dado no banco), o tipo de *crawler*, *URL* da base utilizada e configurações de notificação pelo email de sucesso ou se houve algum erro na coleta.
5. É adicionado o *crawler* na *factory* que é uma classe responsável por rodar os *crawlers* utilizando o argumento de linha do comando para saber qual *crawler* que tem que entrar em execução naquele momento.
6. Ao conferir que os dados provenientes do *crawler* estão sendo corretamente inseridos na tabela utilizando o DBEaver, é implementado o *crawler* no servidor em beta e na *branch* beta e é posteriormente utilizado o DBEaver novamente para gerar a tabela no formato em CSV para que a equipe de operação e análise possa validar os dados.
7. Por fim, caso a validação dos dados do crawler esteja correta, é feito o *pull request* (subir o código para que outros desenvolvedores validem o código) pelo GitKraken e com isso subir o *crawler* para a *branch* de produção e também para o servidor de produção da Ignis.

5.1.2.2 Crawlers pelo Entity Framework no IgnisApi

O processo seguido para a publicação e a validação do *crawler* é o mesmo que o do GeoPortal, com exceção do desenvolvimento e integração com o banco de dados, como alguns pontos que são:

1. Ao criar uma classe base, é feita o mapeamento em cima dela, utilizando anotações disponibilizadas pelo *Entity*, sem a necessidade da criação de uma classe para fazer esse mapeamento, como mostrado na Figura 12.

Figura 12 - Exemplo de mapeamento de classe pelo *Entity*



```
[Table("index_dollar", Schema = "schema_example")]
public partial class Dollar : BaseModel
{
    [Column("year")]
    public int Year { get; set; }

    [Column("month")]
    public int Month { get; set; }

    [Column("price")]
    public double Price { get; set; }

    [Column("insert_datetime", TypeName = "timestamp without time zone")]
    public DateTime InsertDatetime { get; set; }

    [Column("update_datetime", TypeName = "timestamp without time zone")]
    public DateTime UpdateDatetime { get; set; }
}
```

Fonte: Própria do autor (2024)

2. Posteriormente, é criada uma classe que implementa a interface *IEntityConfiguration*, sendo usada para configurar o comportamento e definir regras específicas no contexto do *Entity*.
3. Seguidamente, é criada uma referência da classe base na classe de contexto do *Entity* onde que é usado para fazer a persistência e conexão com o banco.

4. Para conectar a entidade do *crawler* com a tabela no Postgres é gerado uma *migration* que é uma maneira de sincronizar a entidade com uma tabela no banco de dados, além de mostrar o histórico de alterações da tabela.
5. E ao construir o crawler, todas as operações de *insert* (inserir), *update* (atualizar) ou *delete* (deletar) são utilizando o contexto da classe do *Entity*.
6. Outro ponto de atenção presente só no IgnisApi é, ao publicar o *crawler* em produção, segue o mesmo processo de validação do que os *crawlers* no GeoPortal, é para publicar também a API, pois os *crawlers* podem estar correlacionado a mudança em dados já existentes, podendo causar algumas interferências em gráficos existentes na plataforma principal.

5.1.3 Processo de desenvolvimento

Essa seção possui o intuito de explicar o processo de desenvolvimento dos projetos seguintes. Era seguido as seguintes etapas:

1. Começa com a criação de uma *branch* derivada da *branch* de produção, e depois iniciado o processo de desenvolvimento da *feature* ou correção de um bug.
2. Pós-desenvolvimento, é subido a branch derivada para a *branch* beta para a validação do time de operação.
3. Com a validação do time de operação, é feito o *pull request* com as alterações e implementações no código para a *branch* de produção.

5.1.4 Backend da aplicação principal

O discente teve contato com o *backend* da aplicação principal que está tanto presente no repositório da GeoPortal quanto no IgnisApi, sendo responsável pela implementação de *endpoints* (pontos de conexão entre o *frontend* e o *backend*), serviços que continham regras de negócios, além de fazer correções quando necessário e participar de processos de refatoração de código.

5.1.5 Frontend da aplicação principal

Nesse projeto, que está tanto presente no repositório do Ignis-FrontEnd com exceção do Mapa que é a *feature* principal da Ignis que ainda tá bastante presente no GeoPortal. O discente ficou responsável pela implementação de componentes, páginas, utilitários e roteamento utilizando REACT, seguindo as orientações dos meus superiores para seguir padrões de codificações.

5.1.6 Backend da aplicação administrativa

No *backend* da aplicação administrativa, o discente participou em dois momentos distintos. Primeiro, antes da migração da plataforma administrativa do repositório do GeoPortal para o IgnisApi, teve como responsabilidade entender o fluxo de autenticação, principais *features* que eram utilizadas e a complexidade dessa migração. No início da migração, esteve responsável pelo desenvolvimento dos *endpoints*, refatoração de código e ajustes nas regras de negócios juntamente com outro desenvolvedor, porém o discente ficou mais responsável pelo *frontend*.

5.1.7 Frontend da aplicação administrativa

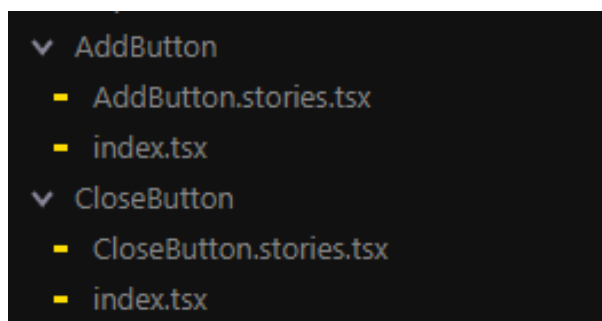
O *frontend* da aplicação administrativa é o projeto onde o discente possuiu mais responsabilidade juntamente com o projeto de *crawlers*, pois com as discussões sobre a migração, o discente ficou com a responsabilidade de iniciar o repositório IgnisAdmin-FrontEnd e todo o processo para o desenvolvimento e documentação das *features* no *frontend*.

Primeiramente, o discente propôs em reuniões do clube DEV (reunião explicada na seção de fundamentação teórica) a utilização do Storybook para a documentação de componentes, utilitários e *hooks*⁴ que eram utilizados durante todo o processo de desenvolvimento. Com o objetivo de garantir uma documentação clara, detalhada e concisa para os demais desenvolveres, o discente desenvolveu o seguinte processo para documentar:

⁴ O que são hooks: [React hooks: o que é, porque usar e exemplos! – Insights para te ajudar na carreira em tecnologia | Blog da Trybe \(betrybe.com\)](#)

1. Implementação da *feature* que o *card* exigia.
2. Refatoração de código e quebra de componentes, pois apenas os componentes que eram reutilizáveis em outras páginas ou por outros componentes seriam documentados.
3. Eram gerados arquivos com o nome dos componentes seguidos de “stories” no final, assim permitindo ao Storybook identificar que se tratava de um arquivo para documentar um componente, como exemplificado na Figura 13.

Figura 13 - Exemplo de organização de pastas de documentação



Fonte: Própria do autor (2024)

4. Com todos os componentes documentados da *feature*, é adicionado os utilitários e *hooks* em dois arquivos, o `UtilsDocs.stories.tsx` e o `HookDocs.stories.tsx`.

Outro ponto que o discente propôs em reuniões do clube DEV, fora a utilização do Tanstack Router⁵ para o roteamento das páginas, algo iniciado na plataforma administrativa e que já foi mapeado para ser implementado juntamente com o Storybook para a plataforma principal.

⁵ Documentação do Tanstack Router: [TanStack Router](#)

6 CONSIDERAÇÕES FINAIS

Durante todo o processo de estágio supervisionado na empresa Ignis, o discente pôs em prática vários conhecimentos essenciais que adquiriu durante toda a sua formação acadêmica no curso superior de tecnologia em análise e desenvolvimento de sistemas, além de vivenciar toda experiência de um desenvolvedor de *software* no mercado de trabalho.

Compreendendo esses conhecimentos que o discente obteve em sua formação acadêmica no curso de ADS, podemos elencar as disciplinas como Padrões de Projeto, Processos de Desenvolvimento, Desenvolvimento de Aplicações Corporativas, Banco de Dados I e II e Projeto I que fizeram com que o discente tivesse uma base sólida sobre normas, padrões e princípios (SOLID⁶ e GRASP⁷) no desenvolvimento de *softwares*, além da logística de lidar com os *stakeholders* (pessoas interessadas no *software*) desde abranger de como entender o que é realmente necessário para a aplicação até como fazer apresentações propostas pela empresa. Também teve o contato com a metodologia ágil para o desenvolvimento o *scrum* durante essas disciplinas que foi vital para a contratação do discente como estagiário na Ignis e a sua efetivação ao fim do período de estágio.

O discente, no começo, teve dificuldades com o processo de GitFlow, desenvolvimento, publicação de *crawlers* no servidor, no preenchimento dos cards e também ao desenvolver no *frontend* da aplicação. Por isso, é destacável a importância que as reuniões de 101 com meu supervisor do estágio, Leonardo Hummel de Sousa, os *Onboarding* e o suporte dos outros desenvolvedores para o entendimento e evolução do discente durante todo o estágio supervisionado dentro da empresa Ignis.

Em relação aos pontos positivos que o discente teve enquanto estagiário na Ignis foi a comunicação que toda equipe tem entre si, qualquer possível gargalo durante as *sprints* era resolvido nas *dailies* ou nas demais reuniões.

É válido salientar que, tiveram os processos incentivados e pagos pela empresa de adaptações com as ferramentas, processos e tecnologias através da plataforma Alura (plataforma de conteúdo cursos sobre diferente de várias áreas,

⁶ O que é SOLID: O guia completo para você entender os 5 princípios da POO

⁷ Padrões GRASP — Padrões de Atribuir Responsabilidades

principalmente de informática), além da flexibilidade com o horário que todo funcionário possui.

Outro ponto, é que desde que o discente chegou na Ignis tivera a responsabilidade de desenvolver e prestar manutenção de *crawlers*, sendo essa uma tarefa de grande importância na Ignis e que depois de um tempo ficou como uma das tarefas principais do discente juntamente com a inicialização e desenvolvimento da plataforma administrativa.

Após esse período de um ano como estagiário na Ignis, o discente recebeu a proposta de efetivação na empresa como desenvolvedor PJ *fullstack* em *home office*, e posteriormente depois da compra da Ignis pela Alvarez & Marsal, o discente continuou atuando na Ignis que agora é um dos vários módulos da Alvarez & Marsal.

Por fim, é importante destacar a maturidade que o discente adquiriu ao passar por cada projeto dentro da Ignis, um ambiente inserido no mercado de trabalho, e como os conhecimentos em SQL, programação orientada a objetos e padrões de projetos com JAVA e todo o processo de metodologia ágil desenvolvida e simulado dentro da formação acadêmica fornecida pelo Instituto Federal da Paraíba - *Campus Monteiro* estão alinhados a realidade de um desenvolvedor e analista. Contudo, também é válido salientar, toda a dedicação e perseverança que o discente teve, tanto no sentido técnico quanto no comportamento, foram importantes para alcançar os objetivos propostos pelo estágio supervisionado.

REFERÊNCIAS

CASTELO BRANCO, Dácio. **O que é web scraping e como ocorre?**. [S. l.], 23 out. 2021. Disponível em: <https://canaltech.com.br/seguranca/o-que-e-web-scraping/>. Acesso em: 10 mar. 2024.

FIGUEREDO, Vitor. **O que é uma landing page? [guia completo]**. [S. l.], 22 jan. 2024. Disponível em: <https://nimbusdigital.com.br/blog/landing-page-guia-completo/#:~:text=Landing%20page%20nada%20mais%20%C3%A9%20do%20que%20uma,ofertas%20dentro%20de%20uma%20estrat%C3%A9gia%20de%20marketing%20digital>. Acesso em: 7 abr. 2024.

GIT: o que é, para que serve e principais comandos Git!. [S. l.], 13 abr. 2022. Disponível em: <https://blog.betrybe.com/git/>. Acesso em: 13 maio 2024.

MARTIN, Robert C. Ensino, Aprendizagem e Habilidade: Conclusão. *In: O CODIFICADOR LIMPO*. [S. l.: s. n.], 2011. cap. 14.

MICROSOFT. **O que é o Azure DevOps?: Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019**. [S. l.], 25 mar. 2024. Disponível em: <https://learn.microsoft.com/pt-br/azure/devops/user-guide/what-is-azure-devops?view=azure-devops>. Acesso em: 13 abr. 2024.

MOREIRA NETO, Rivaldo. **Descrição da sua experiência como Diretor da Ignis**. [S. l.], [s.d.]. Disponível em: <https://www.linkedin.com/in/rivaldomoreiraneto/>. Acesso em: 16 set. 2024.

OLIVA, Maria Gabriela. **[PT-BR] Conhecendo o DBeaver**. [S. l.], 24 jan. 2020. Disponível em: <https://dev.to/mariagcoliva/pt-br-conhecendo-o-dbeaver-2ka8>. Acesso em: 18 ago. 2024.

PEDROSO, Murillo Godoi. **Git Flow: entenda o que é, como e quando utilizar**. [S. l.], 18 set. 2023. Disponível em: <https://www.alura.com.br/artigos/git-flow-o-que-e-como-quando-utilizar>. Acesso em: 16 set. 2024.

RIBAS, Thomaz. **Scrum: O que É, Como Funciona e Exemplos Práticos [GUIA]**. [S. l.], 2021. Disponível em: <https://thomazribas.com/agile/scrum#t-1628084680682>. Acesso em: 13 abr. 2024.

SHILMAN, Michael. **Storybook DocsPage**. [S. l.], 21 ago. 2019. Disponível em: <https://medium.com/storybookjs/storybook-docspage-e185bc3622bf>. Acesso em: 18 ago. 2024.

TUA CARREIRA. **Estágio supervisionado: entenda o que é e saiba qual a sua importância**. [S. l.], 1 jan. 2021. Disponível em: <https://www.tuacarreira.com/estagio-supervisionado/#:~:text=Est%C3%A1gio%20supervisionado%3A%20o%20que%20%C3%A9%3F%20O%20est%C3%A1gio%20supervisionado,exemplo%3A%20o%20aluno%20est%C3%A1%20na%20faculdade%20cursando%20Letras>. Acesso em: 30 mar. 2024.


ANEXO A - Detalhes do desafio

O desafio

Desenvolver um sistema de torneio de futebol de pontos corridos onde todos os clubes se enfrentam em dois turnos. Este sistema deverá ter os seguintes recursos:

1. Ler um TextArea contendo uma lista de times e seu respectivo estado separados por ";" Exemplo:
Vasco;Rio de Janeiro
Flamengo;Rio de Janeiro
Palmeiras;São Paulo
Santos;São Paulo
Cruzeiro;Minas Gerais
Internacional;Rio Grande do Sul
2. Exibir na tela a combinação de jogos para todas as equipes, indicando a cidade do jogo, turno de ida, por exemplo: Vasco vs Flamengo - Rio de Janeiro - Rodada 1
Palmeiras vs Cruzeiro - São Paulo - Rodada 1
Santos vs Internacional - São Paulo - Rodada 1
Flamengo vs Santos - Rio de Janeiro - Rodada 2
Palmeiras vs Internacional - São Paulo - Rodada 2
Cruzeiro vs Vasco - Minas Gerais - Rodada 2
3. Printar na tela a combinação de jogos de retorno, invertendo o mandante de cada jogo do turno de ida.
4. Gerar um resultado randômico para cada jogo gerado no turno e retorno.
5. Determinar o campeão, considerando: Vitória = 3 pontos; Empate = 1 ponto, sem levar em conta saldo de gols.
6. Sinalizar as partidas com o texto "Rodada Dupla" quando houverem dois jogos na mesma cidade na mesma rodada, como no exemplo abaixo:

Vasco vs Flamengo - Rio de Janeiro - Rodada 1
Palmeiras vs Cruzeiro - São Paulo - Rodada 1 (RODADA DUPLA)
Santos vs Internacional - São Paulo - Rodada 1 (RODADA DUPLA)

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Monteiro - Código INEP: 25284940
	Pb-264, S/N, Serrote, CEP 58500-000, Monteiro (PB)
	CNPJ: 10.783.898/0008-41 - Telefone: (83) 3351-3700

Documento Digitalizado Ostensivo (Público)

Relatório de Estágio

Assunto:	Relatório de Estágio
Assinado por:	Arthur Pereira
Tipo do Documento:	Relatório
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Arthur Pereira da Silva, DISCENTE (202115020022) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - MONTEIRO**, em 23/12/2024 19:01:22.

Este documento foi armazenado no SUAP em 23/12/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1348428

Código de Autenticação: 048106bcd

