

Estágio na empresa omnium:retail na área de Retalho

Nelson Aguiar a39110

*Relatório de Estágio apresentado à Escola Superior de Tecnologia e Gestão para
obtenção do Grau de Mestre em Informática*

Trabalho realizado sob a orientação de:

Professor Rui Pedro Lopes

Engenheiro António Moreira

Bragança

2023

Resumo

Este relatório, no âmbito da unidade curricular "Dissertação/Trabalho de Projeto/Estágio", visa relatar os conhecimentos adquiridos, trabalhos realizados e experiências durante o estágio realizado na *omnium:retail*, sob a orientação do Engenheiro António Moreira e do Professor Rui Pedro Lopes.

O estágio concentrou-se principalmente em programação *Backend*, utilizando a linguagem C# em conjunto com as tecnologias .NET/MongoDB/PostgreSQL, com especial ênfase na área do retalho. Os projetos desenvolvidos ao longo do estágio tiveram como foco a implementação de um conjunto de micro-serviços, destinados a gerir fluxos de informação bidirecionais entre canais de venda físicos e digitais.

Durante o período do estágio, obtive um vasto leque de conhecimentos tanto no âmbito prático como no social. Neste relatório será abordada a metodologia de trabalho adotada, bem como a importância do trabalho em equipa no seio da empresa e os benefícios proporcionados por esta experiência profissional.

Destaco, entre os principais resultados alcançados durante o estágio, a implementação bem-sucedida dos micro-serviços de gestão de fluxos de informação, que contribuíram para uma maior integração e eficiência dos canais de venda da empresa. Além disso, pude aprofundar os meus conhecimentos em programação *Backend* e adquirir experiência prática valiosa na utilização das tecnologias C#, .NET, MongoDB e PostgreSQL.

Esta experiência de estágio teve um impacto significativo no meu crescimento pessoal e profissional. Através do trabalho em equipa e da orientação dos meus supervisores, pude aprimorar as minhas habilidades de comunicação, resolução de problemas e colaboração, as quais considero fundamentais para o sucesso em ambientes profissionais. Estou entusiasmado com o que aprendi e confiante de que essas experiências iram-me capacitar enfrentar desafios futuros no campo da programação e desenvolvimento de sistemas.

Palavras-chave: omnium:retail, C#, .NET, MongoDB, PostgreSQL

Abstract

This report, as part of the course "Dissertation/Project Work/Internship," aims to describe the acquired knowledge, tasks performed, and experiences during the internship at omnium:retail, under the guidance of Engineer António Moreira and Professor Rui Pedro Lopes.

The internship primarily focused on Backend programming using the C# language in conjunction with .NET/MongoDB/PostgreSQL technologies, with a special emphasis on the Retail industry. The projects developed during the internship focused on implementing a set of microservices designed to manage bidirectional information flows between physical and digital sales channels.

Throughout the internship period, I gained a wide range of practical and social knowledge. This report will address the adopted working methodology, the importance of teamwork within the company, and the benefits provided by this professional experience.

Among the main achievements during the internship, I highlight the successful implementation of the information flow management microservices, which contributed to a greater integration and efficiency of the company's sales channels. Additionally, I was able to deepen my knowledge of Backend programming and gain valuable hands-on experience with C#, .NET, MongoDB, and PostgreSQL technologies.

This internship experience had a significant impact on my personal and professional growth. Through teamwork and guidance from my supervisors, I was able to improve my communication, problem-solving, and collaboration skills, which I consider fundamental for success in professional environments. I am excited about what I have learned and confident that these experiences will empower me to tackle future challenges in the field of programming and system development.

Keywords: omnium:retail, C#, .NET, MongoDB, PostgreSQL

Índice Geral

Conteúdos

Capítulo 1	Introdução	1
1.1.	Enquadramento	1
1.2.	Objetivos	2
1.3.	Estrutura do documento	3
Capítulo 2	Contexto do estágio	5
2.1.	omnium:retail	5
2.2.	Plano de Estágio.....	6
2.3.	Estado da Arte.....	7
2.3.1.	Setor do Retalho	7
2.3.2.	Tecnologias e Frameworks	9
2.3.2.1.	Swagger	9
2.3.2.2.	Entity Framework.....	10
2.3.2.3.	RabbitMQ.....	11
2.3.3.	Micro-Serviços e a sua Evolução	13
2.3.4.	Metodologias	16
Capítulo 3	Desenvolvimento e interações do Estágio	18
3.1.	Formações	18
3.2.	Acompanhamento Inicial	21
3.3.	Interações e Trabalho em Equipa.....	23
3.3.1.	Colaboração em Equipa	23
3.3.2.	Comunicação com os Gestores e Mentores	23
3.4.	Estrutura Diária	23
3.4.1.	Reuniões e Acompanhamentos.....	23
3.4.2.	Tarefas e Horas	24
3.4.3.	Revisões e Feedbacks	25
Capítulo 4	Projetos	26
4.1.	Projetos.....	26
4.1.1.	Projeto X.....	26
4.1.2.	Conector HUB PostgreSQL.....	28
4.1.3.	Projeto PrestaShop.....	30
Capítulo 5	Conclusão	34
5.1.	Reflexão e Resumo do Estágio	34
5.2.	Conclusão.....	34

Bibliografia	1
--------------------	---

Índice de Figuras

Figura 1: omnium:retail	5
Figura 2: Plano.....	6
Figura 3: Retalho	8
Figura 4: Swagger documentation	9
Figura 5: Publish/Subscribe Messaging.....	12
Figura 6: Scrum	17
Figura 7: Antes e depois (Daily-report).....	22
Figura 8: Clockify tarefas	24
Figura 9: Fluxo do produto.....	28
Figura 10: Fluxo dos dados.....	30
Figura 11:PrestaShop.....	30
Figura 12:Fluxo de integração	32

Capítulo 1 Introdução

O objetivo deste relatório é apresentar uma análise detalhada dos conhecimentos adquiridos, dos trabalhos realizados e das experiências vivenciadas durante o estágio na omnium:retail. Ao longo deste documento, serão abordadas as atividades desenvolvidas, desde formações até projetos concretos, que visaram a implementação de um conjunto de micro-serviços para gerir fluxos de informação entre canais de venda físicos e digitais.

No final deste relatório, será possível compreender a relevância e os benefícios deste estágio, tanto em termos de crescimento pessoal e profissional, quanto em relação à contribuição para a omnium:retail. Espera-se que este relatório sirva como uma fonte de conhecimento e reflexão sobre a experiência vivenciada e as aprendizagens obtidas ao longo do estágio.

1.1. Enquadramento

O estágio na omnium:retail, no âmbito do Mestrado em Informática na ESTiG do IPB, concentrou-se no Backend para o setor de retalho, sendo parte da unidade curricular de "Dissertação; Trabalho de Projeto; Estágio" necessária para a conclusão do curso.

Este relatório detalha as atividades realizadas durante o estágio, fornecendo uma análise das tarefas, desafios, tecnologias utilizadas e a sua relevância para o meu desenvolvimento profissional.

Explorando especificamente o desenvolvimento de soluções de Backend no contexto do retalho, o estágio foi fundamental para aplicar os conhecimentos teóricos, na prática.

O relatório visa apresentar de forma sucinta as experiências e aprendizados obtidos durante o estágio na omnium:retail, destacando a importância prática dos conhecimentos adquiridos.

1.2. Objetivos

Objetivos do Estágio:

1. Adquirir conhecimentos práticos na programação Backend utilizando a linguagem C# e as tecnologias .NET/MongoDB/PostgreSQL aplicadas ao setor do Retalho.
2. Desenvolver competências na implementação de micro-serviços para gerir fluxos de informação entre canais de venda físicos e digitais.
3. Familiarizar-se com a metodologia de trabalho adotada na omnium:retail e compreender a importância do trabalho em equipa na empresa.
4. Obter uma compreensão aprofundada dos desafios e práticas específicas do setor de Retalho no contexto da programação Backend.
5. Aperfeiçoar habilidades de resolução de problemas, comunicação e colaboração num ambiente profissional.
6. Desenvolver uma compreensão abrangente dos princípios de arquitetura de micro-serviços, destacando a capacidade de projetar, implementar e otimizar soluções no contexto do setor de Retalho.
7. Aplicar os conhecimentos teóricos adquiridos durante o curso em projetos reais, contribuindo para o desenvolvimento e aprimoramento dos sistemas da omnium:retail.
8. Demonstrar habilidades avançadas na concepção e execução de projetos práticos, focando na integração eficiente de sistemas de backend, abrangendo desde a análise de requisitos até a implementação e documentação.
9. Estabelecer uma base sólida de experiência profissional no campo da programação Backend e do setor de Retalho, visando aprimorar a empregabilidade e as oportunidades futuras.
10. Refletir sobre a experiência de estágio, identificando os principais aprendizados, desafios enfrentados e as contribuições pessoais para a equipa e para a empresa.

Estes objetivos direcionaram as atividades realizadas durante o estágio, buscando um desenvolvimento prático e teórico alinhado com as necessidades e demandas da omnium:retail e do setor de Retalho.

1.3. Estrutura do documento

Este relatório está dividido em 5 Capítulos e é organizada da seguinte forma:

- **Capítulo 1:** Introdução

Na secção de introdução, será fornecido um contexto geral do estágio, descrevendo os objetivos a serem alcançados durante o período de estágio e a estrutura do documento.

- **Capítulo 2:** Contexto do Estágio

Esta secção descreverá o contexto do estágio, começando com uma visão geral da omnium:retail, apresentando a empresa onde o estágio foi realizado. Em seguida, será abordado o plano de estágio, destacando a sua importância na orientação das atividades e o alinhamento com os requisitos do programa académico. Por fim teremos o estado da arte o que irá trazer uma ampla visão aos temas abordados neste relatório.

- **Capítulo 3:** Desenvolvimento e Interações do Estágio

Neste capítulo concentra-se na experiência geral do estágio, com ênfase no trabalho em equipa e na colaboração. Serão discutidas as formações realizadas, fornecendo uma visão dos tópicos abordados e das habilidades adquiridas. Além disso, serão abordados o acompanhamento recebido descrevendo as interações com colegas e as reuniões de equipa, além da estrutura diária e as tarefas específicas desempenhadas durante o estágio.

- **Capítulo 4:** Projetos

Neste capítulo mergulhamos na parte mais prática do estágio apresentando os projetos nos quais foram trabalhados durante o estágio, incluindo detalhes sobre o projeto X e o projeto PrestaShop, destacando as tarefas realizadas, desde a análise de requisitos até ao desenvolvimento do código, testes e documentação.

- **Capítulo 5:** Conclusão

Por fim, serão apresentadas as conclusões, refletindo sobre a experiência do estágio como um todo, o seu impacto no desenvolvimento pessoal e profissional, e destacando os principais conhecimentos e contribuições.

Capítulo 2

Contexto do estágio

O presente capítulo oferece uma visão abrangente do contexto em que se inseriu o estágio realizado na omnium:retail. Dividido em três partes fundamentais, este capítulo explora detalhadamente a empresa, delinea o plano estratégico do estágio e apresenta o estado da arte relacionado ao ambiente de trabalho.

2.1. omnium:retail

A omnium:retail(Figura 1) é uma empresa líder no setor do retalho, especializada em soluções tecnológicas para o comércio, com o objetivo de fornecer soluções inovadoras e eficientes para otimizar as operações de retalho e melhorar a experiência do cliente.

“TRAZEMOS DÉCADAS DE EXPERIÊNCIA NO SETOR DO RETALHO PARA UMA PLATAFORMA SAAS OMNI-CHANNEL TOTALMENTE INTEGRADA, CAPAZ DE ELEVAR O SEU NEGÓCIO PARA A ERA BASEADA EM DADOS. O NOSSO PRODUTO É UMA PLATAFORMA MODERNA DE GESTÃO DE INTEGRAÇÕES QUE NOS PERMITE NÃO SER APENAS UMA SOLUÇÃO GENÉRICA PARA TODOS, MAS SIM TER UMA ABORDAGEM PERSONALIZADA PARA OS REQUISITOS DE INTEGRAÇÃO DO CLIENTE, CAPACITANDO O SEU NEGÓCIO COM UMA SOLUÇÃO DE GESTÃO OMNICANAL SEM INTERRUPÇÕES. A OMNIUM TORNA FÁCIL APROVEITAR OS SEUS DADOS DE ACORDO COM AS SUAS NECESSIDADES.” [1]

The logo for omnium:retail features the word "omnium" in a bold, lowercase, blue sans-serif font, followed by "retail" in a lighter weight of the same font. A colon is placed between the two words.

Figura 1: omnium:retail

Durante o estágio na omnium:retail, tive a oportunidade de vivenciar um ambiente dinâmico e desafiador. Esta experiência proporcionou um enriquecimento profissional e contribuiu para o desenvolvimento de competências essenciais na área da tecnologia e do comércio.

2.2. Plano de Estágio

No âmbito do estágio, foi elaborado um Plano que serviu como guia para as atividades realizadas. O plano foi estruturado em diferentes etapas, cada uma delas com objetivos específicos a serem alcançados.

- T1: Introdução à framework interna, plataformas a utilizar e boas práticas de desenvolvimento, incluindo formação nas ferramentas utilizadas.
- T2: Especificação/Levantamento de requisitos.
- T3: Estado de Arte, avaliação comparativa de soluções e ferramentas de desenvolvimento.
- T4: Desenho de solução e Planeamento.
- T5: Desenvolvimento do projeto.
- T6: Testes Unitários e de qualidade.
- T7: Fecho e apresentação.

Estas etapas, delineadas no plano de estágio, proporcionaram uma estrutura organizada e direcionada para as atividades desenvolvidas, permitindo um acompanhamento adequado do progresso e a obtenção dos resultados esperados. Apesar de contratempos o plano foi executado como planeado.

ID Actividade	2022		2023					
	Novembro	Dezembro	Janeiro	Fevereiro	Março	Abril	Maio	Junho
T1 - Formação								
T2 - Análise								
T3 - Estado de Arte								
T4 - Desenho Solução								
T5 - Desenvolvimento								
T6 - Testes								
T7 - Documentação e Apresentação								

Figura 2: Plano

2.3. Estado da Arte

No contexto deste estágio, a exploração do estado da arte torna-se essencial para contextualizar e fundamentar as práticas e tecnologias implementadas. Esta seção visa fornecer uma visão abrangente das tendências, ferramentas e métodos predominantes que moldam a paisagem do desenvolvimento Backend, concentrando-se especialmente na integração de micro-serviços e manipulação de dados no setor do retalho.

2.3.1. Setor do Retalho

O setor do retalho [2]Figura 3) compreende uma ampla gama de atividades comerciais, incluindo a venda de produto direto ao consumidor. Isso pode ocorrer em lojas físicas, online e por meio de outros canais de vendas. O retalho desempenha um papel fundamental na economia, pois abrange desde pequenos estabelecimentos locais até grandes cadeias de lojas. No contexto da omnium:retail, o retalho envolve a venda de produtos por meio de soluções tecnológicas avançadas.

Principais Desafios do Setor do Retalho:

O setor do retalho enfrenta diversos desafios, alguns dos quais a omnium:retail procura enfrentar por meio de soluções tecnológicas. Alguns dos principais desafios incluem:

- **Gestão de Stock:** Manter um controlo preciso do stock é essencial para evitar escassez ou excesso de produtos. Isso requer sistemas eficientes de gestão de stock e logística.
- **Experiência do Cliente:** A satisfação do cliente desempenha um papel fundamental no sucesso do retalho. Oferecer uma experiência de compra satisfatória é crucial para reter clientes e conquistar novos.
- **Concorrência:** A concorrência é intensa no setor do retalho. As empresas devem encontrar maneiras de se destacar e atrair consumidores em um mercado saturado.
- **Mudanças nas Preferências dos Consumidores:** As preferências dos consumidores estão em constante evolução segundo as tendências, e as empresas precisam se adaptar rapidamente a essas mudanças.

Beneficiários da Otimização e Aplicações no Setor do Retalho:

A otimização e a aplicação de soluções tecnológicas têm um impacto significativo no setor do retalho. Diversos grupos se beneficiam dessas melhorias, incluindo:

- **Empresas de Retalho:** As empresas de retalho, como as que colaboram com a omnium:retail(Figura 3), podem melhorar a sua eficiência operacional, aumentar o seu lucro e oferecer uma experiência mais atraente aos clientes por meio da implementação de soluções tecnológicas avançadas.
- **Consumidores:** Os consumidores beneficiam de uma experiência de compra mais conveniente e personalizada, bem como acesso a produtos e serviços inovadores.
- **Funcionários:** Os funcionários do setor de retalho podem beneficiar da automação de tarefas repetitivas, permitindo que se concentrem em atividades de maior valor agregado.
- **Economia Global:** A otimização no setor do retalho pode contribuir para o crescimento da economia global, impulsionando o comércio e a inovação.

Neste contexto, a minha experiência como estagiário de desenvolvimento Backend na omnium:retail, permitiu-me contribuir para a construção de soluções que visam solucionar problemas enfrentados por empresas de retalho.

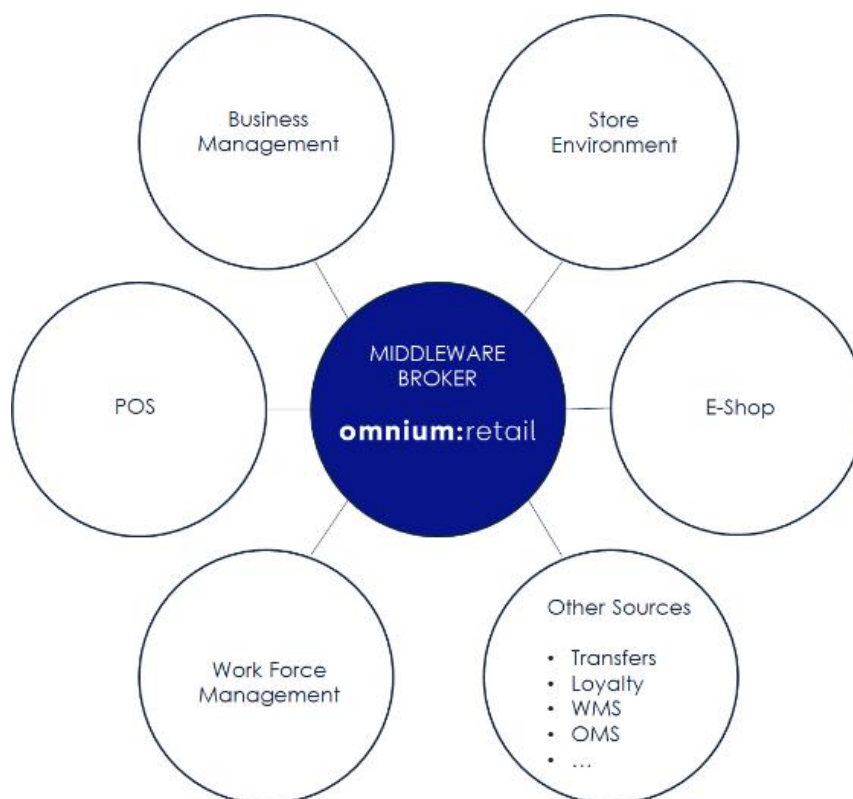


Figura 3: Retalho

2.3.2. Tecnologias e Frameworks

Nesta subsecção, serão exploradas as principais tecnologias e frameworks que desempenham um papel crucial no desenvolvimento Backend, conforme observado durante o estágio. Destaque será dado às seguintes ferramentas:

2.3.2.1. Swagger

Durante o estágio, a ferramenta Swagger foi utilizada para expor de maneira clara e interativa as funcionalidades dos micro-serviços desenvolvidos para a área de retalho. O Swagger, agora parte do projeto OpenAPI, é uma ferramenta essencial para documentar APIs, proporcionando uma interface interativa que simplifica a compreensão e teste de operações diretamente do navegador.

O Swagger vem com as seguintes Funcionalidades e Vantagens:

- Documentação automática baseada em atributos e comentários do código-fonte.
- Interface Swagger UI para teste interativo de endpoints.
- Facilitação da comunicação entre equipas por meio de uma documentação centralizada.
- Geração automática de código cliente em várias linguagens de programação.

A criação de endpoints, um dos aspetos essenciais no desenvolvimento de micro-serviços, foi simplificada pelo Swagger. Este processo envolve a anotação de atributos e comentários diretamente no código-fonte, possibilitando gerar automaticamente uma documentação abrangente. Cada endpoint é detalhadamente descrito, incluindo parâmetros esperados, tipos de dados, e exemplos práticos de requisições e respostas, como se pode visualizar na Figura 4 seguinte.

```
/// <summary>
/// Get Integration Requests
/// </summary>
/// <param name="pPage">Page number</param>
/// <param name="pRecords">Number of records per page.</param>
/// <param name="pIntegrationType">Filter by Integration Type</param>
/// <returns>Returns a list of integration requests.</returns>
/// <response code="200">Returns a list of integration requests.</response>
/// <response code="400">Returns an error explaining why the payload is invalid.</response>
/// <response code="500">Returns an error explaining why the server wasn't able to process the request.</response>
[SwaggerOperation(Description = "Retrieves a paginated list of integration requests.")]
[HttpGet]
[ProducesResponseType(typeof(CPRDPRSIntegrationRequestResponseList), StatusCodes.Status200OK)]
[ProducesResponseType(typeof(Error), StatusCodes.Status400BadRequest)]
[ProducesResponseType(typeof(Error), StatusCodes.Status500InternalServerError)]
```

Figura 4: Swagger documentation

O Swagger não apenas automatiza a documentação, mas também oferece uma interface Swagger UI intuitiva. Esta interface permite não apenas visualizar a documentação, mas também realizar testes interativos de endpoints diretamente no browser, facilitando a validação e debugging das operações. Além disso, o Swagger/OpenAPI oferece flexibilidade nos formatos de exportação, sendo possível gerar documentação em JSON e Web. O formato JSON é ideal para integração automatizada em pipelines de desenvolvimento, enquanto a versão Web é valiosa para interação humana e visualização facilitada. A versatilidade proporciona benefícios adaptados a diferentes cenários de uso, desde automação de processos até revisões manuais detalhadas.

No contexto do Estágio e desenvolvimento de micro-serviços para o setor do retalho, o Swagger desempenhou um papel crucial, não apenas na documentação dos micro-serviços de forma abrangente, mas também desempenhou um papel crucial na agilidade do desenvolvimento e na garantia da qualidade das operações de API.

2.3.2.2. Entity Framework

Entity Framework é uma framework ORM (Object-Relational Mapper) que simplifica o acesso e manipulação de dados em base de dados relacionais a partir de aplicações em C#. No contexto dos micro-serviços para o setor de retalho, Entity Framework foi utilizado para estabelecer a conexão com as bases de dados, simplificando operações de leitura e gravação de dados e proporcionando uma interface eficiente entre a lógica da aplicação e a persistência de dados em bases de dados relacionais, em particular, utilizando PostgreSQL [3]Dbeaver [4]nos projetos deste estágio.

Funcionalidades e Vantagens:

- Mapeamento objeto-relacional simplificado, permitindo que entidades C# representem diretamente tabelas de base de dados.
- Facilidade na execução de operações CRUD (Create, Read, Update, Delete) sem a necessidade de escrever queries manualmente.
- Suporte a diferente base de dados, incluindo PostgreSQL que foi usada nos projetos deste estágio, garantindo flexibilidade na escolha da solução de armazenamento.

Para usar esta framework é preciso entender a diferença entre o modelo orientado a objetos (OO) usado na programação e o modelo relacional utilizado nas bases de dados. Enquanto o modelo OO organiza dados em objetos com propriedades e comportamentos, o modelo relacional adota uma abordagem tabular, representando dados em tabelas inter-relacionadas.

Usando esta framework permite nos automatizar o processo de conversão/adaptação de objetos para relacional por meio de um sistema de mapeamento. Esse processo envolve a definição de classes de entidade em C# que representam os objetos da aplicação, e o Entity Framework cuida da tradução dessas entidades para as tabelas correspondentes nas bases de dados. Esse mapeamento é configurável e permite ajustes específicos, garantindo uma modelagem flexível e adaptável.

Desta forma a adoção do Entity Framework atua como uma ponte, traduzindo automaticamente conceitos OO para estruturas relacionais, simplificando o desenvolvimento e mantendo a coesão entre as camadas da aplicação.

2.3.2.3. RabbitMQ

O RabbitMQ é um sistema de código aberto que atua como um intermediário de mensagens, permitindo a comunicação assíncrona entre diferentes partes de um sistema distribuído. No contexto dos micro-serviços para o setor de retalho, o RabbitMQ foi adotado para coordenar a comunicação eficaz entre diferentes serviços, possibilitando uma arquitetura distribuída.

Funcionalidades e Vantagens:

- Facilitação da comunicação assíncrona, permitindo que diferentes partes da aplicação troquem mensagens de forma eficiente.
- Desacoplamento de componentes, possibilitando que serviços evoluam independentemente uns dos outros.
- Suporte a padrões de mensageria, como filas e trocas, oferecendo flexibilidade na implementação de diferentes cenários de comunicação.

A importância do RabbitMQ, reside na capacidade de lidar com a complexidade inerente dos sistemas distribuídos modernos, essa capacidade é atingida através de Message queues que introduzem uma abordagem eficiente para a comunicação assíncrona em sistemas distribuídos e o modelo publish-subscribe que proporciona uma dinâmica flexível, permitindo troca a de mensagens como representado na Figura 5.

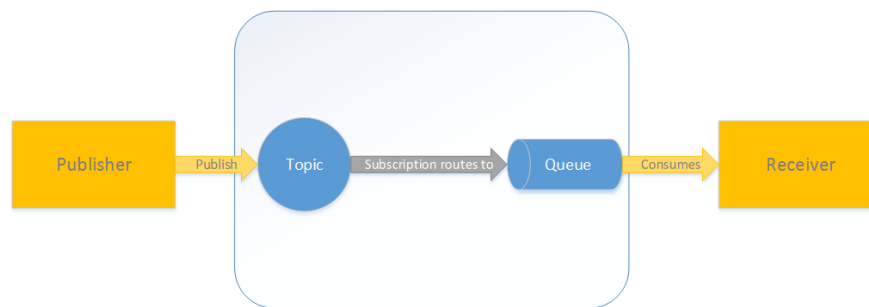


Figura 5: Publish/Subscribe Messaging

Message Queues:

As message queues, ou filas de mensagens, representam um paradigma em que mensagens são armazenadas e transmitidas de forma assíncrona entre os componentes de um sistema distribuído. Elas funcionam como intermediários que permitem a comunicação eficiente entre serviços.

Num cenário de sistemas distribuídos, as message queues funcionam como intermediários que armazenam e transmitem mensagens entre os diferentes componentes do sistema de forma assíncrona. Essa capacidade de armazenar mensagens temporariamente antes que sejam processadas pelos consumidores oferece diversas vantagens:

- **Independência de Componentes:** Ao utilizar filas de mensagens, os componentes do sistema tornam-se independentes uns dos outros. Isso significa que um serviço pode continuar operando mesmo se outro estiver temporariamente indisponível, resultando em uma arquitetura mais resiliente e adaptável.
- **Comunicação Eficiente:** A comunicação assíncrona permite que diferentes partes da aplicação troquem mensagens de maneira eficiente. Os produtores de mensagens (serviços que enviam mensagens para a fila) não precisam aguardar uma resposta imediata, o que acelera o fluxo de operações no sistema distribuído.
- **Escalabilidade:** As filas de mensagens possibilitam a escalabilidade horizontal, onde novas instâncias de um serviço podem ser facilmente adicionadas para lidar com maior carga de trabalho. Isso é fundamental em ambientes dinâmicos onde a demanda pode variar significativamente.

Modelo Publish-Subscribe:

O RabbitMQ suporta o modelo publish-subscribe, um padrão de comunicação em que um emissor (publisher) envia mensagens para um tópico (exchange), e os consumidores (subscribers) que estão interessados nesse tópico recebem as mensagens. Esse modelo

proporciona escalabilidade e flexibilidade na distribuição de informações, permitindo que diferentes partes do sistema recebam apenas as mensagens relevantes para as operações relevantes.

Benefícios do Modelo Publish-Subscribe:

- **Independência:** Permite que publishers e subscribers operem de forma independente. Os publishers não precisam conhecer os detalhes sobre quem consome as mensagens, proporcionando flexibilidade e evolução independente dos serviços.
- **Escalabilidade:** Facilita a escalabilidade horizontal. Novos subscribers podem ser adicionados conforme necessário para lidar com diferentes volumes de mensagens ou requisitos específicos.
- **Flexibilidade na Distribuição de Informações:** Subscribers podem se inscrever em tópicos específicos que correspondam às suas necessidades. Isso garante que cada componente receba apenas as mensagens relevantes, evitando a sobrecarga de dados desnecessários.
- **Padrão Dinâmico:** O modelo Publish-Subscribe oferece uma dinâmica flexível. Novos tópicos podem ser introduzidos conforme necessário, permitindo a evolução contínua da arquitetura.

No contexto do setor de retalho, o RabbitMQ foi aplicado na troca de mensagens relacionadas a eventos de consumo e integração de dados. Por exemplo, quando um produto é adicionado ou atualizado, uma mensagem é publicada no tópico apropriado. Os serviços interessados, como os encarregados da gestão de pedidos ou atualização de catálogo, podem consumir essas mensagens de forma assíncrona, coordenando tarefas assíncronas entre os diversos componentes do sistema. A dinâmica flexível e eficiente exemplifica a contribuição significativa do RabbitMQ para a arquitetura distribuída no setor de retalho.

2.3.3. Micro-Serviços e a sua Evolução

Micro-serviços [5] [6] são unidades de software independentes que realizam tarefas específicas em um sistema distribuído. Cada micro-serviço opera de forma autónoma e comunica com outros serviços por meio de interfaces bem definidas. Esta abordagem permite que cada serviço seja desenvolvido, implementado e dimensionado independentemente, proporcionando flexibilidade, escalabilidade e facilidade de manutenção. A evolução dos micro-serviços representa uma mudança paradigmática na arquitetura de software, onde sistemas monolíticos são substituídos por conjuntos de serviços independentes e especializados.

A coordenação entre micro-serviços pode ser realizada por meio de Dois paradigmas essenciais para coordenar esses micro-serviços são a Coreografia Descentralizada e a Orquestração Centralizada.

Coreografia Descentralizada:

Definição: Neste modelo, não há um ponto central que coordena as interações entre os micro-serviços. Cada serviço interage diretamente com os outros, respondendo a eventos e trocando mensagens.

Benefícios:

- Menos Acoplamento: Os serviços são menos dependentes uns dos outros, promovendo uma maior autonomia.
- Evolução Flexível: Cada serviço pode evoluir independentemente, adaptando-se às mudanças sem afetar outros serviços.

Desafios:

- Maior Complexidade de Rastreamento: A compreensão da lógica de negócios pode se tornar mais complexa devido à distribuição da coordenação.

Orquestração Centralizada:

Definição: Um componente central, frequentemente chamado de orquestrador, coordena as interações entre os micro-serviços. Ele gere o fluxo de trabalho e a execução das operações.

Benefícios:

- Visão Holística: Facilita a visão completa do fluxo de trabalho, proporcionando controle centralizado.
- Coordenação Simplificada: A lógica de coordenação está concentrada no orquestrador, tornando-a mais clara.

Desafios:

- Maior Acoplamento: Os micro-serviços podem se tornar mais dependentes do orquestrador, aumentando o acoplamento.
- Menos Autonomia: Alterações em um serviço podem impactar diretamente o orquestrador e, por extensão, outros serviços.

Durante o estágio na omnium:retail, a abordagem escolhida foi a Coreografia Descentralizada. Isso se alinha com os princípios de flexibilidade e escalabilidade dos micro-serviços,

permitindo que cada serviço se comunique diretamente com outros, promovendo autonomia e evolução independente. A complexidade adicional de rastreamento foi gerida através de boas práticas de documentação, como a integração de Swagger/API REST, facilitando a compreensão e interação entre os diversos componentes.

Benefícios e Desafios dos Micro-serviços

O deploy de micro-serviços pode ser realizado de forma isolada, permitindo a atualização independente de cada serviço. Ferramentas como Docker [7] e Kubernetes são muitas vezes utilizadas para facilitar o empacotamento e a implantação eficiente desses serviços.

Benefícios e Desafios da Arquitetura Micro-Serviço:

Benefícios:

- **Flexibilidade e Escalabilidade:** Os micro-serviços proporcionam flexibilidade no desenvolvimento, implementação e escalabilidade independente de cada serviço. A integração de Swagger/API REST com uma Message Queue, como o RabbitMQ, proporciona uma camada adicional de flexibilidade.
- **Facilidade de Manutenção:** A independência entre serviços facilita atualizações e manutenção, sem impacto em outros componentes do sistema. Os endpoints documentados no Swagger podem servir como referência para a troca assíncrona de mensagens entre serviços, facilitando a compreensão e interação entre os diversos componentes.

Desafios:

- **Complexidade de Gestão:** A gestão de múltiplos serviços pode aumentar a complexidade operacional.
- **Coordenação entre Serviços:** Garantir uma coordenação eficiente entre serviços pode ser um desafio, especialmente em sistemas distribuídos.

Um exemplo emblemático é a transição para micro-serviços no ambiente da **Amazon** que anteriormente operava com um sistema monolítico único, onde todas as funcionalidades estavam integradas em um único código. No entanto, para atender à crescente complexidade e demandas do negócio, eles adotaram a arquitetura de micro-serviços. Agora, cada funcionalidade, como recomendações de produtos, gerência de carrinho de compras e processamento de pedidos, é um micro-serviço independente.

Durante o estágio, a principal responsabilidade foi implementação de três micro-serviços no contexto da empresa omnium:retail. Esses micro-serviços foram projetados para desempenhar funções específicas de consumo e integração de dados. A implementação destes micro-serviços durante o estágio demonstrou a aplicação prática dos conceitos de micro-serviços na integração de dados, que se adequa a uma arquitetura ágil e adaptativa, alinhada com as demandas do setor do retalho.

2.3.4. Metodologias

Durante o estágio na omnium:retail, a aplicação de metodologias de desenvolvimento destacou-se pela adoção da abordagem Scrum [8](**Erro! A origem da referência não foi encontrada.**). A integração em equipas multidisciplinares, reuniões diárias, definição de metas em sprints e a utilização da ferramenta Jira [9] para gestão de tarefas foram práticas essenciais. A metodologia Scrum proporcionou flexibilidade no desenvolvimento, adaptando-se de forma dinâmica às demandas do projeto. A utilização do Jira permitiu a gestão interativa e incremental do desenvolvimento de software, possibilitando ajustes rápidos em resposta a requisitos em constante evolução.

Apesar de a metodologia Scrum ser a principal usada no contexto deste estágio existem outras metodologias todas com as suas vantagens distintas, dependendo das necessidades e características específicas de um projeto ou equipa de desenvolvimento.

Algumas dessas metodologias são:

Agile [10]:

Princípios Centrais:

- Colaboração com o cliente ao longo do desenvolvimento.
- Resposta rápida as mudanças nos requisitos.
- Entregas frequentes e incrementais.

Benefícios:

- Adaptabilidade a mudanças.
- Maior satisfação do cliente.
- Desenvolvimento iterativo e incremental.

Scrum:

Princípios Centrais:

- Sprints de desenvolvimento com duração fixa.
- Papéis definidos (Product Owner, Scrum Master, Team).
- Reuniões diárias para acompanhamento do progresso.

Benefícios:

- Transparência no progresso do projeto.
- Entregas regulares e previsíveis.
- Melhoria contínua através de retrospectivas.

Kanban:

Princípios Centrais:

- Fluxo contínuo de trabalho.
- Limites de trabalho em progresso (WIP).
- Foco na eficiência e redução de desperdício.

Benefícios:

- Visibilidade clara do trabalho em andamento.
- Adaptação dinâmica às mudanças nas prioridades.
- Identificação rápida de gargalos.

Resumindo a abordagem Scrum(Figura 6) permitiu uma maior flexibilidade e adaptabilidade às necessidades do projeto, promovendo a entrega de valor de forma incremental e a melhoria contínua dos processos de desenvolvimento.

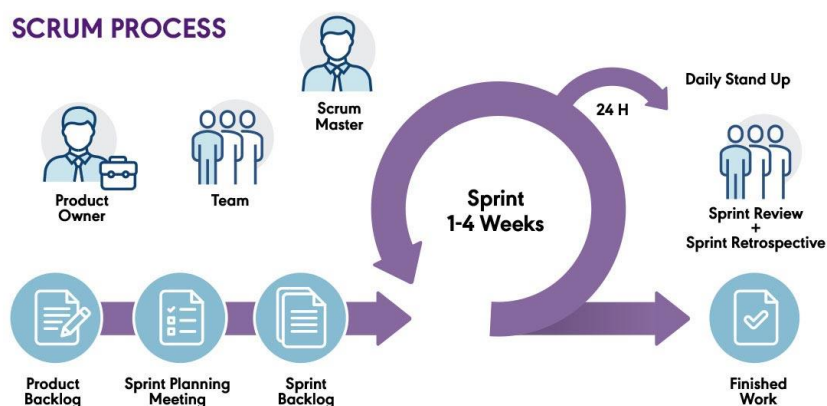


Figura 6: Scrum

Capítulo 3 Desenvolvimento e interações do Estágio

Neste capítulo, abordarei a minha experiência durante o estágio na omnium:retail. Explorarei as interações e o trabalho em equipa que tive a oportunidade de participar, bem como a estrutura diária que seguíamos para garantir a produtividade e o progresso dos projetos. Através deste capítulo, será possível compreender melhor a dinâmica do meu estágio e como contribuíram para o meu crescimento e desenvolvimento profissional.

3.1. Formações

Durante o meu estágio na omnium:retail, participei ativamente em várias formações e treinos que desempenharam um papel crucial no meu desenvolvimento profissional, abrangendo uma variedade de áreas desde boas práticas da empresa até tópicos avançados de programação e tecnologias específicas.

- 1. Boas Práticas da Empresa:** Uma das primeiras formações concentrou-se nas boas práticas da empresa, proporcionando-me conhecimentos sobre os princípios e diretrizes que orientam o desenvolvimento de software na omnium:retail. Essas boas práticas incluíram aspetos como padronização de código, organização de projetos, arquitetura de software e gestão de dependências. Esses conhecimentos foram aplicados diretamente nos projetos em que estive envolvido, assegurando a qualidade e a manutenibilidade do código.
- 2. C# for Beginners:** Participei do curso "C# for Beginners," uma formação online que forneceu uma compreensão abrangente dos fundamentos de C#. Essa formação foi fundamental para solidificar as bases em C# e estabelecer uma compreensão sólida dos conceitos iniciais da linguagem. Esta formação consiste em 19 vídeos no Youtube, todos eles de curta duração, mas claros e diretos ao ponto, tornando os leves e uma boa forma de relembrar e consolidar o básico da indentação do C#. Todo este curso está disponível através deste link: <http://tiny.cc/fhkfvz>.

- 3. Basics of Object-Oriented Programming with C#:** Outra formação online na qual estive envolvido foi "Basics of Object-Oriented Programming with C#." Este curso focou em conceitos cruciais como herança, polimorfismo e encapsulamento, o que me ajudou a aprofundar os meus conhecimentos na programação orientada a objetos, fornecendo insights valiosos sobre a construção e organização de sistemas baseados em objetos. Este curso, disponível na plataforma Udemy, é dividido em sete secções de vídeos que exploram diversos tópicos introdutórios da programação em C#, integrados com os princípios da programação orientada a objetos. Nas secções finais, 6 e 7, existe uma transição da teoria para a prática por meio de um projeto guiado envolvente. Neste projeto, os participantes são desafiados a criar um jogo simples de batalha, utilizando integralmente os conceitos aprendidos. Apesar da simplicidade do projeto, o mesmo revela uma ferramenta eficaz para consolidar o conhecimento, proporcionando uma experiência prática e envolvente. O tema cativante do projeto ajudou na realização do mesmo, motivando-me a dedicar um esforço adicional ao seu desenvolvimento. Este curso está disponível através deste link: "<http://tiny.cc/ijkfvz>".
- 4. Getting Started with Entity Framework:** Participei da formação "Getting Started with Entity Framework," disponível no canal oficial "dotnet" no YouTube. Esta série de vídeos abordou conceitos fundamentais do Entity Framework, incluindo mapeamento objeto-relacional (ORM), migrações e interação com base de dados relacionais e C#. Exemplos práticos facilitaram a compreensão desses conceitos. Foram 6 vídeos, porém mais extensos que os anteriores. Este curso foi essencial no desenvolvimento dos projetos do estágio principalmente na manipulação de dados de base de dados relacionais. É possível aceder a integridade desta formação através deste link: <http://tiny.cc/2kkfvz>.
- 5. C# Intermediate: Classes, Interfaces and OOP:** O curso "C# Intermediate: Classes, Interfaces and OOP" foi uma etapa avançada onde aprofundei os meus conhecimentos em C# e explorei tópicos avançados, incluindo o uso eficaz de classes e interfaces. Os principais tópicos abordados neste curso foram:

- Como trabalhar com classes, construtores, campos, propriedades, métodos e indexadores em C#.
- Utilizar encapsulamento para reduzir o impacto de mudanças no código.
- Melhorar a robustez do seu código.
- Reutilizar código através de herança e composição.
- Compreender porque a composição é preferível à herança.
- Construir aplicações extensíveis utilizando polimorfismo.
- Desenvolver aplicações com acoplamento fraco, extensibilidade e testabilidade utilizando interfaces.

Este curso contribuiu significativamente para a minha capacidade de desenvolver código eficiente e orientado a objetos. Este curso apesar de ser pago pode ser acessado aqui: <http://tiny.cc/5lkfvz>.

- 6. .NET 5 REST API Tutorial:** Além das formações mencionadas, assisti ao tutorial ".NET 5 REST API", através deste tutorial foi possível aprender sobre a criação de APIs RESTful eficientes, enriquecendo a minha experiência em desenvolvimento com .NET. Este tutorial encontra-se no Youtube e é composto por dez vídeos porém todos eles compridos chegando a ultrapassar 1 hora de vídeo. O link para aceder é este: <http://tiny.cc/fmkfvz>.
- 7. Workshop MongoDB:** Por fim participei num treino interno focado em MongoDB [11], uma tecnologia de base de dados não relacional amplamente utilizada pela omnium:retail. O treino abrangeu desde a modelagem de dados no MongoDB até consultas avançadas e operações específicas desta base de dados. Esses conhecimentos foram aplicados em projetos que envolveram integração com o MongoDB, contribuindo para a eficácia na manipulação de dados não relacionais. Este treino foi realizado internamente pela omnium:retail e os colaboradores no escritório dos mesmos, não só eu como estagiário participei, mas sim toda a equipa de desenvolvimento também como a equipa de produto.

3.2. Acompanhamento Inicial

No início do meu estágio, tive a oportunidade de acompanhar e prestar suporte a vários projetos, o que me permitiu entender melhor o funcionamento da empresa. Durante este período estava encarregue de 4 funções.

Integração de Produtos para PIM:

No início do estágio, uma das minhas principais responsabilidades era garantir que os dados dos produtos estivessem em conformidade com os requisitos do PIM(Projctct Information Management) do cliente. Era-me fornecido uma lista com SKUs(Stock Keeping Unit) de produtos. O objetivo era executar manualmente as queries e procedimentos do fluxo de integração de produtos até a chegada do produto até a tabela final, assim era garantido a compatibilidade desses produtos com os processos internos da omnium:retail, permitindo que os produtos passassem pelos fluxos operacionais até atingirem a tabela final sem erros nem interrupções. Por fim gerava um CSV da tabela final e entregava.

Função de Alarmista e Monitorização:

Desempenhando o papel de "alarmista," monitorizei atentamente os fluxos operacionais da empresa. Utilizei uma ferramenta interna denominada "daily-report," um procedimento PostgreSQL executado diariamente. Este procedimento gerava uma tabela detalhada de erros encontrados em cada fluxo operacional da empresa, facilitando a rápida identificação e correção de possíveis problemas. Além desse procedimento, utilizava também um software denominado de Kibana, que é um software de visualização de dados, o que facilitava a monitorização dos fluxos. A minha função era diariamente consultar ambas as ferramentas e reportar qualquer anomalia de forma a tomar providências adequadas ao possível erro.

Resolução de Erros de Fluxo Relacionados com o Produto:

Quando surgiam erros em fluxos operacionais, a minha função consistia em identificar e corrigir esses problemas. Após consultar a tabela de erros gerada pelo "daily-report" de forma a identificar o tipo de erro específico associado a falha do fluxo, através do link do erro, conseguíamos aceder a documentação da empresa propriamente criada de

forma a resolver tal anomalia. Após a identificação do problema, começava por executar os procedimentos mencionados na documentação até chegar ao erro e dessa forma encontrar o procedimento específico onde o problema ocorria. Depois de localizado o ponto de falha começava por fazer debug a esse procedimento (comentado certos campos e executando) de forma a compreender a natureza do problema. Encontrado o procedimento e a condição que originam o problema naquele set de produtos, juntava a informação adquirida e informava o meu superior de forma a ele resolver o erro da forma mais correta.

Alteração do Procedimento "daily-report":

Por fim, recebi uma solicitação para realizar alterações no procedimento PostgreSQL mencionado anteriormente (daily-report). O objetivo era adicionar uma coluna adicional ao resultado desse procedimento. Até então, várias colunas eram geradas, no entanto, a coluna de contagem era acumulativa, o que causava um problema, se o erro não fosse resolvido no mesmo dia e ocorressem mais erros no dia seguinte, a contagem acumulava, e não teríamos nenhuma indicação de novos erros, a menos que comparássemos com a contagem do dia anterior. Na Figura 7 podemos verificar as mudanças.

Boa noite. A Equipa de Delivery e Infraestruturas apresenta o resumo d

Nr_Registrations	Flow	Currency	Date
00	Stock not Send	ALL	2023-06-16
00	New Images to Sitoo	SEK	2023-06-16
99	Produtos em Campanha		2023-06-16
00	Orders Integration Open	SEK	2023-06-16
0	EUR Items With Cost Price Zero or Null	EUR	2023-06-16
1526	GBP Items With Cost Price Zero or Null	GBP	2023-06-16

Boa noite. A Equipa de Delivery e Infraestruturas apresenta o resumo diário do

Daily_count	Nr_Registrations	Flow	Currency
0	00	Stock not Send	ALL
0	00	New Images to Sitoo	SEK
0	00	Orders Integration Open	SEK
0	0	EUR Items With Cost Price Zero or Null	EUR
0	1526	GBP Items With Cost Price Zero or Null	GBP

Figura 7: Antes e depois (Daily-report)

3.3. Interações e Trabalho em Equipe

Durante o meu estágio na omnium:retail, tive a oportunidade de interagir e colaborar com profissionais experientes e talentosos. O ambiente de trabalho favoreceu a troca de conhecimentos e a realização de atividades em equipa.

3.3.1. Colaboração em Equipe

Integrei-me inicialmente à equipa de delivery, composta por programadores, analistas e outros estagiários. Participamos em reuniões diárias, onde discutíamos o progresso dos projetos, compartilhávamos ideias e resolvíamos problemas em conjunto. Qualquer dúvida da minha parte, facilmente entrava em contacto com algum colega de trabalho mais experiente e sempre me disponibilizavam um pouco do seu tempo para me ajudar. Essas interações estimularam o trabalho colaborativo e a procura por soluções eficientes.

3.3.2. Comunicação com os Gestores e Mentores

Além da equipa, tinha sempre a oportunidade de interagir com gestores e mentores, que acompanhavam o meu progresso e forneciam orientações e feedbacks. Ocasionalmente também surgia a oportunidade de reunir com o chefe da empresa e discutir a direção dos projetos e do meu estágio. Essas interações foram essenciais para o meu crescimento profissional, permitindo-me compreender melhor as expectativas da empresa e alinhar o meu trabalho com os objetivos do projeto.

3.4. Estrutura Diária

Durante o estágio, seguíamos uma estrutura diária que proporcionava organização e produtividade. A rotina consistia em:

3.4.1. Reuniões e Acompanhamentos

Iniciávamos o dia com reuniões de acompanhamento via Teams [12], onde cada membro da equipa compartilhava o progresso das suas atividades, discutia desafios

encontrados e solicitava suporte, quando necessário. A cada duas semanas era realizada a reunião de sprint em que discutíamos as tarefas realizadas durante o sprint, os objetivos alcançados, o que correu bem e mal e por fim discutíamos e planeávamos o sprint seguinte. Após as reuniões, recebíamos as tarefas atribuídas para o sprint. Essas tarefas eram definidas em conjunto com a equipa, levando em consideração as prioridades do projeto. As metas diárias eram estabelecidas com base nas entregas esperadas e prazos definidos. Essas reuniões eram fundamentais para manter todos alinhados e garantir que o trabalho estivesse a progredir de acordo com o planeado.

3.4.2. Tarefas e Horas

Através do JIRA [9], conseguia aceder às minhas tarefas de cada sprint de forma a guiar-me durante o dia, após cada dia realizava a marcação das horas através do clockify que é uma app integrada com o JIRA que nos possibilita de forma simples adicionar as horas que foram necessárias para cada tarefa, na Figura 8 podemos ver um exemplo de algumas de muitas tarefas que me foram atribuídas ao longo do estágio.



Figura 8: Clockify tarefas

3.4.3. Revisões e Feedbacks

Após concluir as tarefas, testava e submetia o meu trabalho para revisão, sendo esta efetuada por alguém mais experiente. Essa etapa era essencial para garantir a qualidade do código e o alinhamento com as diretrizes da empresa. Receber feedbacks regulares permitiu-me aprimorar as minhas habilidades e melhorar continuamente o meu trabalho.

Capítulo 4 Projetos

Neste capítulo, abordarei a minha experiência durante o estágio na omnium:retail. Explorarei as interações e o trabalho em equipa que tive a oportunidade de participar, bem como a estrutura diária que seguíamos para garantir a produtividade e o progresso dos projetos. Através deste capítulo, será possível compreender melhor a dinâmica do meu estágio e como contribuíram para o meu crescimento e desenvolvimento profissional.

4.1. Projetos

Vamos agora abordar os principais projetos concretizados ao longo do estágio na omnium:retail, revelando o contexto, os desafios enfrentados e as soluções implementadas. A exploração aprofundada destes projetos oferece uma perspetiva prática das contribuições realizadas, destacando a aplicação direta das competências técnicas adquiridas durante a experiência profissional.

Cada projeto é abordado individualmente, fornecendo uma análise detalhada do seu propósito, do processo de desenvolvimento e das lições aprendidas ao longo do percurso. Esta secção é essencial para compreender a aplicação prática dos conhecimentos adquiridos no ambiente dinâmico e desafiador da omnium:retail.

4.1.1. Projeto X

Scope:

O projeto X teve como objetivo principal migrar produtos de uma plataforma de integração do cliente para o sistema de ponto de venda (POS). Os dados eram inicialmente armazenados em uma base relacional no ambiente do cliente, sendo necessária a criação de um Sistema de

Processamento de Integração (SPI) para realizar a extração, transformação (ETL) e integração desses produtos no POS.

Problema:

A demanda por migrar dados da plataforma de integração do cliente para o POS exigiu a criação do SPI. O desafio envolveu a extração dos produtos da base de dados Postgres do cliente, aplicação de transformações necessárias e integração com o sistema HPRD da omnium:retail para inclusão no POS.

Desenvolvimento:

Durante o projeto, trabalhei na criação do SPI utilizando a linguagem de programação C#. Foi necessário compreender a estrutura e o conteúdo da tabela na base de dados do cliente de forma a implementar e adaptar a extração adequada dos produtos. Inicialmente, foram realizados estudos sobre o formato dos dados no HUB da omnium:retail e fizemos um mapeamento manual numa folha Excel, convertendo o formato da tabela pricat para o formato do HUB. Em seguida, analisei um SPI já existente na omnium:retail para obter bases e conhecimentos necessários para a criação do novo SPI. Após revisão e aprovação do mapeamento, foi iniciado o processo de criação do SPI. Foi utilizado um modelo de projeto criado pela omnium:retail para projetos com base de dados PostgreSQL. Com base nesse modelo, comecei a criar a entidade dos produtos com o formato da tabela pricat. Também foi criado um modelo de resposta com o formato esperado pelo HUB e um repositório com a função de obter os produtos. Em seguida, desenvolvemos um adaptador para converter os produtos recebidos da tabela pricat para o formato da resposta. Por fim, um controller do Swagger foi criado para executar a função de obtenção dos produtos, aplicar o adaptador e gerar uma resposta com os dados desejados. Dessa forma, é possível executar o projeto HPRD que é o responsável pelo consumo e integração dos produtos, conectando-se ao SPI e realizando a integração dos produtos recebidos com base na resposta pelo GET fornecido pelo SPI. Com isso, alcançamos o objetivo de integrar os produtos do cliente na plataforma Sítio [13]. Esta integração foi crucial para garantir a disponibilidade e atualização dos produtos nos canais de venda físicos e digitais do cliente da omnium:retail. Podemos visualizar melhor o funcionamento deste micro-serviço na Figura 9.

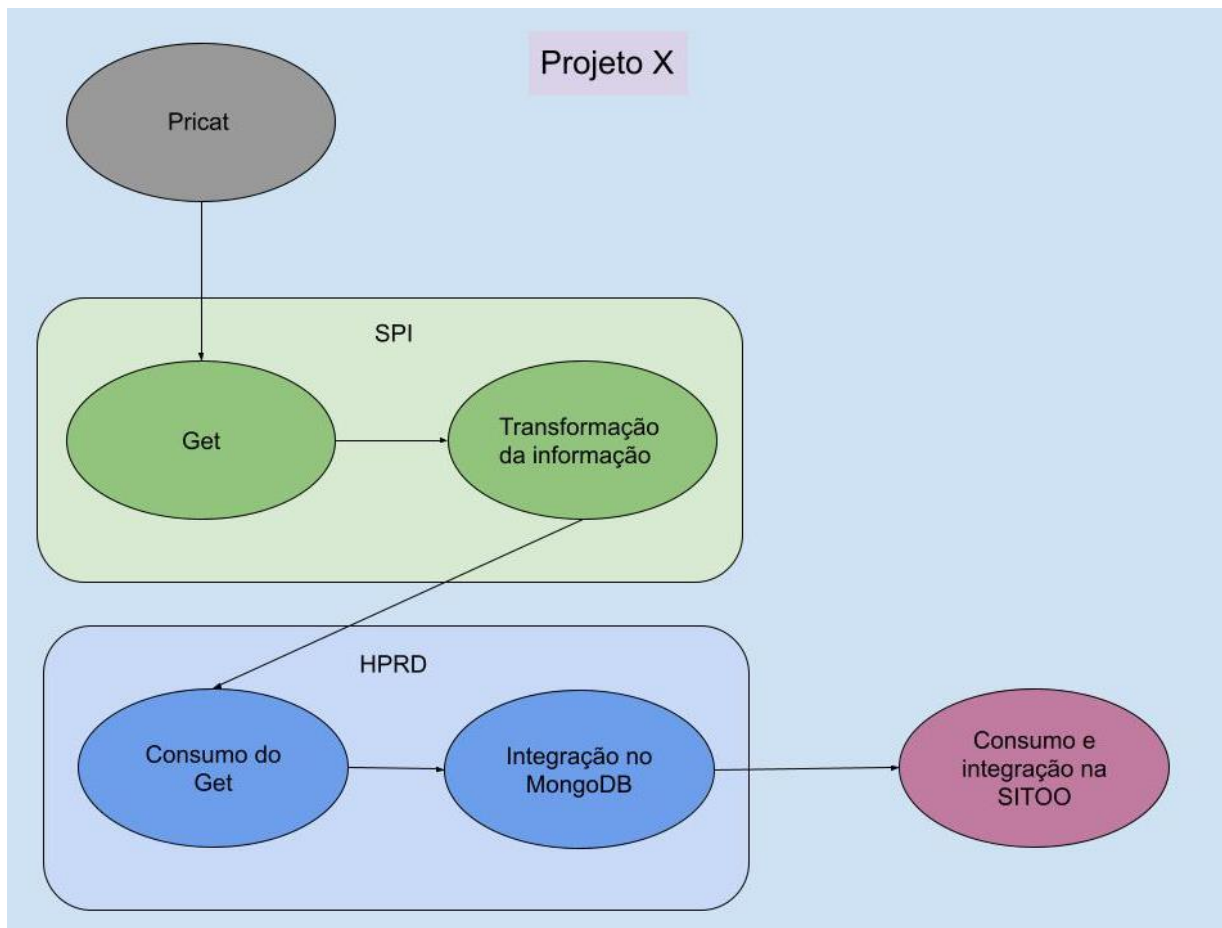


Figura 9: Fluxo do produto

Experiência Obtida:

Participar desse projeto proporcionou insights valiosos sobre o processamento de integração de dados. A análise cuidadosa, o desenvolvimento preciso e a integração eficiente foram fundamentais para garantir o bom funcionamento do sistema e a satisfação do cliente.

4.1.2. Conector HUB PostgreSQL

Scope:

Após a conclusão bem-sucedida do projeto X, surgiu a oportunidade de participar de um projeto intermediário essencial para toda a empresa, visando desenvolver um conector reverso. Este conector tinha o propósito de transferir dados do HUB da omnium:retail de volta para o PostgreSQL. A demanda por esse conector não se limitava ao projeto X, sendo relevante para diversos projetos em andamento na empresa.

Problema:

A necessidade de criar este processo surgiu da falta de experiência com manipulação de dados em base de dados não relacionais como MongoDB que é a tecnologia de base de dados usado no HUB da omnium:retail, sendo assim esta ferramenta permite qualquer colaborador transferir dados do HUB para uma base de dados PostgreSQL de forma poderem ser feitas qualquer alteração através de queries e procedimentos, e logo depois reenviar esses dados de volta para o HUB.

Desenvolvimento:

O projeto seguiu uma abordagem semelhante ao projeto X, combinando o uso do MongoDB e PostgreSQL. O adaptador foi aprimorado, convertendo campos que continham listas de valores em formato JSON para facilitar o transporte e preservar as informações relevantes. Foi criado um controlador Swagger, utilizando o método POST, para executar o processo de integração. A tabela no PostgreSQL foi estruturada com campos do tipo jsonb para lidar com listas de valores. Após os dados chegarem a essa tabela foram criados alguns procedimentos de forma a realizar as transformações necessárias de forma a obtermos o formato anteriormente usado em PostgreSQL. Resumindo esta ferramenta possibilita a importação de dados do HUB até ao PostgreSQL, o tratamento de dados é realizado nessa base de dados atendendo as necessidades específicas, por fim esses dados são de novo integrados no HUB da omnium:retail em MongoDB, este fluxo pode ser visualizado de forma mais facilitada na Figura 10. Esta implementação atendeu às demandas de integração de dados em vários projetos da empresa, consolidando e expandindo os conhecimentos adquiridos no projeto X.

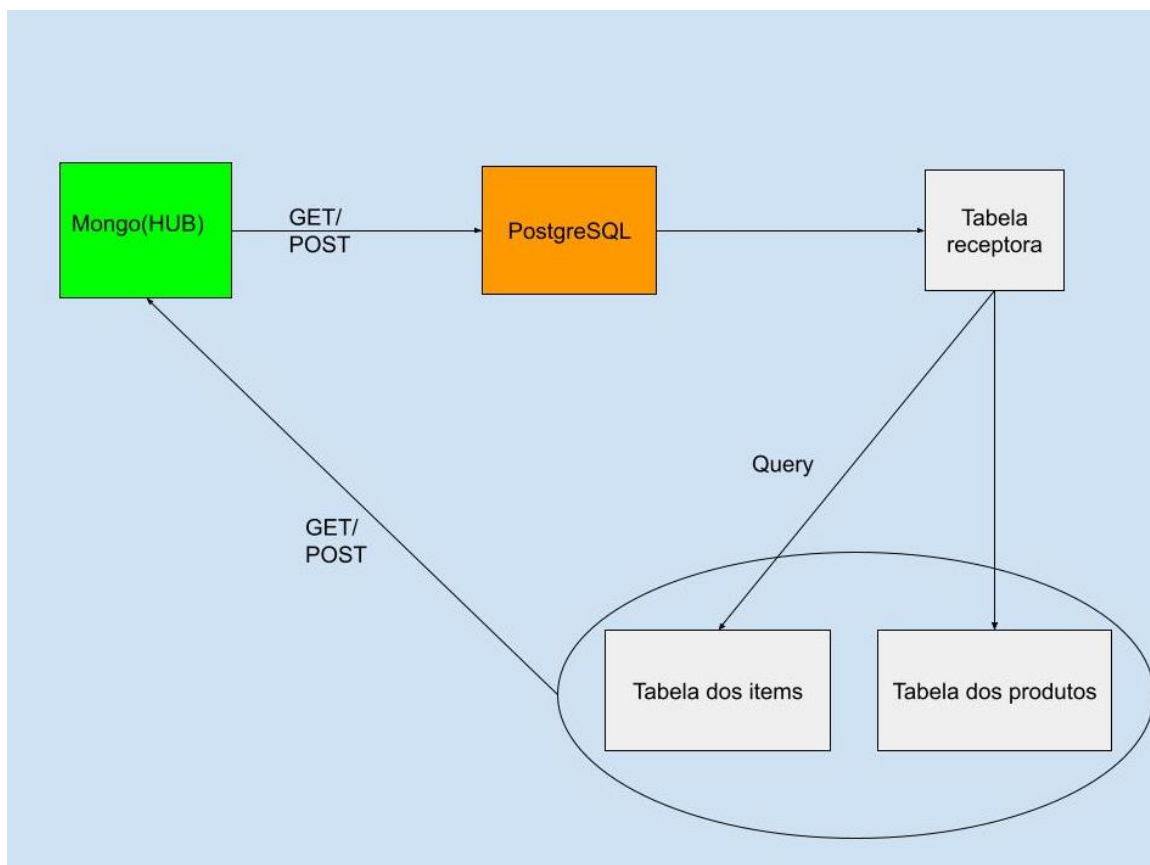


Figura 10: Fluxo dos dados

4.1.3. Projeto PrestaShop

PrestaShop:

O PrestaShop [14] (Figura 11), uma plataforma de e-commerce de código aberto, destaca-se como uma ferramenta versátil e eficaz no universo do retalho online. Lançado em 2007, tornou-se uma escolha popular para empresas de todos os tamanhos, oferecendo uma experiência de compra personalizável e funcionalidades essenciais para o sucesso no e-commerce.



Figura 11: PrestaShop

Contexto:

O projeto PrestaShop foi concebido para estabelecer um ambiente PrestaShop e desenvolver os processos de consumo e integração de produtos na plataforma de e-commerce PrestaShop. O objetivo final era permitir que a omnium:retail aceitasse futuramente clientes que utilizam o PrestaShop assim como aceita atualmente clientes que usem Shopify, ampliando a gama de plataformas suportadas. Como parte desse projeto, fui responsável por aprender e explorar o PrestaShop, estudar e estabelecer um ambiente de desenvolvimento do PrestaShop e desenvolver o consumidor (consumer) das categorias e fornecedores (suppliers) e implementar o processo de integração desses elementos.

Problema:

O principal desafio residia na falta de familiaridade da omnium:retail com a plataforma PrestaShop, exigindo uma abordagem de desenvolvimento desde o início. O projeto envolveu a criação de um ambiente local PrestaShop do zero, a compreensão da arquitetura, e o desenvolvimento de processos eficientes para consumir e integrar categorias e fornecedores.

Desenvolvimento:

Para iniciar, realizei uma análise detalhada do PrestaShop, compreendendo como opera e a arquitetura do mesmo. Esta etapa foi fundamental para garantir uma integração adequada e precisa entre os sistemas, a utilização da documentação [15] do Prestashop foi uma enorme ajuda nesta fase, oferecendo informação valiosa sobre a composição e formatação da API. Mapeei então os campos e dados do PrestaShop com os do HUB da omnium:retail para garantir uma integração precisa. Criar um ambiente de testes local do PrestaShop, usando Docker, foi essencial, pois a omnium:retail ainda não possuía um ambiente de desenvolvimento dedicado. Utilizei um Dockerfile para facilitar a instalação e configuração do ambiente local, incluindo a base de dados necessária e servidor da mesma, além disso a documentação detalhada foi elaborada para facilitar a utilização da imagem Docker por toda a equipa. Esta abordagem permitiu que eu e outros membros da empresa pudessem criar e testar o código de integração num ambiente controlado e independente.

Posteriormente, iniciei o desenvolvimento do consumidor PrestaShop. Esse componente extraía dados relevantes do MongoDB da omnium:retail e os enviava para o HUB do PrestaShop. Este processo incluiu consultas e manipulação de dados para atender aos requisitos específicos do PrestaShop. Implementei também um processo de solicitação de integração, permitindo a

solicitação a integração de produtos específicos do HUB do PrestaShop. Neste ponto vários problemas surgiram devido à conexão não existente com a API do PrestaShop, através de uma pesquisa, consegui descobrir uma biblioteca para C# que integra as funcionalidades da API do PrestaShop, de forma a poder aceder e integrar dados com facilidade. Esta biblioteca chama-se PrestaSharp [16]. Assim foi possível estabelecer conexão ao PrestaShop, também como realizar ações CRUD de forma fácil e eficaz.

Por fim, desenvolvi um controlador para expor as funcionalidades de integração do PrestaShop. O fluxo do processo é representado na Figura 12, mostrando um pedido de integração realizado por um ator. Após a solicitação, o HUB verifica novos produtos no POS, consome e manipula esses produtos para o formato do PrestaShop, publica uma mensagem no RabbitMQ e inicia o processo de integração, onde o PrestaShop consome os produtos e marca como integrados no HUB.

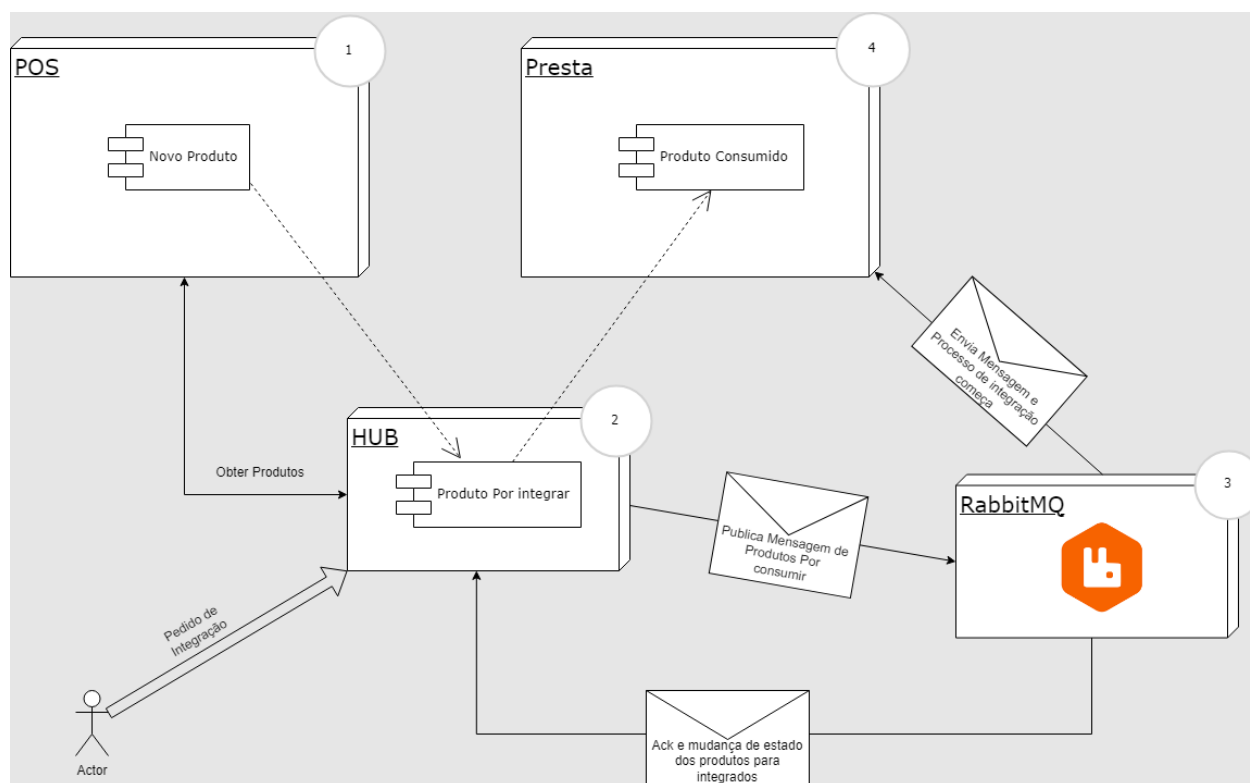


Figura 12: Fluxo de integração

Desta forma, concluí com sucesso o projeto de integração com o PrestaShop, permitindo que os clientes da omnium:retail pudessem utilizar esta popular plataforma de e-commerce de forma integrada e eficiente.

Experiência obtida:

Este projeto proporcionou uma valiosa experiência em lidar com desafios de integração em ambientes desconhecidos. A necessidade de compreender, mapear e implementar um ambiente PrestaShop do zero expandiu significativamente o conhecimento em integrações de diversas plataformas. A exploração detalhada do PrestaShop, a configuração do ambiente local com Docker e a integração eficiente entre o HUB da omnium:retail e o PrestaShop desenvolveram habilidades críticas para lidar com ambientes complexos e tecnologias diversas. Além disso, a criação de processos de consumo e integração, juntamente com a manipulação eficaz de dados para atender aos requisitos específicos do PrestaShop, contribuiu para uma compreensão mais profunda da importância da precisão e eficiência em projetos de integração de dados.

Capítulo 5 Conclusão

Chegamos ao capítulo final do relatório, onde irei resumir os pontos principais e destacar as realizações, lições aprendidas e impacto do estágio no meu futuro, realçando os desafios enfrentados e como os mesmos fizeram-me crescer e preparou-me melhor para o mercado de trabalho.

5.1. Reflexão e Resumo do Estágio

Durante o estágio na omnium:retail, participei ativamente em vários projetos, incluindo a elaboração de micro-serviços para gerir fluxos de informação entre canais de venda físicos e digitais, a migração de dados de uma plataforma de integração do cliente para o sistema de ponto de venda, a criação e elaboração de um conector HUB PostgreSQL e o projeto PrestaShop, foram as principais tarefas que me acompanharam ao longo deste estágio. As tecnologias utilizadas incluíram C#, .NET, MongoDB, PostgreSQL, Swagger, Entity Framework, RabbitMQ e a metodologia Scrum. A arquitetura de micro-serviços foi implementada, utilizando a abordagem de Coreografia Descentralizada, e o RabbitMQ foi utilizado para troca de mensagens relacionadas a eventos de consumo e integração de dados.

5.2. Conclusão

Durante a realização deste relatório tive a oportunidade de refletir na jornada que enfrentei durante estes 7 meses de estágio na omnium:retail. Foi uma oportunidade de aplicar os meus conhecimentos teóricos adquiridos ao longo destes meus 5 anos no IPB, além de conhecer melhor e superar este mundo que é o mercado de trabalho. Durante o estágio, vivenciei períodos de altos e baixos, mantendo sempre o foco na superação das dificuldades. Mesmo diante de problemas aparentemente sem solução, contei com a colaboração dos meus colegas de trabalho e com a minha determinação para superá-los, resultando no fortalecimento da minha confiança dos meus conhecimentos. Desde as formações até aos projetos finais e passando pelo acompanhamento e suporte, tudo me ajudou a crescer e a preparar-me para a etapa seguinte.

Além da minha experiência com o trabalho, adquiri experiência a trabalhar em equipa. Durante este estágio foram absorvidos vários conhecimentos, não só por mim, mas também através da

ajuda dos meus colegas de trabalho, que me ajudaram imenso a superar várias dificuldades com que me deparei.

Este estágio além de proporcionar-me um relatório de fim de mestrado, proporcionou-me experiência de trabalho, experiência social e também conexões e amigos. Isto mostrou-se importante na minha próxima fase de procura de trabalho que já se mostra uma adição valiosa no meu histórico profissional e académico, foi uma etapa crucial na minha formação e estou ansioso para aplicar as lições aprendidas na minha futura carreira profissional.

Foi uma jornada enriquecedora e bastante positiva, que me deixa com vontade de continuar e sem medo de enfrentar os desafios e oportunidades que o futuro me reserva.

Bibliografia

Bibliografia

- [1] “Success Path to Omni-Channel,” [Online]. Available: <https://omnium-retail.com/index.html>.
- [2] M. & W. B. A. Levy, “Retailing management,” em *Retailing management*, McGraw-Hill Education, 2012.
- [3] “PostgreSQL Introduction,” [Online]. Available: https://www.w3schools.com/postgresql/postgresql_intro.php.
- [4] “Dbeaver,” [Online]. Available: <https://dbeaver.io/about/>.
- [5] J. Smith, “The Art of Microservices: A Practical Guide to Microservice Architecture and Design,” em *Microservices: A Practical Guide*, O'Reilly Media, 2020.
- [6] S. Newman, “Building Microservices: Designing Fine-Grained Systems,” em *Building Microservices*, O'Reilly Media, 2015.
- [7] “Accelerated Container Application Development,” [Online]. Available: <https://www.docker.com/>.
- [8] K. & S. J. Schwaber, “The Scrum Guide,” 2017. [Online]. Available: [Scrum.org](https://www.scrum.org).
- [9] “Jira: Issue & Project Tracking Software,” [Online]. Available: <https://www.atlassian.com/software/jira>.
- [10] “What is Agile Software Development?,” [Online]. Available: <https://www.clearart.com/what-is-agile-software-development.html>.
- [11] [Online]. Available: https://www.w3schools.com/mongodb/mongodb_get_started.php.
- [12] Microsoft, “Video Conferencing, Meetings, Calling: Microsoft Teams,” Microsoft Teams, [Online]. Available: <https://www.microsoft.com/en-us/microsoft-teams/group-chat-software>.
- [13] “Sitoo® Point of Sale System bringing the future of retail to life,” [Online]. Available: <https://www.sitoo.com/>.
- [14] P. Team, “Prestashop,” [Online]. Available: <https://prestashop.com/>.
- [15] “PrestaShop Developer Documentation,” PrestaShop, [Online]. Available: <https://devdocs.prestashop-project.org>.
- [16] “Bukimedia/PrestaSharp: CSharp .Net client library for the PrestaShop API via web service,” GitHub, [Online]. Available:

<https://github.com/Bukimedia/PrestaSharp>.