



UNIVERSIDADE
LUSÓFONA

Pullu

Trabalho Final de curso

Relatório Intercalar 2º Semestre

Pedro Santiago, a22303437, Informática de Gestão

Orientador: Rui Santos

Departamento de Engenharia Informática da Universidade Lusófona

Centro Universitário de Lisboa

12 de Abril 2026

Direitos de cópia

Pullu, Copyright de Pedro Santiago, ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimientos

Resumo

No contexto atual do ecommerce, a Shopify afirma-se como a maior e mais utilizada plataforma para criação de lojas online, servindo milhões de lojistas em todo o mundo. Entre os modelos mais populares dentro do ecommerce, o dropshipping destaca-se por permitir que os vendedores comercializem produtos sem manter stock próprio, dependendo de fornecedores externos. A maioria destes produtos não são patenteados nem exclusivos, o que significa que estão legalmente disponíveis em diferentes plataformas, como Temu, AliExpress ou até outras lojas Shopify. Este modelo exige a testagem constante de novos produtos, levando muitos lojistas a importar informações dessas fontes externas. No entanto, transformar estes dados numa listagem completa dentro da Shopify continua a ser um processo manual e demorado: é necessário copiar descrições, traduzir textos, organizar variantes, editar imagens e ajustar vários detalhes técnicos. Esta repetição diária de tarefas cria um estrangulamento operacional que se torna cada vez mais evidente à medida que cresce o número de produtos testados.

A Pullu é uma SaaS (Software as a Service) baseada na web desenvolvida para simplificar a criação de listagens de produtos na Shopify. Os utilizadores podem iniciar sessão com email ou conta Google e aceder a uma dashboard intuitiva, onde podem inserir o link de um produto proveniente do AliExpress, Temu ou até de outra loja Shopify. A plataforma recolhe automaticamente todos os dados relevantes do produto, incluindo imagens, variantes e metadados, e permite que os utilizadores editem textos e imagens com recurso a inteligência artificial antes de importar a listagem para a Shopify. Assim, um processo que normalmente demoraria entre 20 a 45 minutos passa a demorar menos de um minuto, aumentando drasticamente a eficiência e libertando tempo para outras tarefas estratégicas no ecommerce.

A Pullu é construída com Next.js tanto para front-end como para back-end, utilizando Supabase para gestão de dados e autenticação, e integrando outras APIs como Shopify, Potoroom, ChatGPT e Fal.ai. Esta combinação tecnológica permite automatizar a recolha, edição e otimização de dados e imagens dos produtos, simplificando todo o processo de criação de listagens e aumentando significativamente a eficiência das operações de ecommerce.

Palavras-chave: SaaS, comércio eletrónico, dropshipping, importação de produtos, automação com AI, Shopify.

Abstract

In the current ecommerce landscape, Shopify stands out as the largest and most widely used platform for creating online stores, serving millions of merchants worldwide. Among the most popular business models in ecommerce, dropshipping is notable for allowing sellers to market products without holding their own stock, relying instead on external suppliers. Most of these products are neither patented nor exclusive, meaning they are legally available on different platforms such as Temu, AliExpress, or even other Shopify stores. This model requires constant testing of new products, leading many merchants to import information from these external sources. However, turning this data into a complete listing within Shopify remains a manual and time-consuming process: it is necessary to copy descriptions, translate texts, organize variants, edit images, and adjust various technical details. This daily repetition of tasks creates an operational bottleneck that becomes increasingly evident as the number of products tested grows.

Pullu is a web-based SaaS (Software as a Service) developed to simplify the creation of product listings on Shopify. Users can log in with their email or Google account and access an intuitive dashboard where they can insert the link of a product from AliExpress, Temu, or even another Shopify store. The platform automatically collects all relevant product data, including images, variants, and metadata, and allows users to edit texts and images using artificial intelligence before importing the listing to Shopify. Thus, a process that would normally take 20 to 45 minutes now takes less than a minute, drastically increasing efficiency and freeing up time for other strategic ecommerce tasks.

Pullu is built with Next.js for both front-end and back-end, using Supabase for data management and authentication, and integrating other APIs such as Shopify, Photoroom, ChatGPT, and Fal.ai. This technological combination allows the automation of data and image collection, editing, and optimization, simplifying the entire listing creation process and significantly boosting the efficiency of ecommerce operations.

Keywords: SaaS, ecommerce, dropshipping, product import, AI automation, Shopify.

Índice

1 Introdução	1
1.1 Enquadramento	1
1.2 Motivação e Identificação do Problema	3
1.3 Objetivos	3
1.4 Estrutura do Documento	4
2 Pertinência e Viabilidade	5
2.1 Pertinência	5
2.2 Viabilidade	9
2.3 Análise Comparativa com Soluções Existentes	10
2.3.1 Soluções existentes	10
2.3.2 Análise de benchmarking	11
2.4 Proposta de inovação e mais-valias	11
2.5 Identificação de oportunidade de negócio	12
3 Especificação e Modelação	13
3.1 Análise de Requisitos	13
3.1.1 Enumeração de Requisitos	13
3.1.2 Descrição detalhada dos requisitos principais	15
3.1.3 Casos de Uso/User Stories	16
1. Conexão e Sustentabilidade	16
2. Extração e Otimização da Listagem (Otimização de Tempo)	16
3. Publicação e Fluxo de Trabalho	16
3.2 Modelação	17
3.3 Protótipos de Interface	19
4 Solução Proposta	21
4.1 Apresentação	21
4.2 Arquitetura	23
4.3 Tecnologias e Ferramentas Utilizadas	25
4.4 Ambientes de Teste e de Produção	27
4.5 Abrangência	28
4.6 Componentes	29
4.6.1 Sistema de Autenticação e Gestão de Utilizadores	29
4.6.2 Sistema de Gestão de Créditos e Subscrições	30
4.7 Interfaces	35
5 Testes e Validação	39
6 Método e Planeamento	40
6.1 Planeamento inicial	40
6.2 Análise Crítica ao Planeamento	41
7 Resultados	42
7.1 Resultados dos Testes	42

7.2 Cumprimento de requisitos	42
8 Conclusão	43
8.1 Conclusão	43
8.2 Trabalhos Futuros	43
Bibliografia	44
Glossário	46

Lista de Figuras

Figura 1 – Experiência no sector de e-commerce (Inquérito).	5
Figura 2 – Número de produtos importados semanalmente (Inquérito).	5
Figura 3 – Tempo médio gasto na importação de um produto (Inquérito).	6
Figura 4 – Tarefas mais demoradas no processo de importação (Inquérito).	7
Figura 5 – Fontes de produtos utilizadas pelos inquiridos.	7
Figura 6 – Ferramentas de importação utilizadas.	8
Figura 7 – Nível de satisfação com o processo atual de importação.	8
Figura 8 – Modelo de Entidade-Relação da Base de Dados.	17
Figura 9 – Landing Page da Pullu (Protótipo).	18
Figura 10 – Dashboard de Importação de Produtos (Protótipo).	19
Figura 11 – Interface de Edição de Produto com AI (Protótipo).	20
Figura 12 – Fluxo de Dados Típico da Aplicação.	23
Figura 13 – Diagrama de Arquitetura Serverless.	24
Figura 14 – Landing Page (CTA e Inserção de Link).	34
Figura 15 – Fluxo de Valor (3 passos).	35
Figura 16 – Comparação Pullu vs Concorrentes.	35
Figura 17 – Secção de Pricing.	36
Figura 18 – Secção FAQ.	36
Figura 19 – Footer da Landing Page.	36
Figura 20 – Dashboard (Inserção de URL).	37
Figura 21 – Ecrã de Informações do Produto.	37

Lista de Tabelas

Tabela 1 – Análise comparativa entre Pullu, Kopy e Poky	11
Tabela 2 – Lista de requisitos	13

Lista de Siglas

AI – Artificial Intelligence / Inteligência Artificial.

API – Application Programming Interface / Interface de Programação de Aplicações.

CSS – Cascading Style Sheets.

HTML – HyperText Markup Language / Linguagem de Marcação de Hipertexto.

JS – JavaScript.

JWT – JSON Web Token.

LEI – Licenciatura em Engenharia Informática.

LIG – Licenciatura em Informática de Gestão.

MVP – Minimum Viable Product / Produto Mínimo Viável.

OAuth – Open Authorization (protocolo de autorização).

ODS – Objetivos de Desenvolvimento Sustentável.

RLS – Row Level Security.

SaaS – Software as a Service.

SEO – Search Engine Optimization.

SSR – Server-Side Rendering.

TFC – Trabalho Final de Curso.

UI – User Interface / Interface de Utilizador.

ULHT – Universidade Lusófona de Humanidades e Tecnologias.

1 Introdução

1.1 Enquadramento

O comércio eletrónico tem registado um crescimento exponencial nas últimas décadas, com vendas globais a ultrapassarem os 5,7 triliões de dólares em 2022, representando aproximadamente 20% de todas as vendas a retalho mundiais. Este crescimento democratizou o acesso ao mercado global, permitindo que empreendedores individuais e pequenas empresas estabeleçam presença digital com investimentos iniciais modestos. Neste contexto, a Shopify consolidou-se como uma das plataformas líderes de e-commerce, servindo mais de 4,4 milhões de lojas ativas em 175 países através de um modelo SaaS (Software-as-a-Service) que elimina barreiras técnicas e oferece ferramentas integradas de gestão de produtos, pagamentos e marketing.

O dropshipping emergiu como modelo de negócio predominante entre novos empreendedores digitais. Trata-se de um método de fulfillment onde o retalhista não mantém stock físico, transferindo pedidos diretamente para fornecedores que enviam ao cliente final. Este modelo elimina investimentos em inventário e infraestrutura logística. A maioria dos produtos comercializados são artigos não patenteados, produtos genéricos disponibilizados por múltiplos fornecedores em plataformas como AliExpress, Temu e Alibaba, onde a competitividade depende de marketing e velocidade de execução, não de exclusividade de produto.

O sucesso no dropshipping baseia-se na estratégia de "product testing", a testagem sistemática de múltiplos produtos para identificar "winners" rentáveis. Dados da indústria indicam que apenas 3-5% dos produtos testados alcançam rentabilidade consistente, criando necessidade de testar grandes volumes. Operações estabelecidas importam tipicamente 5 a 10 produtos diários para testes contínuos. Novos projetos enfrentam um desafio diferente: necessitam de um catálogo inicial substancial de 1.000 a 5.000 produtos para conferir credibilidade à loja, criar múltiplos pontos de contacto com clientes e permitir que as próprias vendas indiquem quais produtos merecem investimento adicional em marketing.

Esta dinâmica cria um estrangulamento operacional: a importação manual de produtos na Shopify envolve múltiplas etapas demoradas, incluindo cópia de informações, organização de variantes (tamanhos, cores), download e upload de imagens, tradução de conteúdo e otimização de descrições. Este processo consome entre 20 a 45 minutos por produto. Projetando para os cenários operacionais típicos, um lojista que teste 10 produtos diários dedica 5 horas apenas a importação. Lançar uma loja com 1.000 produtos requer aproximadamente 500 horas de trabalho manual. Este paradoxo limita fundamentalmente a capacidade de escala: quanto mais produtos se pretende testar, mais recursos são consumidos em tarefas repetitivas.

A solução para este estrangulamento passa por automatizar o processo de importação através de tecnologia que simplifique radicalmente o fluxo de trabalho. O conceito fundamental consiste em permitir que um lojista importe produtos completos simplesmente inserindo um URL da página do produto de origem, seja de plataformas como AliExpress, Temu, ou mesmo de outras lojas Shopify. A tecnologia extrai automaticamente todas as informações relevantes: título, descrições, especificações, variantes, preços e imagens, organizando-as de forma estruturada e pronta para publicação. Este approach reduz drasticamente o tempo de importação, transformando um processo de 30 minutos numa operação de poucos segundos.

Contudo, a mera importação automática, embora valiosa, não resolve completamente o problema da qualidade. Os produtos importados de fornecedores frequentemente apresentam descrições de baixa qualidade, traduções deficientes, imagens com fundos inconsistentes ou má iluminação, e organização pouco clara de variantes. É aqui que a inteligência artificial acrescenta valor transformador. Modelos de linguagem de grande escala (Large Language Models) podem gerar automaticamente descrições profissionais e comercialmente persuasivas a partir das informações básicas importadas, traduzir contextualmente para qualquer idioma mantendo tom adequado, e reorganizar especificações técnicas de forma clara. Simultaneamente, tecnologias de processamento de imagens baseadas em redes adversariais generativas e modelos de difusão permitem melhorias visuais automáticas: remoção de fundos, correção de iluminação e contraste, redimensionamento inteligente, e criação de consistência visual entre produtos.

A integração destas capacidades no fluxo de importação pode reduzir o tempo necessário em 80% ou mais, não apenas mantendo mas frequentemente superando a qualidade dos processos manuais. O verdadeiro diferencial está no processamento em massa: um lojista pode selecionar centenas de produtos, definir parâmetros gerais como idioma de destino e estilo de descrição desejado, e obter todos os produtos processados, otimizados e prontos para publicação em minutos. Esta abordagem elimina o compromisso tradicional entre quantidade e qualidade, permitindo testar grandes volumes mantendo padrões elevados.

Esta solução tecnológica liberta os lojistas das tarefas operacionais repetitivas, permitindo-lhes dedicar mais tempo a áreas estratégicas do negócio como análise de dados, criação e otimização de campanhas publicitárias, atendimento ao cliente e desenvolvimento de estratégias de marketing que verdadeiramente diferenciam operações bem-sucedidas no competitivo mercado de e-commerce.

1.2 Motivação e Identificação do Problema

O interesse pelo presente tema surge da existência de uma brecha evidente no mercado das ferramentas de importação e otimização de produtos para Shopify. Apesar de existirem várias soluções, muitas delas falham em aspetos fundamentais: ou apresentam preços pouco acessíveis, ou não conseguem tirar partido da inteligência artificial de forma realmente útil neste contexto, ou acabam por ser demasiado complexas para o utilizador comum. Estas limitações tornam o processo de importação e gestão de produtos mais difícil e menos eficiente do que poderia ser.

A motivação pessoal para desenvolver esta solução resulta do meu contacto direto com pessoas que trabalham na área do e-commerce e que manifestam necessidade de uma ferramenta mais prática, acessível e eficaz. Além disso, conheço também pessoas que poderão utilizá-la e ajudar a promovê-la através de referral marketing, o que reforça o potencial real e imediato da aplicação.

1.3 Objetivos

O presente trabalho tem como objetivo geral desenvolver uma aplicação capaz de automatizar e otimizar o processo de importação de produtos para a plataforma Shopify, integrando funcionalidades de inteligência artificial e garantindo uma experiência simples, eficiente e acessível para o utilizador.

Para alcançar este objetivo geral, definem-se os seguintes objetivos específicos:

- **Desenvolver a aplicação** incorporando todos os requisitos identificados, incluindo extração automática de informação, otimização de descrições, tratamento de imagens e organização de variantes.
- **Integrar inteligência artificial** de forma adequada ao caso de uso, garantindo que a ferramenta melhora a qualidade do conteúdo gerado e reduz significativamente o trabalho manual.
- **Criar uma interface intuitiva** que simplifique o processo para utilizadores com diferentes níveis de experiência técnica.
- **Realizar testes com utilizadores reais** da área do e-commerce para validar funcionalidades e identificar oportunidades de melhoria.
- **Lançar a aplicação em ambiente real** e alcançar um determinado número de utilizadores pagantes (objetivo preliminar: entre **10 e 30**), permitindo avaliar a viabilidade e aceitação da solução no mercado.

1.4 Estrutura do Documento

O presente relatório está organizado em oito capítulos, complementados por anexos e outros elementos de apoio, de forma a apresentar de maneira clara e sistemática todo o trabalho desenvolvido. A estrutura é a seguinte:

- **Capítulo 1 – Introdução:** Apresenta o enquadramento do tema, a motivação e identificação do problema, os objetivos do trabalho e a organização geral do documento.
- **Capítulo 2 – Pertinência e Viabilidade:** Analisa a relevância e oportunidade do projeto, a sua viabilidade técnica e económica, a comparação com soluções existentes no mercado e a proposta de inovação associada.
- **Capítulo 3 – Especificação e Modelação:** Descreve os requisitos funcionais e não funcionais do sistema, os casos de uso, a modelação conceptual e os protótipos iniciais de interface.
- **Capítulo 4 – Solução Proposta:** Apresenta a solução desenvolvida, incluindo a arquitetura, as tecnologias utilizadas, os ambientes de teste e produção, a abrangência do projeto e a descrição dos seus componentes e interfaces.
- **Capítulo 5 – Testes e Validação:** Detalha os testes realizados, a metodologia aplicada e os resultados obtidos na validação da solução.
- **Capítulo 6 – Método e Planeamento:** Expõe o planeamento inicial do projeto, as fases de desenvolvimento e uma análise crítica ao cumprimento dos prazos e ajustamentos efetuados.
- **Capítulo 7 – Resultados:** Reúne os resultados dos testes, avalia o cumprimento dos requisitos definidos e analisa o desempenho da solução.
- **Capítulo 8 – Conclusão:** Resume as principais conclusões do trabalho, destaca os contributos alcançados e apresenta propostas de trabalhos futuros.

Para além dos capítulos principais, o relatório inclui ainda:

- **Bibliografia**, onde são listadas todas as fontes consultadas;
- **Anexo 1**, dedicado às recomendações de formatação do relatório;
- **Glossário**, contendo termos técnicos e relevantes utilizados ao longo do documento.

2 Pertinência e Viabilidade

2.1 Pertinência

A pertinência da Pullu foi validada através de um inquérito direcionado a utilizadores e potenciais utilizadores de plataformas de e-commerce, com especial foco em profissionais que trabalham com dropshipping e importação de produtos para lojas Shopify. O inquérito foi respondido por 10 participantes e permitiu identificar padrões de utilização, necessidades não satisfeitas e oportunidades claras de melhoria no processo atual de importação de produtos. Os resultados demonstram que existe uma necessidade real e mensurável no mercado para uma solução como a Pullu.

A primeira pergunta do inquérito procurou caracterizar a experiência dos respondentes no sector do e-commerce. Os resultados revelam que **80% dos inquiridos trabalham ativamente com e-commerce**, sendo que 40% está no sector há menos de 6 meses, 20% entre 6 a 12 meses, e 20% há mais de um ano. Os restantes 20% estão a planear começar a trabalhar na área.

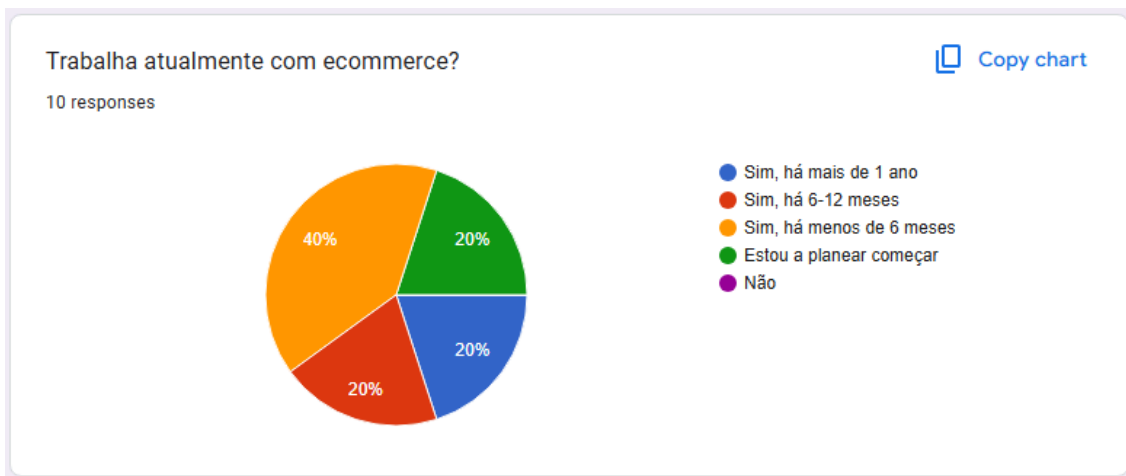


Figura 1

Quando questionados sobre o número de produtos que importam semanalmente, **60% dos respondentes indicaram importar entre 30 a 50 produtos por semana**, o que representa um volume operacional elevado. Os restantes 40% distribuem-se entre volumes menores (0-5, 6-15, 16-30 produtos) e volumes superiores (mais de 50 produtos). Este dado é fundamental para demonstrar a pertinência da Pullu: se considerarmos que cada produto demora entre 20 a 45 minutos a importar manualmente (conforme será demonstrado na próxima secção), um lojista que importe 40 produtos semanalmente dedica **entre 13 a 30 horas por semana apenas a tarefas de importação**. Esta carga operacional representa um estrangulamento crítico que a Pullu pretende resolver.



Figura 2

A análise do tempo médio gasto na importação de um único produto revela um problema de eficiência crítico: **60% dos inquiridos gastam mais de 30 minutos por produto** (30% entre 30-45 minutos e 30% mais de 45 minutos), 20% gastam entre 20-30 minutos, e apenas 20% conseguem importar produtos em menos de 20 minutos. Estes dados validam diretamente a motivação central do projeto apresentada no capítulo 1.2: o processo manual de importação é extremamente demorado e cria um compromisso forçado entre quantidade e qualidade. A Pullu, ao reduzir este tempo para menos de um minuto (conforme objetivo definido no resumo), representa uma **redução de 80-95% no tempo de importação**, libertando recursos humanos para tarefas estratégicas como análise de dados, marketing e atendimento ao cliente.

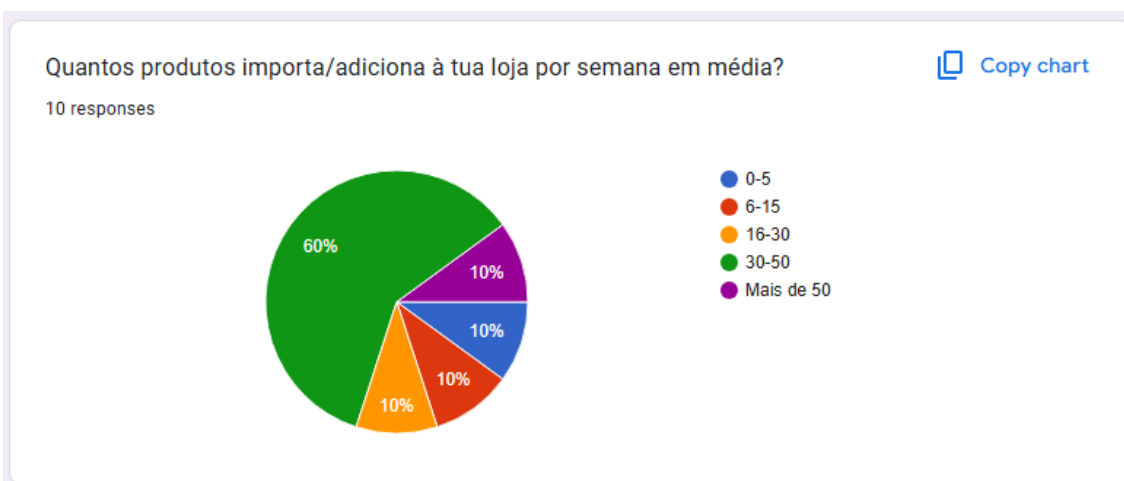


Figura 3

Figura 4 apresenta a distribuição das respostas à pergunta «Quanto tempo gasta em média a importar um produto?», com um total de 10 participantes. Os resultados mostram uma concentração no intervalo entre 10 e 30 minutos, que agrupa 60% das respostas (30% entre 10–20 minutos e 30% entre 20–30 minutos). 20% dos inquiridos indicam tempos entre 30 e 45 minutos, enquanto 10% referem menos de 10 minutos e 10% mais de 45 minutos. Em conjunto, 30% do painel reporta tempos superiores a 30 minutos por produto, o que sustenta a ideia de que a importação manual continua a consumir blocos relevantes de tempo, mesmo quando parte dos respondentes consegue concluir o processo em menos de meia hora. Estes

valores reforçam a pertinência de uma ferramenta que reduza o tempo por listagem e normalize a qualidade do conteúdo importado.

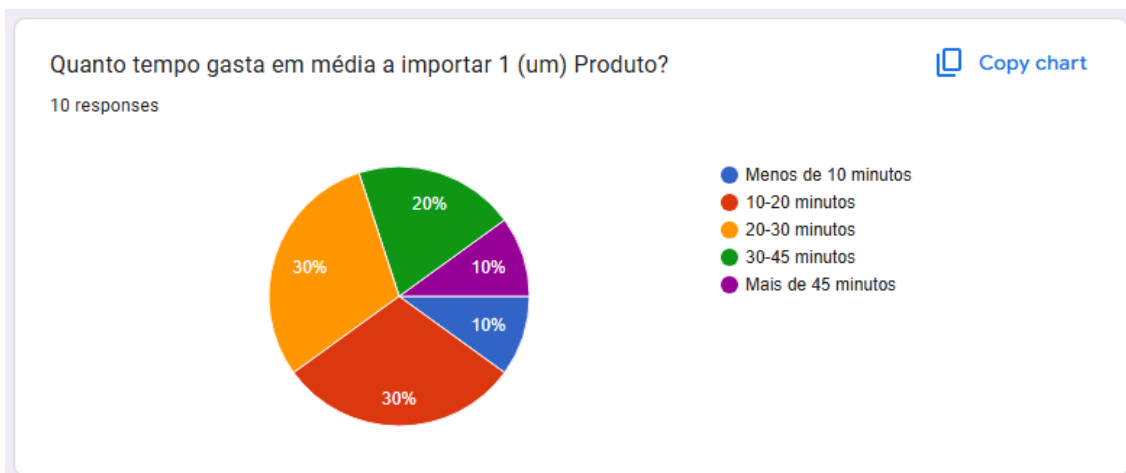


Figura 4

Este gráfico identifica os pontos de fricção específicos no fluxo de trabalho atual. **A edição e melhoramento de imagens foi identificada por 90% dos respondentes como a tarefa mais demorada**, seguida por organização de variantes (60%), ajuste de preços (60%), tradução de conteúdo (60%), download e upload de imagens (50%), e cópia de descrições (40%). Estes resultados validam diretamente os requisitos funcionais da Pullu, especialmente o RF-015 (edição de imagens com AI) e RF-014 (tradução e otimização de texto com AI). A concentração de respostas na edição de imagens justifica a priorização desta funcionalidade no desenvolvimento e reforça a proposta de inovação apresentada na secção 2.4: nenhuma solução concorrente oferece melhoramento automático de imagens com AI integrado no fluxo de importação.

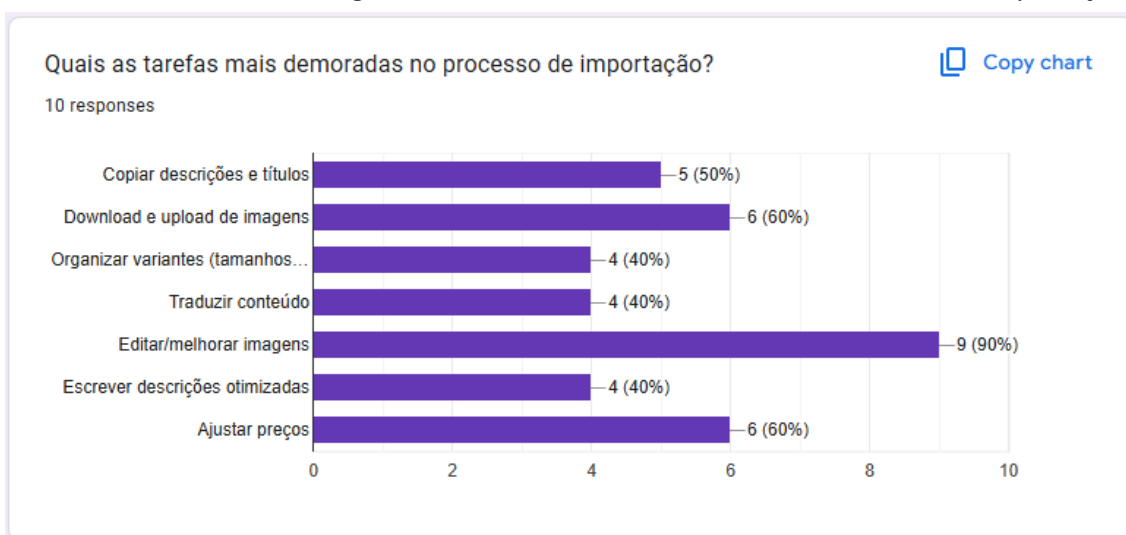


Figura 5

A análise das fontes de produtos revela que **88.9% dos inquiridos importam produtos de outras lojas Shopify**, confirmando que esta é a fonte primária e validando a decisão de priorizar a integração Shopify no MVP (RF-008, RF-009). Adicionalmente, 33.3% utilizam

AliExpress e 11.1% utilizam Temu, o que justifica a inclusão destas plataformas na roadmap de desenvolvimento futuro (RF-010), mas confirma que a decisão de focar primeiro em Shopify está alinhada com as necessidades reais do mercado.

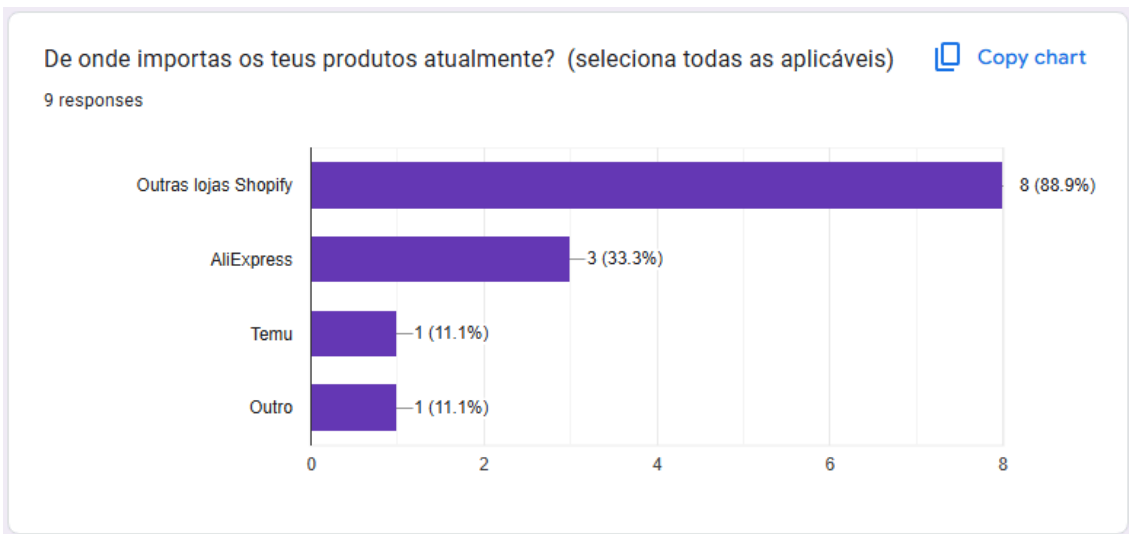


Figura 6

Quando questionados sobre o uso de ferramentas de importação, **50% dos respondentes realizam a importação manualmente**, sem recurso a qualquer aplicação especializada. Dos restantes 50%, 30% utilizam Kopy e 20% utilizam Poky. Este dado tem duas implicações fundamentais: (1) confirma que existe uma fatia significativa do mercado (50%) que ainda não adotou ferramentas de automação, representando uma oportunidade clara de aquisição de utilizadores; (2) entre os que utilizam ferramentas, a distribuição entre Kopy e Poky demonstra que nenhuma das soluções existentes domina completamente o mercado, o que sugere insatisfação com as opções atuais e abertura para alternativas melhores.

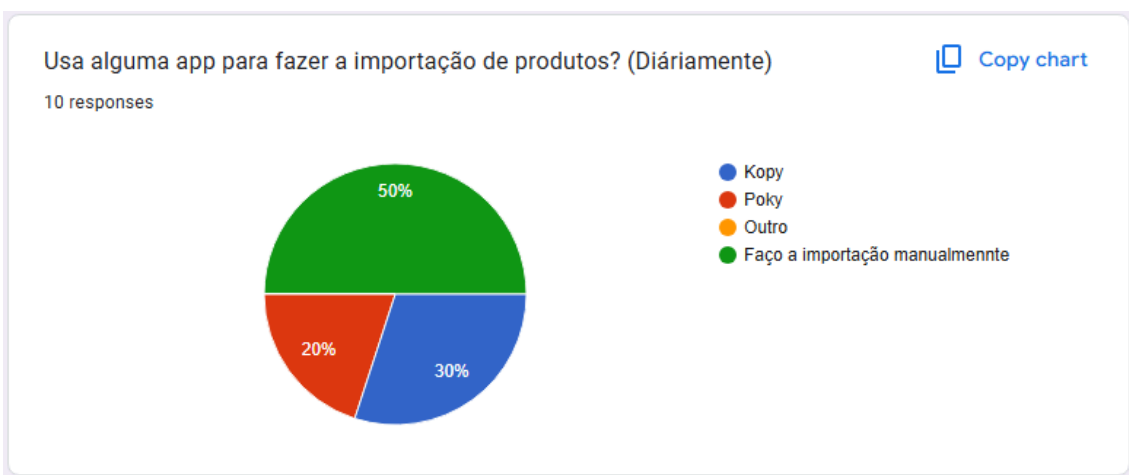


Figura 7

A pergunta final do inquérito avaliou o nível de satisfação com o processo atual de importação numa escala de 1 a 5. Os resultados revelam **uma insatisfação generalizada**: a maioria dos respondentes classificou a sua satisfação em níveis baixos (1-3), com poucos respondentes nos níveis superiores (4-5). Este dado é o mais crítico para demonstrar

pertinência: mesmo aqueles que já utilizam ferramentas de automação como Kopy ou Poky não estão satisfeitos com a solução atual, validando a necessidade de uma alternativa que combine simplicidade, funcionalidades avançadas de AI e modelo de preços acessível, conforme proposto pela Pullu.

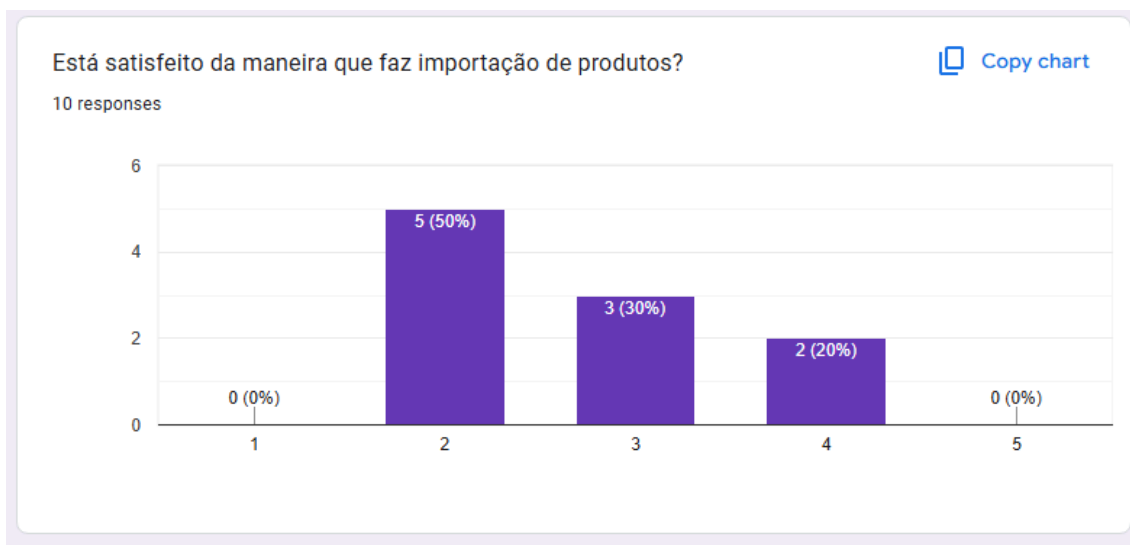


Figura 8

2.2 Viabilidade

A **Viabilidade** é a medida essencial que avalia a capacidade de uma solução ser implementada e, crucialmente, **sustentada com sucesso** a longo prazo. Esta avaliação deve ser realizada com base em **critérios objetivos e econométricos**, demonstrando que a solução proposta possui potencial para ser continuada após a conclusão do Trabalho Final de Curso (TFC), não se esgotando como um mero projeto acadêmico.

A viabilidade depende de uma reflexão e avaliação detalhada dos seguintes fatores: **técnicos, económicos, sociais e ambientais**.

- **Viabilidade Técnica:** Refere-se à disponibilidade e adequação de ferramentas, tecnologias e conhecimentos necessários para desenvolver e executar a solução. Pode ser avaliada através de testes com um **protótipo interativo** (para a primeira entrega) ou com um **protótipo funcional** e respetivos **testes de aceitação** (para o relatório final).
- **Viabilidade Económica:** Foca-se nos custos e benefícios da solução, determinando se é acessível, lucrativa e sustentável financeiramente. A sua avaliação deve ser feita através de uma **análise de custo-benefício** e pela projeção do **retorno do investimento** esperado.
- **Viabilidade Social:** Diz respeito à aceitação e ao apoio da solução por parte das partes interessadas (*stakeholders*), utilizadores e beneficiários envolvidos ou afetados pelo problema, avaliando se atende às suas necessidades, expectativas e preferências. Pode ser comprovada através de **inquéritos, entrevistas e inquéritos de satisfação**.

- **Alinhamento com os Objetivos de Desenvolvimento Sustentável (ODS):** Este aspeto integra a dimensão ambiental e social. Deve ser feita uma reflexão sobre o alinhamento da solução com os **Objetivos de Desenvolvimento Sustentável (ODS)**, identificando os ODS específicos que o projeto pretende abordar e como se alinha com as metas de cada um. É obrigatório avaliar o impacto esperado do projeto nos ODS escolhidos.

2.3 Análise Comparativa com Soluções Existentes

2.3.1 Soluções existentes

No mercado de ferramentas de importação de produtos para Shopify, existem várias soluções que procuram automatizar parte do processo de criação de listagens. As principais alternativas identificadas são:

Kopy – Aplicação desenvolvida especificamente para importação de produtos de lojas Shopify concorrentes, permitindo aos utilizadores copiar listagens completas com apenas um clique. A ferramenta extrai automaticamente todas as informações do produto, incluindo título, descrição, imagens, variantes e preços. Oferece funcionalidades de edição básica de texto e imagens, mas sem recurso a inteligência artificial avançada. A Kopy permite importação em massa e sincronização de produtos, sendo particularmente popular entre lojistas que procuram replicar produtos bem-sucedidos de outras lojas. O modelo de preços baseia-se em subscrição mensal, com planos que variam consoante o número de produtos importados [Kopy, 2024].

Poky – Ferramenta de importação que suporta múltiplas plataformas, incluindo AliExpress, Temu e outras lojas Shopify. A Poky destaca-se pela sua interface simples e processo de importação rápido, permitindo aos utilizadores inserir um URL e obter automaticamente todos os dados do produto. Oferece funcionalidades básicas de tradução automática e edição de descrições, mas sem capacidades avançadas de inteligência artificial para otimização de conteúdo ou melhoramento de imagens. A plataforma inclui importação em massa e permite personalização de variantes e preços antes da publicação. O modelo de preços é baseado em créditos, onde cada importação consome um crédito, com pacotes mensais disponíveis [Poky, 2024].

Estas soluções, embora funcionais para importação básica de produtos, apresentam limitações significativas quando comparadas com a proposta da Pullu. Nenhuma delas integra capacidades avançadas de inteligência artificial para melhoramento de imagens, geração de vídeos promocionais ou otimização completa de descrições de produtos. Adicionalmente, a experiência de utilizador tende a ser menos intuitiva, exigindo múltiplos passos para alcançar resultados que poderiam ser automatizados. Os modelos de preços, embora variados, nem sempre são transparentes ou acessíveis para lojistas em fase inicial.

2.3.2 Análise de benchmarking

	Pullu	Kopy	Poky
Importação instantânea de produtos	X	X	X
Importação em bulk	X	X	X
Tradução e melhoramento de texto com AI	X	X	X
Customizar todas as opções do produto	X		
Preço acessível	X		X
Melhoramento de Imagens	X		
Geração de imagens com AI	X		
Geração de vídeos com AI	X		
Fácil de se usar	X		

Tabela 1

2.4 Proposta de inovação e mais-valias

A Pullu distingue-se das soluções existentes no mercado por integrar numa única plataforma funcionalidades que atualmente não são oferecidas por nenhuma das plataformas de ecommerce tradicionais. Em particular, não existe uma ferramenta que combine a importação automática de produtos, edição de textos e imagens com inteligência artificial, e publicação direta na Shopify de forma simples, acessível e rápida.

A solução foca-se na facilidade de utilização, proporcionando uma interface intuitiva que permite a qualquer utilizador, mesmo sem experiência técnica, criar listagens profissionais em minutos. Além disso, o modelo baseado em subscrições e créditos torna a ferramenta financeiramente acessível, permitindo que lojistas de diferentes dimensões testem e escalem os seus produtos sem custos elevados iniciais.

Para os clientes finais, estas melhorias traduzem-se numa experiência mais eficiente e consistente, com produtos apresentados de forma clara, imagens de qualidade

superior e descrições otimizadas. Por fim, a automatização de tarefas repetitivas aumenta a eficiência operacional dos lojistas, reduz o tempo gasto na criação de listagens em até 80% e libera recursos para atividades estratégicas. A combinação de inovação tecnológica, facilidade de utilização e impacto direto nos resultados dos clientes representa a principal mais-valia da Pullu em comparação com soluções concorrentes.

2.5 Identificação de oportunidade de negócio

Pretende-se explorar a oportunidade de negócio associada à plataforma Pullu através de uma abordagem faseada. No período compreendido entre o início do desenvolvimento e a entrega final, será implementado e testado um modelo "free" para *user testing*. Nesta fase, os utilizadores poderão experimentar as funcionalidades da plataforma de forma gratuita, utilizando as suas próprias API keys (como as da OpenAI ou similares) para o processamento de inteligência artificial. Esta estratégia permite validar a ferramenta com utilizadores reais e recolher feedback direto sobre a usabilidade, sem custos de infraestrutura de IA para o projeto nesta fase inicial.

Após esta fase de experimentação, será avaliada a pertinência de evoluir para um modelo de monetização baseado em subscrições e créditos. Caso se confirme a viabilidade e aceitação por parte dos utilizadores, a implementação do sistema de pagamentos será realizada através da Stripe, garantindo segurança e escalabilidade na gestão das transações. Os preços e níveis de funcionalidades serão definidos de acordo com o valor percebido durante os testes e a aceitação demonstrada pelo mercado.

Este modelo pretende fomentar a potencial monetização da solução e a retenção de clientes, permitindo que o projeto evolua de uma prova de conceito para uma solução comercial sustentável no mercado de ecommerce, dependendo dos resultados obtidos na fase de testes.

3 Especificação e Modelação

3.1 Análise de Requisitos

3.1.1 Enumeração de Requisitos

O levantamento e a análise de requisitos são uma etapa fundamental do planeamento de um projeto, pois é nesta fase que se identificam e documentam todas as necessidades do sistema. Os requisitos encontram-se organizados segundo categorias distintas: **RF (Requisitos Funcionais)**, que descrevem o que o sistema deve fazer; **RNF (Requisitos Não Funcionais)**, que definem características de qualidade como desempenho, segurança ou usabilidade; **RS (Requisitos de Sistema)**, que representam dependências técnicas e condições necessárias para o funcionamento da solução; e **BN (Requisitos de Negócio)**, que estabelecem regras e objetivos associados ao modelo de negócio. Na tabela seguinte apresenta-se a lista completa dos requisitos identificados.

ID	Requisito	Prioridade	Estado
RF-001	Disponibilizar landing page da SaaS com toda a informação pertinente	Must	Implementado
RF-002	Autenticação por email com magic link e Google OAuth	Must	Implementado
RF-003	Sistema de subscrições via Stripe	Must	Adiado/Substituído
RF-004	Dashboard de utilizador para utilizadores autenticados	Must	Implementado
RF-005	Gestão de perfil e subscrição do utilizador	Must	Implementado
RF-006	Conexão de lojas Shopify via OAuth	Must	Implementado
RF-007	Gestão e alternância entre múltiplas lojas	Must	Implementado
RF-008	Scraping de produtos de lojas Shopify	Must	Implementado
RF-009	Scraping coleções de lojas Shopify	Must	Implementado
RF-010	Scraping de produtos do AliExpress e Temu	Should	Planeado

RF-011	Validação e detecção automática de plataforma	Could	Planeado
RF-012	Visualização de informação do produto importado	Must	Implementado
RF-013	Edição manual de todas as informações do produto	Must	Implementado
RF-014	Edição e tradução de texto com AI	Must	Implementado
RF-015	Edição e processamento de imagens com AI	Must	Parcialmente Implementado
RF-016	Publicação de produtos na loja Shopify selecionada	Must	Implementado
RF-017	Sistema de gestão de créditos/balance	Must	Adiado/Substituído
RNF-001	O sistema deve importar dados de um produto em menos de 2,5 segundos	Must	Implementado
RNF-002	Tokens sensíveis (Shopify, Stripe, Supabase) devem ser armazenados de forma segura e encriptada.	Must	Planeado
RNF-003	A interface deve ser intuitiva e permitir que um utilizador sem experiência técnica complete uma importação	Must	Implementado
RS-001	O sistema depende das APIs Shopify, Supabase, Stripe e serviços de AI para funcionar corretamente.	Must	Implementado
RS-002	O sistema requer ligação à internet para todas as funcionalidades principais.	Must	Implementado
RS-003	A aplicação deve ser publicada em ambiente de produção na Vercel.	Must	Implementado
BN-001	O acesso às funcionalidades premium deve ser regulado por um modelo de subscrição mensal.	Must	Adiado/Substituído

Tabela 2

3.1.2 Descrição detalhada dos requisitos principais

Requisitos de Interface e Acesso - O requisito RF-001 exige a disponibilização de uma landing page completa, que deve atuar como ponto de contacto inicial para o utilizador, apresentando a proposta de valor, funcionalidades e documentação do produto. Em termos de acesso, o RF-002 define o método de autenticação, que será por email com magic link ou via conta Google, garantindo um processo de login seguro e simplificado. Para sustentar a viabilidade económica futura, previu-se a integração com o Stripe (RF-003) para gestão de subscrições; contudo, nesta fase de projeto, este requisito encontra-se adiado, priorizando-se a validação técnica da plataforma através de um modelo de utilização gratuita.

Requisitos de Importação e Scraping - A funcionalidade central é a Extração (Scraping) de informação de produto (RF-004). O objetivo é que, ao ser introduzido um URL de fontes externas como AliExpress, Temu ou outras lojas Shopify, a Pullu recolha automaticamente todos os dados relevantes do produto: título, descrições, preços, variantes e imagens. Esta operação tem como meta de performance o cumprimento do requisito não funcional de importar os dados em menos de 2,5 segundos (RNF-001), valor que é validado em condições normais de rede. Antes da publicação, o RF-005 trata da Conexão de lojas Shopify, um passo que permite estabelecer uma ligação segura e autorizada entre a Pullu e a loja do utilizador. Por fim, o RF-012, Visualização de informação do produto importado, assegura que todos os dados extraídos sejam apresentados de forma clara e estruturada num dashboard intuitivo para a etapa de revisão e edição.

Requisitos de Otimização e Publicação com AI - O processo de otimização começa com o RF-013, que permite a Edição manual de todas as informações do produto no dashboard, garantindo que o utilizador tem controlo total sobre a listagem. A grande inovação surge com o RF-014, a Edição e tradução de texto com AI, que utiliza modelos de linguagem para otimizar, reescrever e traduzir descrições e títulos. Nesta versão, a utilização destas funcionalidades depende da configuração de uma chave de API própria pelo utilizador. Paralelamente, o RF-015 aplica a AI na Edição e processamento de imagens, utilizando serviços como Fal.ai para corrigir falhas visuais e melhorar as imagens de produto. Uma vez otimizado, o processo é concluído pelo RF-016, a Publicação de produtos na loja Shopify selecionada, onde a listagem final é carregada diretamente para a plataforma do utilizador.

Requisitos de Negócio e Desempenho - O RNF-001 estabelece um critério de desempenho fundamental: a importação de dados deve ocorrer em menos de 2,5 segundos. Este critério é vital para que a Pullu cumpra a sua promessa de transformar um processo longo e manual numa operação rápida. Para gerir os custos operacionais e o valor entregue, o requisito original de gestão de créditos (RF-017) foi substituído pelo modelo Bring Your Own Key (BYOK). Desta forma, o utilizador gere o seu próprio consumo diretamente com os provedores de IA, eliminando a necessidade de um sistema de balance gerido pela Pullu

nesta fase, o que assegura a viabilidade económica e o controlo de custos nesta etapa de testes.

3.1.3 Casos de Uso/*User Stories*

As User Stories são apresentadas no formato padrão: **Como um [Tipo de Utilizador], eu quero [Funcionalidade], para que [Benefício/Valor].**

1. Conexão e Sustentabilidade

- **Conexão à Loja:** Como um Lojista, eu quero ligar a minha loja Shopify à Pullu de forma segura (RF-005), **para que a Pullu possa publicar os meus produtos otimizados diretamente**, sem passos intermédios.
- **Configuração de Chave de API:** Como um Lojista, eu quero configurar a minha própria chave de API da OpenAI nas definições (**RF-017 - Substituído**), para que possa utilizar as funcionalidades de inteligência artificial de forma direta e gerir o meu próprio consumo sem depender de um sistema de créditos interno.

2. Extração e Otimização da Listagem (Otimização de Tempo)

- **Extração Instantânea:** Como um Lojista, eu insiro um URL de um fornecedor (AliExpress, Temu) e quero que o sistema extraia todos os dados (RF-004), **para que o processo de importação de um produto demore segundos, não minutos**, libertando o meu tempo para tarefas de marketing (RNF-001).
- **Qualidade da Descrição:** Como um Lojista, eu quero usar a AI para reescrever as descrições (RF-014), **para que o texto final seja profissional e persuasivo** e as minhas listagens não pareçam cópias mal traduzidas do fornecedor.
- **Melhoria Visual:** Como um Lojista, eu quero remover fundos, logotipos ou marcas de água de todas as imagens do produto (RF-015), **para que as minhas fotografias sejam limpas, apelativas e reflitam a marca da minha loja.**
- **Ajuste de Preço Rápido:** Como um Lojista, eu quero poder editar rapidamente os preços de custo e de venda (RF-013), **para garantir que a minha margem de lucro está correta** antes de publicar.

3. Publicação e Fluxo de Trabalho

- **Revisão Final:** Como um Lojista, eu quero que todos os campos (título, descrição, variantes e imagens) estejam visíveis e editáveis após a extração (RF-012), **para que eu possa fazer uma revisão final rápida** antes de enviar para a loja.
- **Finalização do Processo:** Como um Lojista, eu quero clicar num botão para enviar o produto totalmente otimizado e *clean* (RF-016), **para que a listagem apareça imediatamente na minha loja Shopify pronta para vender.**
- **Autonomia:** Como um Lojista, eu quero ter controlo total sobre o produto final (RF-013), **para garantir que a minha listagem cumpre todos os critérios de SEO e branding da minha loja.**

3.2 Modelação

O modelo de Entidade-Relação apresentado foca-se na normalização das entidades chave para a operação da Pullu: a configuração de credenciais de IA (RF-017) e a integração de lojas Shopify (RF-005). As entidades principais – **user_openai_credentials** e **shopify_stores** – estão em conformidade com a 3ª Forma Normal e são ligadas diretamente ao utilizador através da Chave Estrangeira (FK) `user_id`. Esta **FK** referencia a tabela **auth.users.id**, que pertence ao schema de autenticação do **Supabase** e não é detalhada no diagrama principal por uma questão de escopo. Este modelo garante que as chaves de API personalizadas e os identificadores das lojas Shopify estão corretamente mapeados a cada utilizador, permitindo que a plataforma execute operações de IA e publicação de forma isolada e segura, sem a necessidade de um sistema de créditos interno nesta fase.

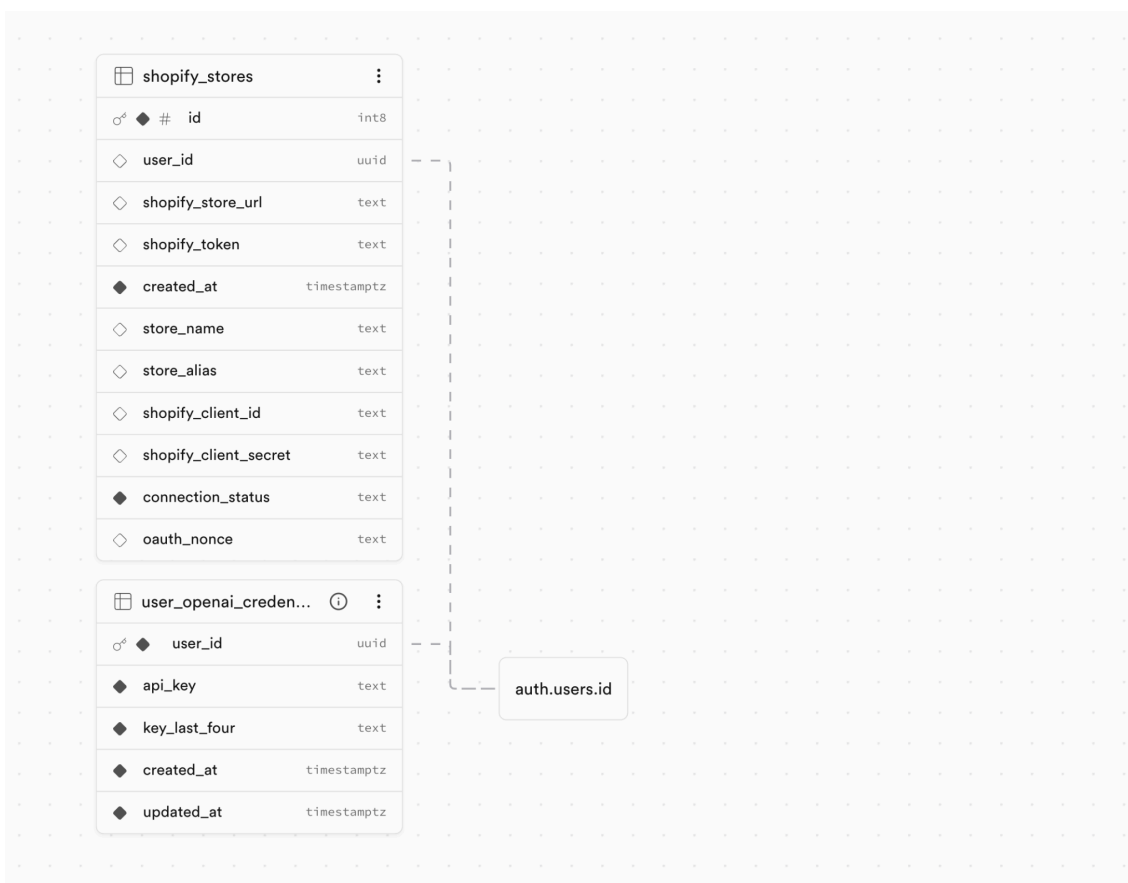


Figura 9

3.3 Protótipos de Interface

As interfaces apresentadas de seguida correspondem a mockups conceptuais que ilustram a proposta de design e fluxo de navegação da Pullu. Estes protótipos visuais não representam o produto final, mas servem para validar a arquitetura de informação e

demonstrar a abordagem simplificada da experiência de utilizador. O design foca-se na clareza e eficiência, permitindo que utilizadores completem o processo de importação e edição de produtos com o mínimo de fricção possível.

Landing Page da Pullu: Landing page que apresenta a proposta de valor da plataforma, destacando as três funcionalidades principais (importação, tradução com AI e edição de imagens). Inclui secção interativa "Try it now" para experimentação sem registo e botões de autenticação (RF-001, RF-002).

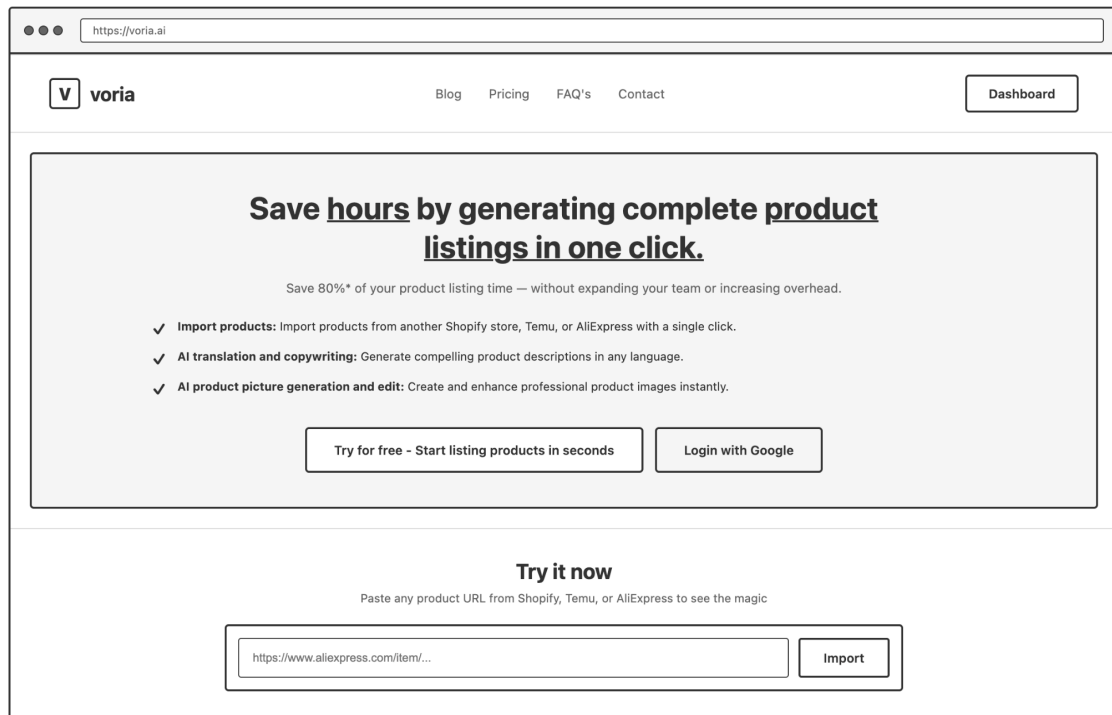


Figura 10

Dashboard de Importação de Produtos: Interface principal após autenticação, com sidebar simplificada (Home e Product Listing). Apresenta duas funcionalidades: importação individual de produtos via URL e importação em massa de coleções (RF-004, RF-008, RF-009, RF-010).

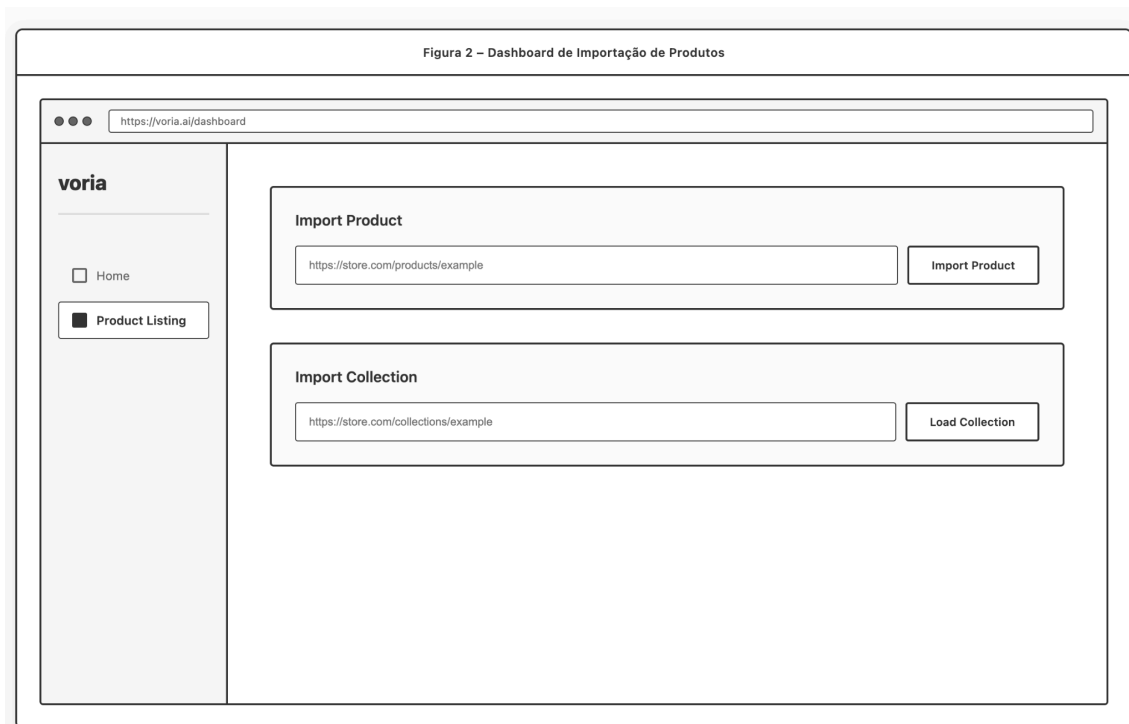


Figura 11

Interface de Edição de Produto com AI: Ecrã de edição organizado em duas colunas: à esquerda, informações básicas e descrição editável; à direita, gestão de imagens e configurações. Inclui botão "Translate All" para otimização com AI e botão final "Add to Shopify Store" para publicação (RF-012, RF-013, RF-014, RF-015, RF-016).

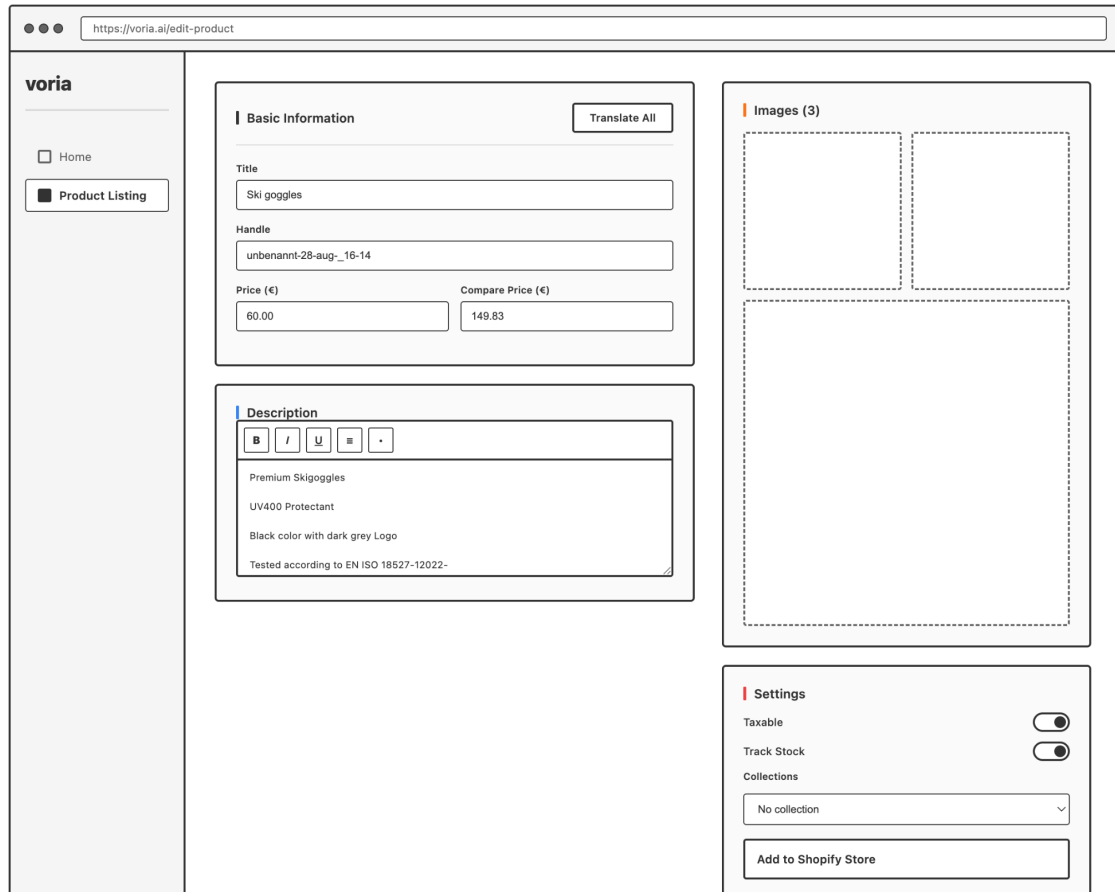


Figura 12

4 Solução Proposta

4.1 Apresentação

A Pullu é uma aplicação web SaaS desenvolvida para automatizar e otimizar o processo de importação de produtos para lojas Shopify. A solução permite que utilizadores de e-commerce, especialmente no modelo de dropshipping, importem produtos de fontes externas (lojas Shopify, AliExpress, Temu) através da simples inserção de um URL. O sistema extrai automaticamente todas as informações do produto – incluindo título, descrições, imagens, variantes e preços – e disponibiliza ferramentas de inteligência artificial para otimização de texto e processamento de imagens. Após edição, o produto pode ser publicado diretamente na loja Shopify do utilizador com um único clique, reduzindo um processo que normalmente demora 20 a 45 minutos para menos de um minuto. Comparativamente às soluções apresentadas no benchmarking (Kopy e Poky), a Pullu distingue-se pela integração completa de funcionalidades de AI para melhoramento de imagens e geração de descrições profissionais, pela interface intuitiva e simplificada, e por um modelo de experimentação flexível onde o utilizador utiliza as suas próprias chaves de API.

O estado atual do desenvolvimento corresponde a um **MVP funcional** que valida o conceito principal da solução. Nesta fase, a Landing Page encontra-se totalmente desenvolvida e publicada, servindo como ponto de contacto principal e apresentação do produto. O Dashboard está também concluído e totalmente funcional, permitindo a gestão completa do fluxo de trabalho. Encontram-se implementados os seguintes componentes: sistema de autenticação (email e Google OAuth), importação completa de produtos e coleções de lojas Shopify via scraping, interface de edição de produto com capacidade de alterar todos os campos (textos, imagens, variantes, preços), integração com API da Shopify para publicação de produtos, e otimização de texto através de modelos de linguagem da OpenAI. Componentes planeados para evoluções futuras ou em fase de avaliação incluem: sistema de pagamentos via Stripe, processamento avançado de imagens com Photoroom e Fal.ai, importação direta do AliExpress e Temu, e geração de vídeos promocionais com AI.

Recursos disponíveis:

- **Repositório Git:** <https://github.com/DEISI-ULHT-TFC-2025-26/TFC-ALUNO2214-PULLU>
- **Aplicação funcional:** <https://pullu.app/>
- **Vídeo demonstrativo:** <https://youtu.be/DwtoYLIMrko>

Este capítulo está organizado nas seguintes secções: **4.2 Arquitetura**, onde é apresentada a estrutura técnica da aplicação e a justificação das escolhas arquiteturais; **4.3 Tecnologias e Ferramentas Utilizadas**, que detalha o stack tecnológico e as integrações com serviços externos; **4.4 Ambientes de Teste e de Produção**, descrevendo a configuração dos

ambientes de desenvolvimento e deployment; **4.5 Abrangência**, que relaciona o projeto com os conhecimentos adquiridos ao longo do curso; **4.6 Componentes**, onde cada módulo funcional da aplicação é descrito em detalhe; e **4.7 Interfaces**, que referencia os protótipos visuais apresentados anteriormente.

4.2 Arquitetura

A Pullu adota uma **arquitetura serverless monolítica** baseada em Next.js, que unifica frontend e backend numa única aplicação. Esta escolha justifica-se pela simplicidade de desenvolvimento, rapidez de iteração e redução de custos operacionais, características essenciais para um MVP.

Modelo Arquitetural

A aplicação segue o padrão **Full-Stack Next.js**, onde:

- O **frontend** é renderizado com React e Server-Side Rendering (SSR)
- O **backend** opera através de API Routes serverless
- Não existe separação física entre camadas, simplificando deployment e manutenção

Esta arquitetura elimina a complexidade de gerir múltiplos servidores e permite auto-scaling automático através da Vercel.

Componentes Principais

Frontend (Camada de Apresentação)

- React com componentes funcionais
- Tailwind CSS para estilização
- SSR para performance e SEO
- State management local com React Hooks

Backend (Camada de Lógica)

- API Routes como endpoints serverless
- Processamento de scraping de produtos
- Orquestração de chamadas a APIs externas
- Validação de dados e aplicação de regras de negócio

Base de Dados

- PostgreSQL gerido pelo Supabase

- Row Level Security (RLS) para isolamento de dados
- Tabelas normalizadas (3FN): users, user_balances, subscriptions, shopify_stores

Integrações Externas

1. **Shopify API:** Publicação de produtos
2. **OpenAI + DeepL:** Otimização de texto
3. **Photoroom + Fal.ai:** Processamento de imagens
4. **Stripe:** Pagamentos e subscrições

Infraestrutura e Deployment

- **Hosting:** Vercel (auto-scaling, CDN global, CI/CD)
- **Base de Dados:** Supabase Cloud (backups automáticos)
- **Storage:** Supabase Storage (imagens temporárias)
- **Segurança:** HTTPS, JWT, variáveis de ambiente encriptadas

Fluxo de Dados Típico

O fluxo ilustra o percurso típico de uma operação na Pullu: o utilizador interage com o frontend, que envia requests para as API Routes onde ocorre validação e autenticação. Dependendo da operação, são acionados serviços de scraping, AI ou Shopify API, sendo todas as transações registradas no Supabase para débito de créditos e logging. A resposta é então devolvida ao frontend e apresentada ao utilizador, completando o ciclo.

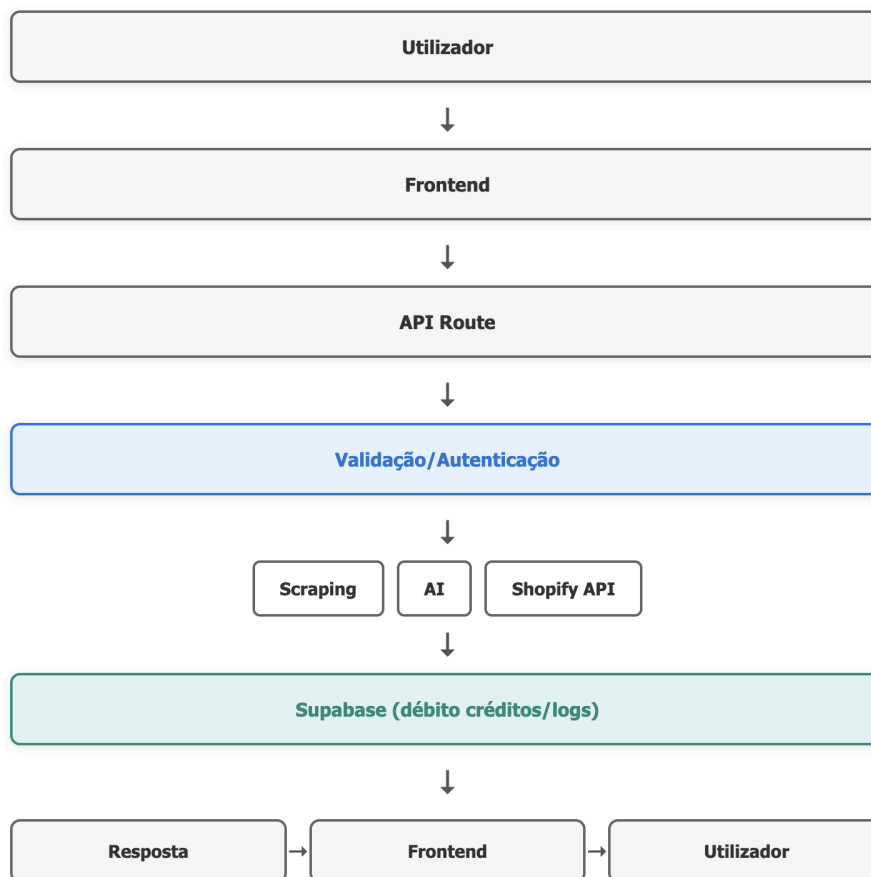


Figura 13

Diagrama de Arquitetura

O diagrama apresenta a arquitetura serverless da Pullu, onde o utilizador interage com uma aplicação Next.js hospedada na Vercel. Esta aplicação integra o frontend React e as API Routes backend, que comunicam com três camadas de serviços externos: Supabase para gestão de dados e autenticação, APIs externas para funcionalidades de scraping e AI, e Stripe para processamento de pagamentos.

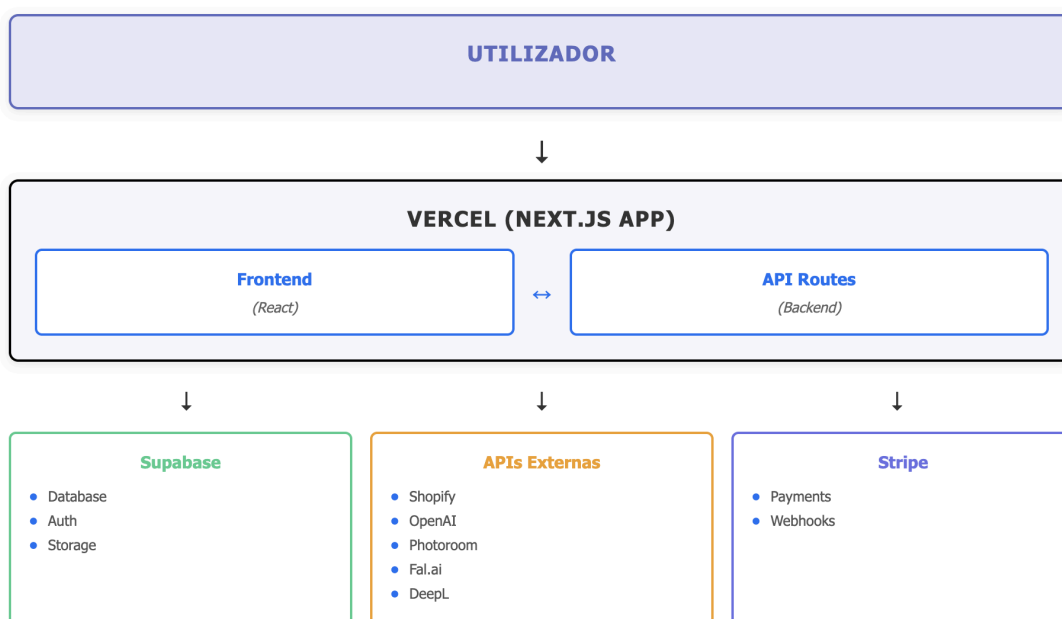


Figura 14

4.3 Tecnologias e Ferramentas Utilizadas

O desenvolvimento da Pullu utiliza um conjunto de ferramentas robustas e modernas, justificadas pela necessidade de criar um Produto Mínimo Viável (MVP) que seja performático, fácil de manter e preparado para o crescimento.

Next.js - Framework full-stack escolhido para unificar o desenvolvimento frontend e backend através das API Routes. Esta decisão permite a reutilização de código, reduz a complexidade da arquitetura e acelera o desenvolvimento, sendo essencial para cumprir os requisitos de performance (RNF-001) e garantir um time-to-market reduzido.

TypeScript - Linguagem de programação adotada para garantir robustez do código através de tipagem estática. Permite detetar erros em tempo de desenvolvimento, facilita a manutenção e refatoração do código, e melhora a experiência de desenvolvimento com

autocompletar inteligente. É fundamental para garantir a qualidade e escalabilidade da aplicação à medida que novos requisitos são implementados.

Tailwind CSS - Framework de estilização utility-first escolhido para criar interfaces responsivas e profissionais de forma eficiente. Permite desenvolvimento rápido de componentes visuais, mantém consistência visual em toda a aplicação e reduz significativamente o código CSS customizado. Essencial para cumprir o requisito de interface intuitiva (RNF-003) e garantir uma experiência visual moderna.

Supabase - Plataforma backend-as-a-service adotada como solução integrada para base de dados PostgreSQL e sistema de autenticação. Fornece uma API REST automática, autenticação segura por email e OAuth (RF-002), e gestão de utilizadores pronta a usar. Esta escolha elimina a necessidade de configurar infraestrutura backend complexa, permitindo focar no desenvolvimento de funcionalidades core e reduzindo drasticamente o tempo de implementação dos requisitos de dados e autenticação.

Stripe - Plataforma de pagamentos escolhida como padrão da indústria para SaaS. Essencial para processar pagamentos de forma segura (RF-003), gerir o ciclo de vida das subscrições e integrar o sistema de créditos (RF-017). Oferece APIs robustas, webhooks para sincronização automática de estados de pagamento e conformidade com regulamentações internacionais (PCI-DSS). A sua adoção garante confiança dos utilizadores e simplifica a implementação do modelo de negócio.

Shopify API - Interface de programação necessária para estabelecer a ligação segura com a loja do utilizador através de OAuth (RF-006). Permite a leitura de dados da loja, publicação direta de produtos (RF-016) e gestão de variantes e imagens. A integração com a Shopify API é o núcleo tecnológico da proposta de valor da Pullu, sendo indispensável para automatizar o fluxo de importação e publicação de produtos.

OpenAI API (ChatGPT) - Modelo de linguagem de grande escala utilizado para otimização e reescrita de texto de marketing (RF-014). Permite gerar descrições persuasivas, melhorar a estrutura de conteúdo e adaptar o tom ao público-alvo. A integração desta API é fundamental para transformar descrições genéricas de fornecedores em conteúdo profissional que aumenta as taxas de conversão.

PhotoRoom API - Serviço especializado em remoção automática de fundos de imagens de produtos. Utilizado para processar imagens em lote (RF-015), garantindo consistência visual e qualidade profissional das listagens. Esta ferramenta elimina a necessidade de edição manual de imagens, um dos passos mais demorados no processo tradicional de importação de produtos.

Fal.ai - Plataforma de APIs de inteligência artificial para processamento avançado de imagens, incluindo correção de iluminação, melhoria de qualidade e outras otimizações visuais (RF-015). Complementa o Photoroom ao oferecer capacidades adicionais de edição que elevam a qualidade visual das listagens de produtos.

Vercel - Plataforma de hosting otimizada para aplicações Next.js. Permite deployment contínuo integrado com Git, gestão automática de infraestrutura escalável, CDN global para performance otimizada e previews automáticas de cada commit. Esta escolha elimina a necessidade de configuração manual de servidores (RS-003), garante alta disponibilidade e permite escalar automaticamente conforme a procura aumenta.

4.4 Ambientes de Teste e de Produção

O ambiente de desenvolvimento executa localmente na máquina do programador através do comando `npm run dev`, que inicia o Next.js Development Server no endereço `localhost:3000`.

Requisitos computacionais mínimos:

- **RAM:** 8GB (funcional com 4GB, mas 8GB recomendado para operação confortável)
- **Processador:** Dual-core 2.0GHz ou superior
- **Armazenamento:** 10GB de espaço livre em disco
- **Ligação à Internet:** Necessária para acesso às APIs externas

Serviços utilizados:

- **Supabase:** Ligação direta ao projeto de desenvolvimento hospedado na cloud, sem necessidade de emulação local
- **Stripe Test Mode:** Utilização do ambiente de testes do Stripe para validar fluxos de pagamento e subscrições sem processar transações reais
- **APIs de AI:** Acesso direto aos serviços (OpenAI, DeepL, Photoroom, Fal.ai) através de chaves de desenvolvimento com rate limits reduzidos

Este ambiente permite desenvolvimento rápido com hot-reload automático, facilitando a iteração sobre funcionalidades sem necessidade de deployment.

O ambiente de produção será hospedado na **Vercel** e utiliza os seguintes serviços cloud em modo produtivo:

Infraestrutura:

- **Vercel:** Hosting serverless com auto-scaling e CDN global
- **Supabase:** Base de dados PostgreSQL e serviço de autenticação em produção
- **Stripe:** Processamento real de pagamentos e gestão de subscrições
- **APIs de AI:** OpenAI, DeepL, Photoroom e Fal.ai em modo produtivo

Requisitos de rede:

- **Bandwidth estimado:** 50-100GB/mês para operação com 100 utilizadores ativos
- **Latência crítica:** < 2.5 segundos para importação completa de produto (RNF-001)

- **Disponibilidade:** 99.9% uptime garantido pelos SLAs da Vercel e Supabase

O deployment em produção é automatizado através de integração contínua com Git, onde cada push para a branch principal desencadeia um build e deployment automático na Vercel.

4.5 Abrangência

O desenvolvimento da Pullu aplica conhecimentos adquiridos em diversas unidades curriculares do curso de Informática de Gestão:

Fundamentos de Programação e Algoritmia e Estruturas de Dados - Forneceram as bases de lógica de programação, manipulação de dados estruturados e algoritmos essenciais, que permitiram a aprendizagem autónoma de JavaScript/TypeScript para o desenvolvimento da aplicação.

Programação Web - Estabeleceu os fundamentos de desenvolvimento web e conceitos de programação que facilitaram a transição para Next.js, implementação de interfaces responsivas e integração de APIs RESTful.

Bases de Dados - Modelação e normalização da base de dados PostgreSQL, gestão de relacionamentos entre entidades (utilizadores, subscrições, lojas) e otimização de queries para performance.

Engenharia de Requisitos e Testes - Levantamento e especificação de requisitos funcionais e não funcionais, definição de casos de uso, desenvolvimento de planos de teste e validação de critérios de aceitação.

Teoria e Prática de Marketing - Compreensão do contexto de e-commerce e dropshipping, análise de necessidades do mercado-alvo e desenvolvimento de proposta de valor diferenciadora.

Cálculo Financeiro e Gestão Financeira - Definição de modelo de preços, análise de custos operacionais variáveis (APIs de AI) e fixos (hosting), e avaliação de viabilidade económica do negócio.

Direito Informático - Compreensão de regulamentações de proteção de dados (RGPD), gestão segura de informações de utilizadores e lojas, e conhecimento de aspetos legais relacionados com processamento de pagamentos e armazenamento de dados sensíveis.

Engenharia de Software - Aplicação de boas práticas de desenvolvimento, arquitetura de software escalável, separação de responsabilidades e gestão de código através de controlo de versões (Git).

Interação Humano-Máquina - Design de interfaces intuitivas (RNF-003), prototipagem de ecrãs, e otimização da experiência de utilizador para reduzir fricção no fluxo de importação de produtos.

Inteligência Artificial - Integração de modelos de linguagem (ChatGPT) para otimização de texto e APIs de processamento de imagem (Photoroom, Fal.ai) para melhoramento visual automático.

Sistemas de Informação na Nuvem - Deployment em ambiente serverless (Vercel), utilização de serviços cloud (Supabase) e gestão de infraestrutura escalável sem configuração manual de servidores.

Gestão de Projetos - Planeamento de sprints, gestão de backlog de requisitos, monitorização de progresso e ajuste de planeamento conforme dificuldades encontradas durante o desenvolvimento.

4.6 Componentes

4.6.1 Sistema de Autenticação e Gestão de Utilizadores

Este componente é responsável por toda a camada de autenticação e gestão de sessões de utilizador, implementado através do Supabase Auth (RF-002).

Implementação técnica:

- Autenticação por **magic link** enviado por email, eliminando a necessidade de gestão de passwords
- Integração com **Google OAuth** para login social simplificado
- Gestão automática de **sessões** através de tokens JWT armazenados em cookies HTTP-only
- Middleware de Next.js que verifica autenticação em todas as rotas protegidas
- Tabela auth.users gerida pelo Supabase com políticas RLS (Row Level Security) ativadas

Fluxo de autenticação:

1. Utilizador insere email ou clica em "Login with Google"
2. Supabase gera token de autenticação e redireciona para dashboard
3. Token é armazenado em cookie seguro e validado em cada request
4. Renovação automática de token antes da expiração

4.6.2 Sistema de Gestão de Créditos e Subscrições

Componente que controla o modelo de negócio da aplicação, gerindo créditos de utilizador e integrando com o Stripe para processamento de pagamentos (RF-003, RF-017, BN-001).

Implementação técnica:

- Tabela `user_balances` que armazena saldo de créditos por utilizador
- Tabela `subscriptions` que mapeia planos Stripe aos utilizadores
- **Webhooks do Stripe** que sincronizam estados de pagamento em tempo real
- Middleware que valida saldo suficiente antes de executar operações pagas (scraping com AI, otimização de imagens)
- Sistema de débito atómico de créditos usando transações PostgreSQL para evitar race conditions

Operações principais:

1. Criação de checkout session do Stripe para compra de créditos
2. Processamento de webhooks: `checkout.session.completed`, `invoice.paid`, `customer.subscription.deleted`
3. Débito automático de créditos ao utilizar funcionalidades premium
4. Consulta de saldo em tempo real no dashboard

4.6.3 Integração com Shopify

Componente responsável pela publicação de produtos na loja Shopify do utilizador (RF-016).

Implementação técnica:

- Utilização da **Shopify Admin REST API** (versão 2024-01) para criação de produtos
- Autenticação via `X-Shopify-Access-Token` armazenado de forma segura
- Endpoint `Next.js /api/shopify/publish` que recebe dados estruturados do produto

Estrutura de dados publicados:

- Informações base: título, handle (URL slug), descrição HTML, vendor, tipo de produto
- **Imagens:** array de URLs com texto alternativo
- **Opções:** até 3 níveis (ex: Cor, Tamanho, Material)
- **Variantes:** combinações de opções com preços, SKU, barcode, inventory, peso
- Status definido como `active` para publicação automática
- Scope `web` para disponibilidade em todos os canais de venda

Fluxo de publicação:

1. Request POST com dados estruturados do produto
2. Construção do payload conforme formato da Shopify API
3. Criação do produto via API

4. **Segunda chamada** para atualizar imagens específicas de cada variante (Shopify requer isto separadamente)
5. Se collectionIds fornecidos, adiciona produto às coleções através da API de Collects
6. Retorno de confirmação com ID do produto criado

Gestão de variantes:

- Suporte para produtos sem opções (single variant)
- Mapeamento correto de option1, option2, option3 baseado no número de opções
- Configuração de inventory management e taxable por variante
- Políticas de inventário configuráveis (deny/continue quando stock esgotado)

Tratamento de erros:

- Validação de resposta da Shopify API
- Logging detalhado de erros para debugging
- Falhas em operações secundárias (imagens de variantes, coleções) não bloqueiam a publicação principal

4.6.4 Web Scraping de Produtos Shopify

Componente responsável pela extração automática de informação de produtos de lojas Shopify (RF-008, RF-009).

Implementação técnica:

- Acesso direto ao endpoint público .json das lojas Shopify
- Parsing de dados estruturados retornados em formato JSON
- Normalização e formatação dos dados para apresentação ao utilizador
- Sem necessidade de web scraping tradicional (DOM parsing) ou Puppeteer

Método de extração:

- URL de produto: <https://loja.com/products/nome-produto>
- Endpoint JSON: <https://loja.com/products/nome-produto.json>
- Request GET simples retorna objeto JSON completo com toda a informação do produto

Dados extraídos:

- Título, descrição (body_html), vendor, tipo de produto
- Imagens: array com URLs em alta resolução, IDs e posições
- Opções: nome e valores possíveis (ex: Color: [Red, Blue, Green])

- Variantes: todas as combinações de opções com:
 - Preços (price, compare_at_price)
 - Disponibilidade (available, inventory_quantity)
 - SKU, barcode, peso (grams)
 - ID de imagem associada à variante
- Tags, metafields, created_at, updated_at

Parsing e apresentação:

- Conversão de HTML da descrição para formato editável
- Organização de variantes em estrutura hierárquica por opções
- Mapeamento de imagens às variantes correspondentes
- Cálculo de estatísticas (preço mínimo/máximo, stock total)
- Formatação de dados para interface amigável no dashboard

Performance:

Extração instantânea via JSON (< 1 segundo)

Sem necessidade de navegação headless ou rendering de JavaScript

Cache opcional dos dados extraídos para evitar requests repetidos

Cumprimento do requisito de performance RNF-001 (< 2.5 segundos total)

4.6.5 Otimização de Texto com Inteligência Artificial

Componente que melhora e traduz descrições de produtos usando modelos de linguagem (RF-014).

Implementação técnica:

- Integração com OpenAI API (GPT-4o-mini para otimização rápida)
- Integração com DeepL API para traduções profissionais
- Prompts especializados para diferentes tipos de otimização: reescrita persuasiva, SEO-friendly, tradução contextual
- Sistema de fallback: se OpenAI falhar, utiliza DeepL; se ambos falharem, mantém texto original
- Débito automático de créditos por cada operação de AI

Funcionalidades:

Tradução de descrições para qualquer idioma mantendo contexto comercial

Reescrita de texto genérico em descrição persuasiva e profissional

Otimização de títulos para SEO

Geração de bullet points a partir de descrições longas

4.6.6 Processamento de Imagens com Inteligência Artificial

Componente que melhora automaticamente a qualidade visual das imagens de produto (RF-015).

Implementação técnica:

- Integração com Potoroom API para remoção de fundos
- Integração com Fal.ai para correção de iluminação e upscaling
- Processamento em batch de múltiplas imagens em paralelo
- Upload temporário de imagens processadas para Supabase Storage
- Compressão automática para otimizar tamanho sem perder qualidade

Pipeline de processamento:

Download de imagens originais

Remoção de fundo (Potoroom)

Correção de iluminação e contraste (Fal.ai)

Redimensionamento para dimensões standard Shopify

Compressão WebP para web

Upload para storage temporário

URLs finais retornadas para publicação

4.6.7 Dashboard e Interface de Utilizador

Componente frontend que apresenta toda a experiência de utilizador (RF-001, RF-004, RF-012, RF-013, RNF-003).

Implementação técnica:

- **React Components** funcionais com hooks (useState, useEffect, useContext)
- **Tailwind CSS** para estilização responsiva
- **React Hook Form** para gestão de formulários de edição

- State management local para dados de produto em edição
- **Real-time updates** de saldo de créditos via Supabase Realtime

Componentes principais:

- ProductImportForm - Input de URL e trigger de scraping
- ProductEditor - Interface de edição completa de produto
- ImageGallery - Visualização e gestão de imagens
- VariantManager - Gestão de variantes (cores, tamanhos, preços)
- CreditBalance - Display de saldo e link para compra
- StoreSelector - Alternância entre lojas Shopify conectadas

Fluxo de utilizador:

Inserção de URL → Scraping automático (loading state)

Visualização de dados extraídos → Edição manual ou com AI

Preview final → Publicação com um clique

Feedback visual de sucesso/erro

4.7 Interfaces

Os protótipos de interface apresentados nesta secção correspondem ao estado atual da aplicação e servem para contextualizar os requisitos de usabilidade (RNF-003) e a experiência de utilizador. Nesta fase, tanto a Landing Page como o Dashboard encontram-se num estado de desenvolvimento avançado e funcional, representando fielmente o fluxo de trabalho pretendido. Embora a estrutura principal e as funcionalidades core estejam implementadas, o sistema continua em evolução, podendo ainda ser integradas funcionalidades secundárias e refinamentos visuais até à versão final. Os elementos de desenvolvimento presentes nos cantos das screenshots (como indicadores de debug) são ferramentas de suporte e não fazem parte da interface final entregue ao utilizador.

1. Landing Page (CTA e Inserção de Link Interativo) – Esta interface atua como o ponto de contacto inicial para o utilizador (RF-001). Apresenta *Call to Actions* (CTAs) essenciais para Login e Criação de Conta (RF-002). O elemento central é o campo interativo que permite ao visitante inserir o URL de um produto de fornecedor, demonstrando a funcionalidade imediata de *scraping* (RF-004).

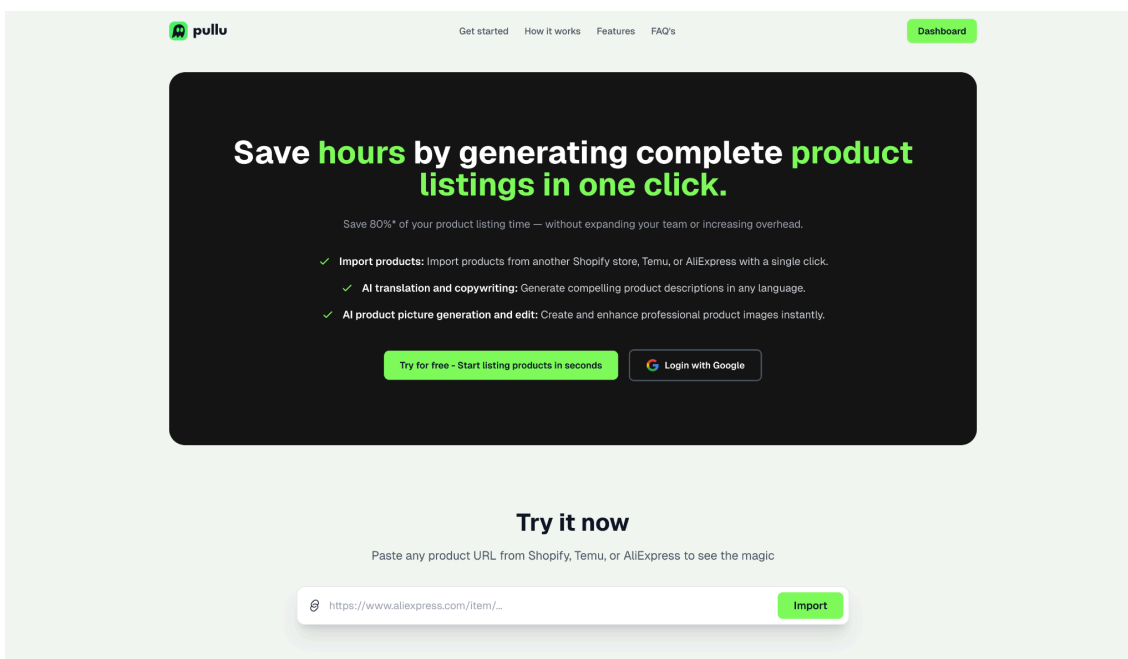


Figura 15

2. Fluxo de Valor (3-Step Illustration) – Esta imagem ilustra de forma clara e concisa o fluxo de trabalho simplificado da Pullu em apenas três passos: Colar o *link* do produto, Otimizar o conteúdo com AI, e Publicar o produto na Shopify. Serve para comunicar o cumprimento do requisito de eficiência (RNF-001).

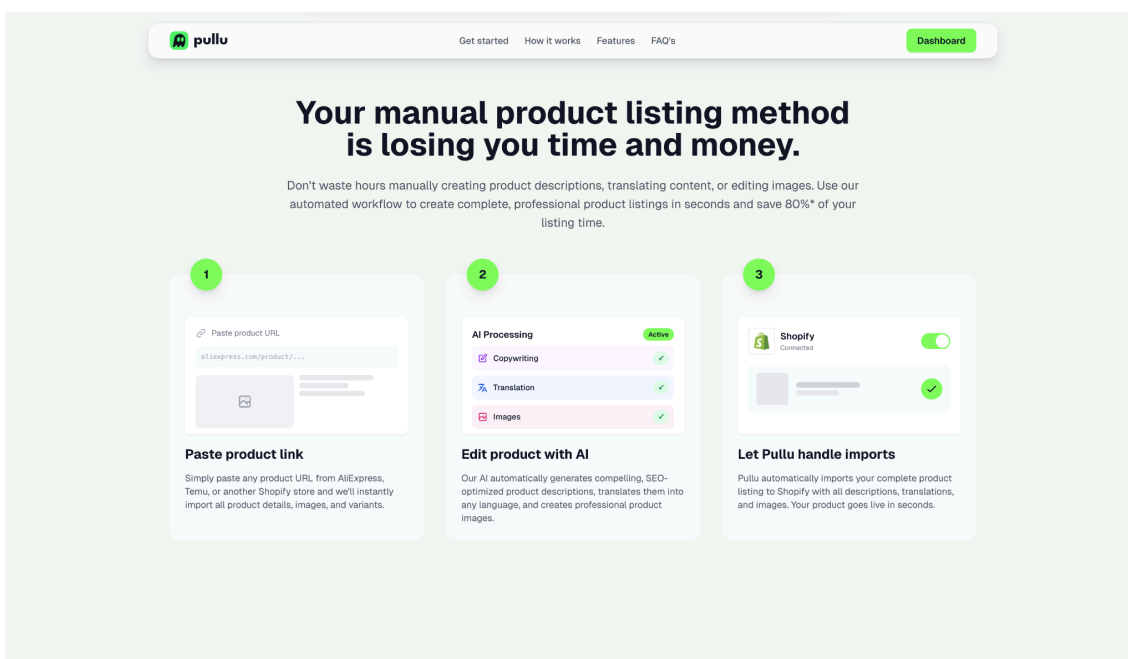


Figura 16

3. Pullu vs Competitors (Análise Comparativa) – Esta *screenshot* apresenta um quadro comparativo que destaca a proposta de valor e as funcionalidades de inovação da Pullu (e.g., otimização de imagem, geração de vídeo por AI - RF-015), em contraste com a concorrência direta no mercado de *dropshipping*.

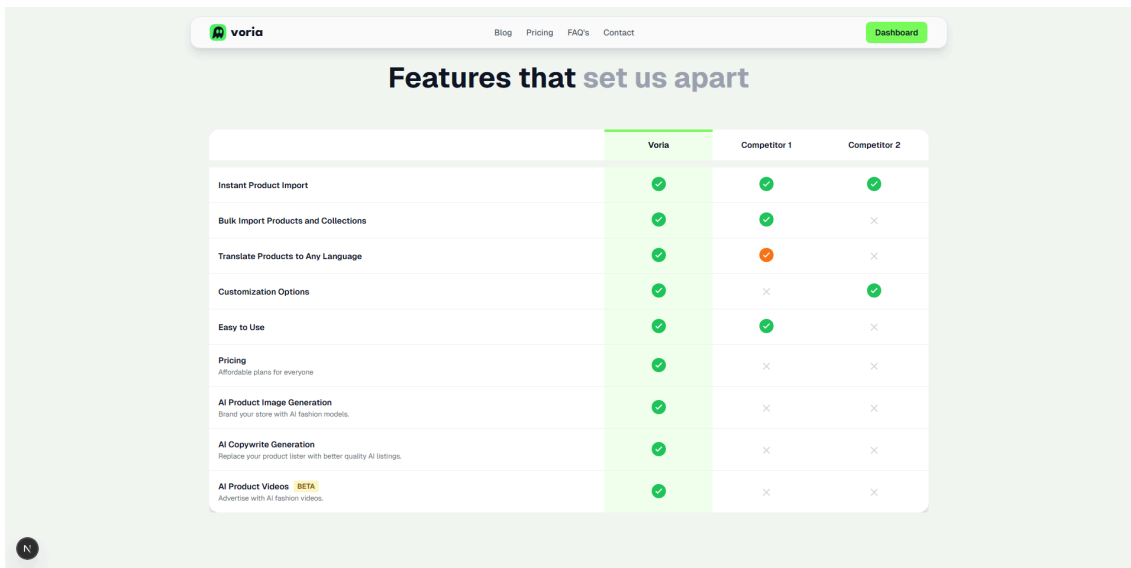


Figura 17

5. FAQ (Perguntas Frequentes) – Esta secção da Landing Page fornece respostas para as questões mais comuns, melhorando a experiência de utilizador e a documentação básica do produto (RF-001).

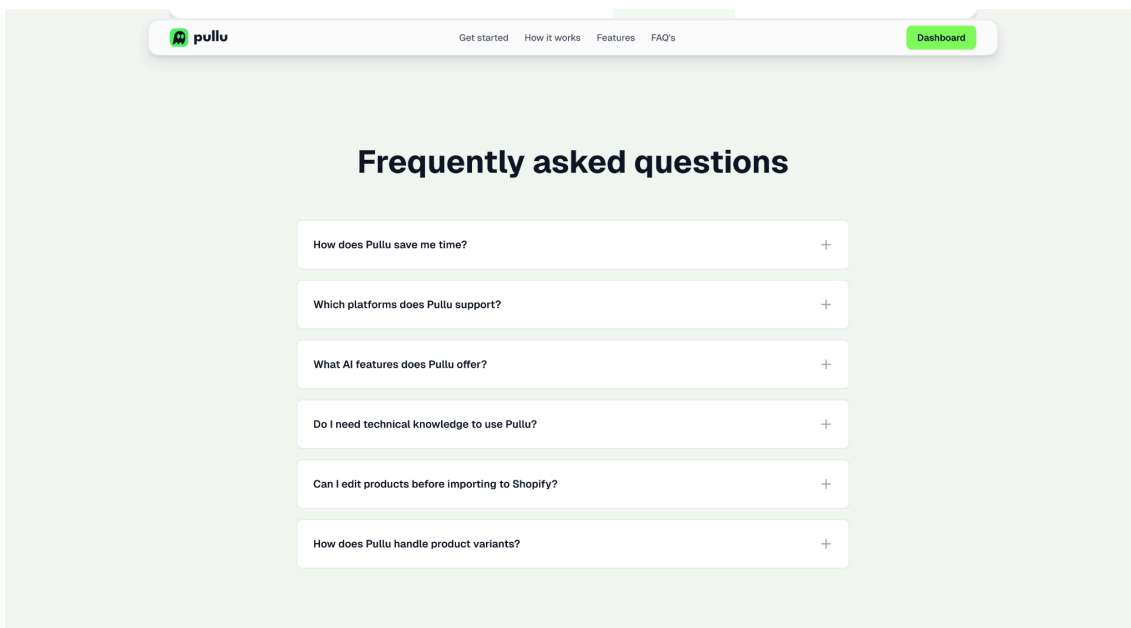


Figura 19

6. Footer (Web) – Representação do rodapé da página, que contém *links* de navegação essenciais e informações legais básicas, conforme o padrão de *websites* profissionais.

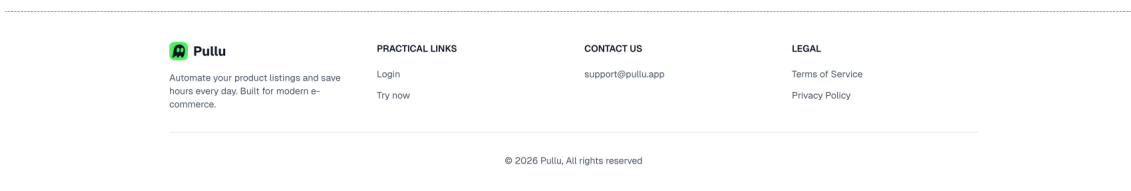


Figura 20

7. Termos e Condições e Política de Privacidade – Página com a fundamentação legal e regras de utilização da plataforma. A página de Política de Privacidade mantém a mesma estrutura visual e organizacional.

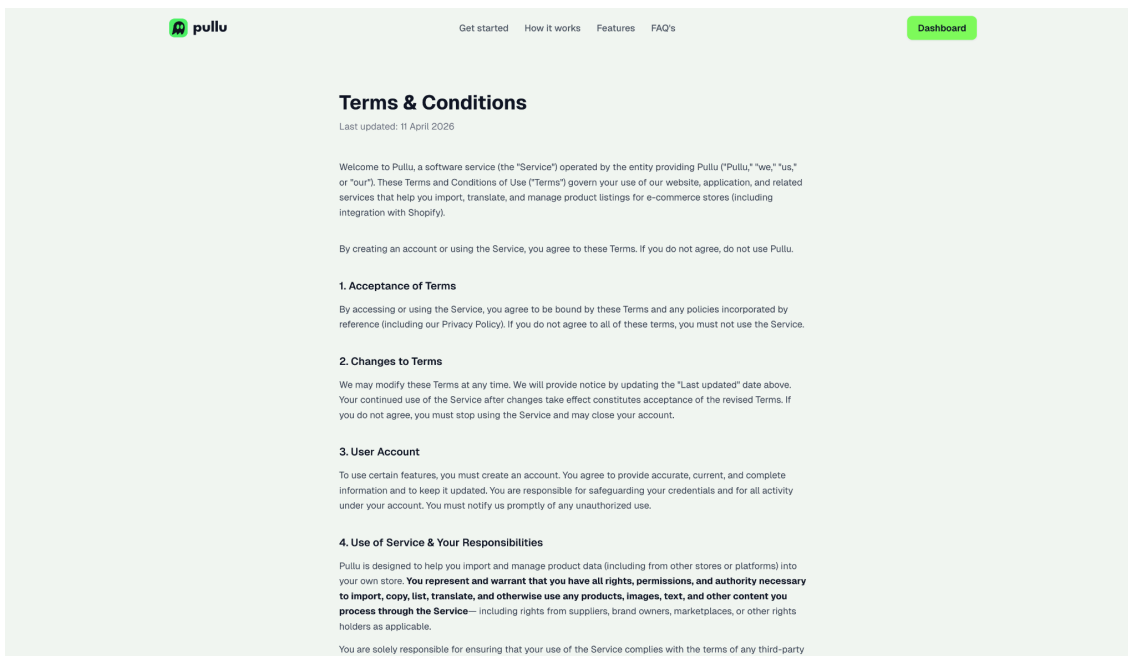


Figura 21

8. Página de Login – Interface simples e direta para autenticação de utilizadores registados na plataforma.

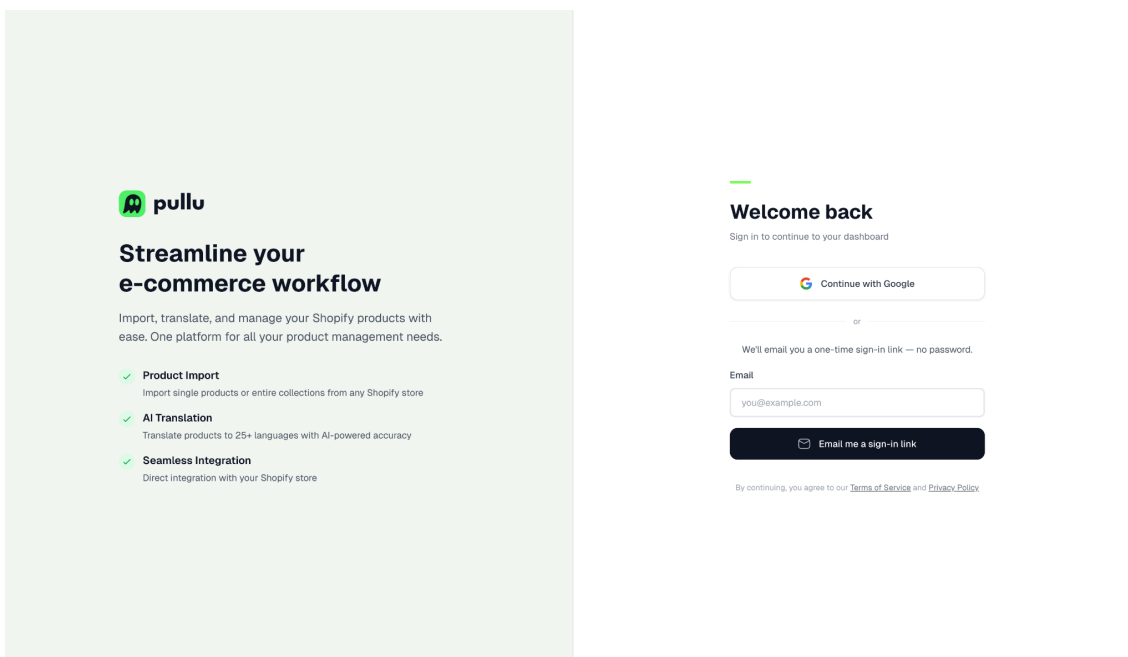


Figura 22

9. Guia de Conexão (/guide) – Manual passo-a-passo que explica ao utilizador como obter as credenciais da Shopify e conectar a sua loja.

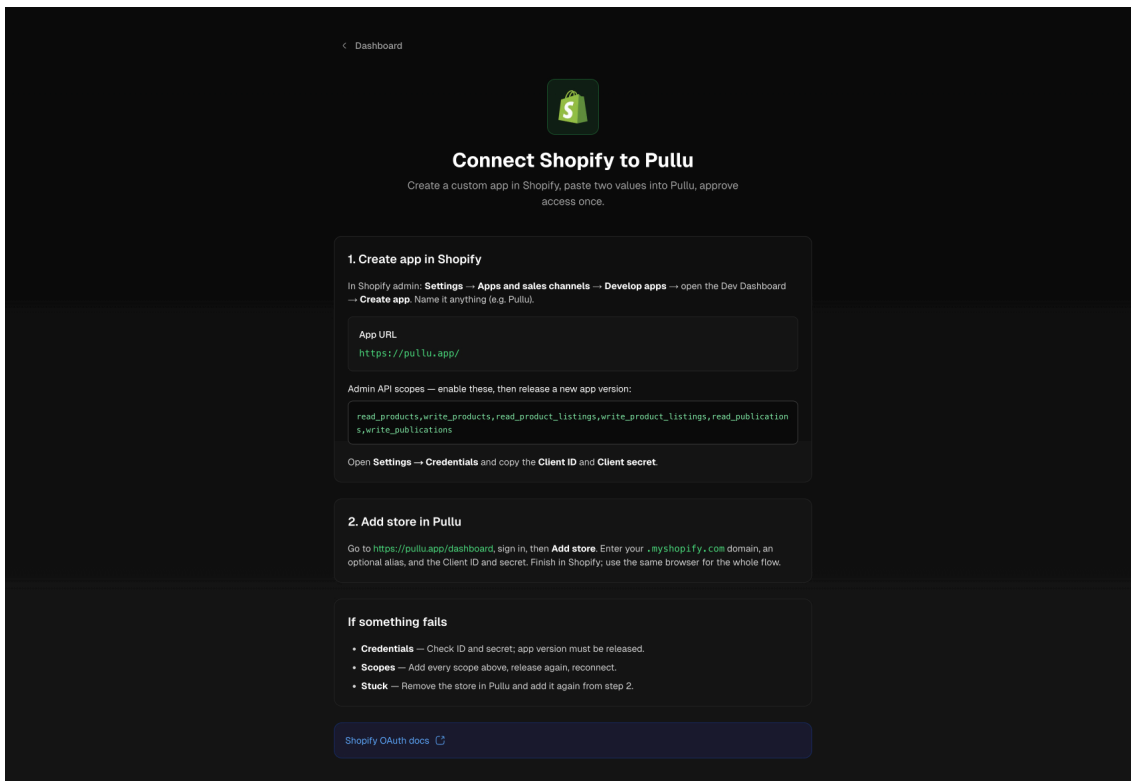


Figura 23

10. Dashboard (Estado Bloqueado) – Tela inicial que obriga o utilizador a conectar uma loja antes de permitir o acesso às ferramentas de importação.

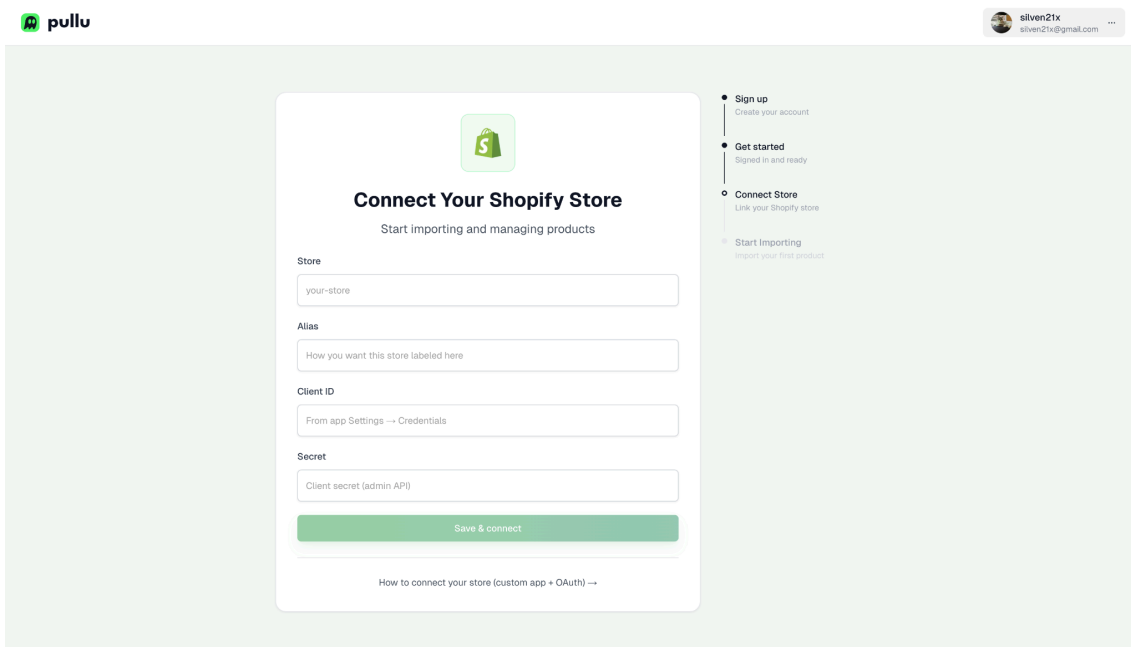


Figura 24

11. Dashboard Principal – Painel de controlo central onde o utilizador pode gerir as funcionalidades após a configuração da loja.

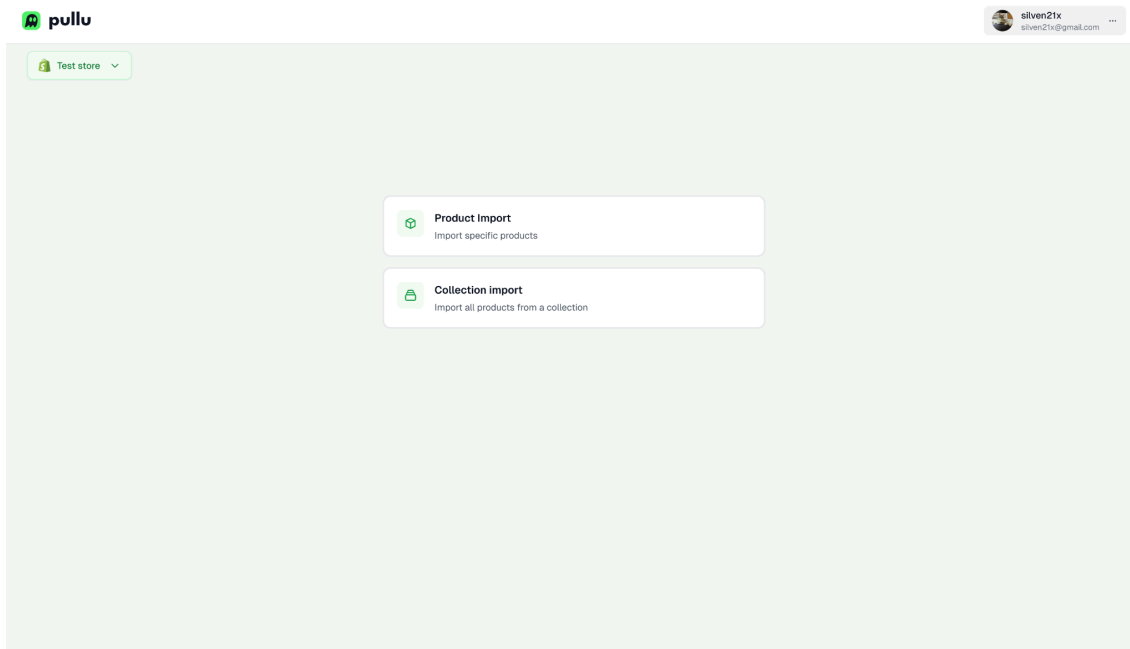


Figura 25

12. Definições de Conta – Tela que permite ao utilizador gerir o seu perfil, adicionar a sua API Key da Shopify ou apagar a conta.

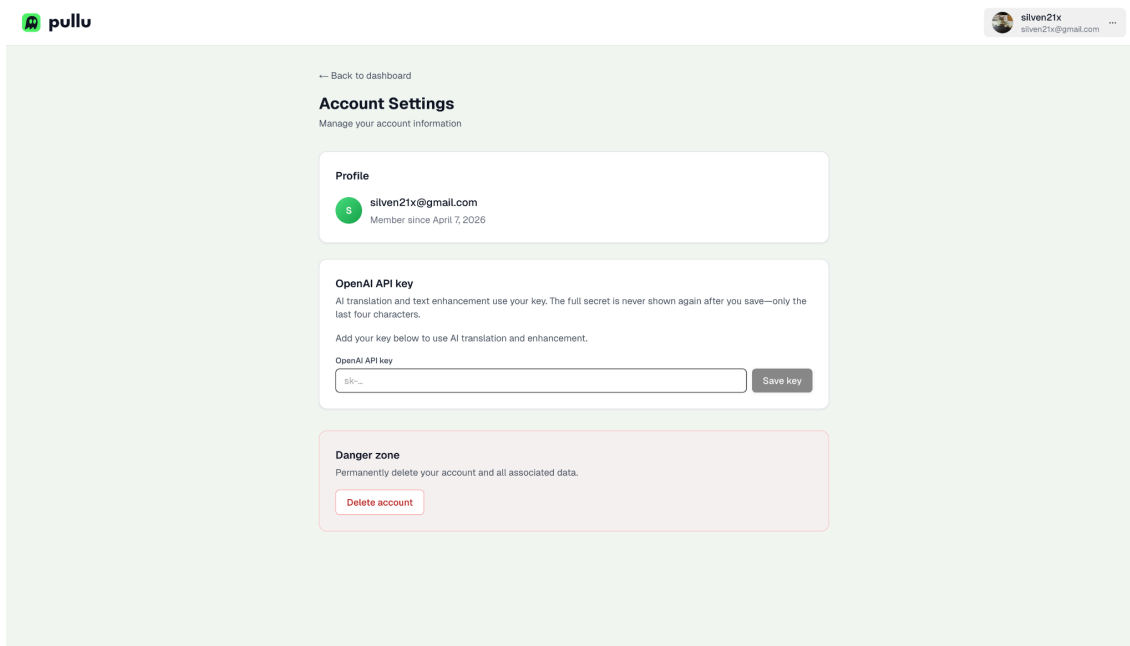


Figura 26

13. Inserção de Link – Interface para colocar o link do produto ou da coleção que se pretende importar; o funcionamento é idêntico para ambos os casos.

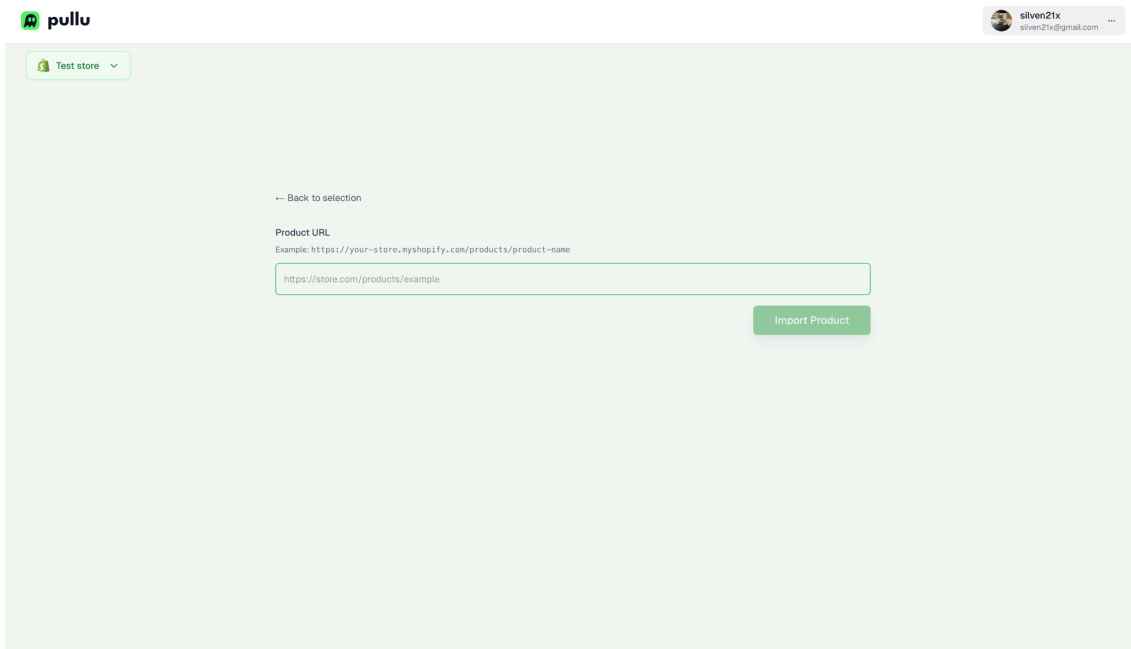


Figura 27

14. Importação de Produto (Configuração) – Primeira fase da importação onde são apresentados os dados extraídos para revisão do utilizador.

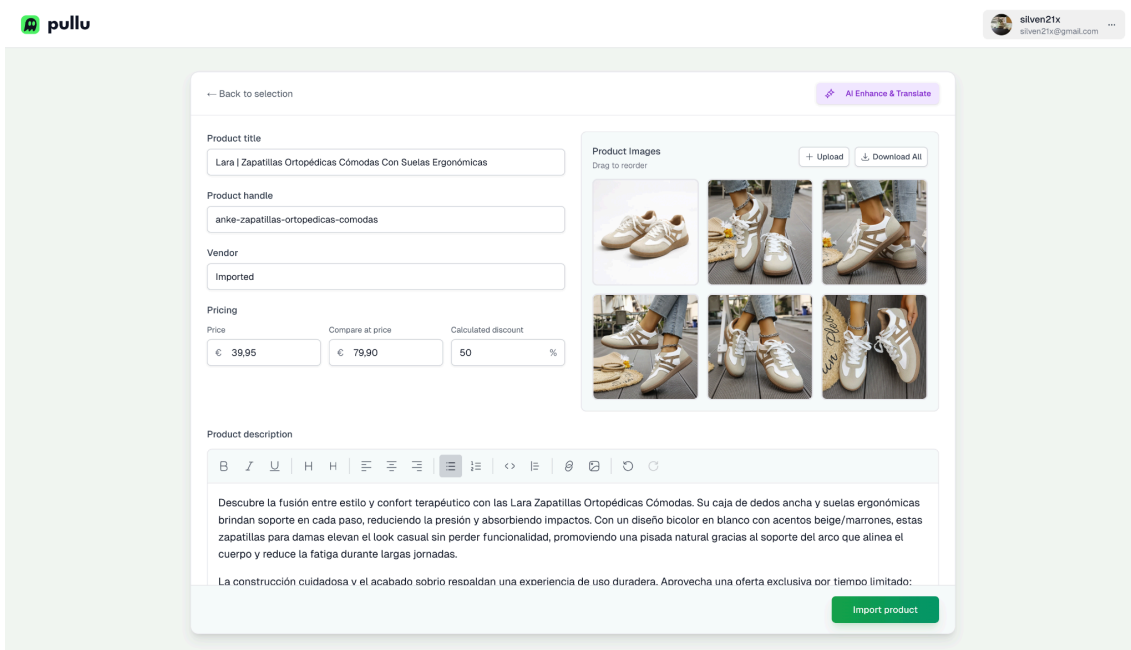


Figura 28

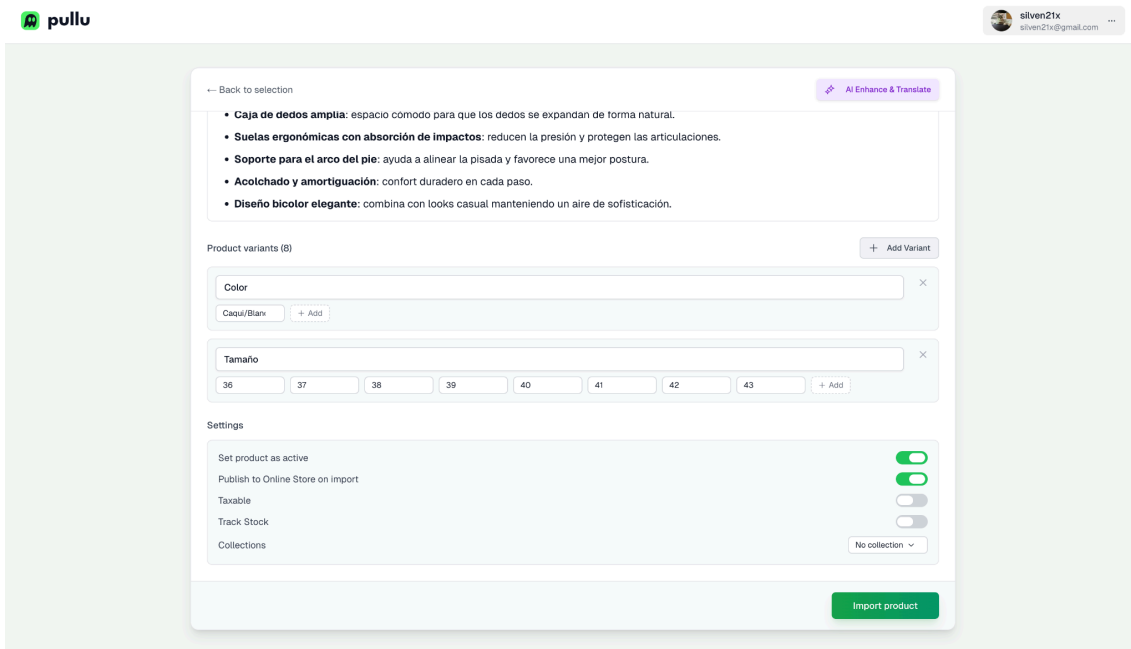


Figura 29

15. Importação de Coleção – Interface específica que demonstra o processo de importação de múltiplos produtos em massa através de um link de coleção.

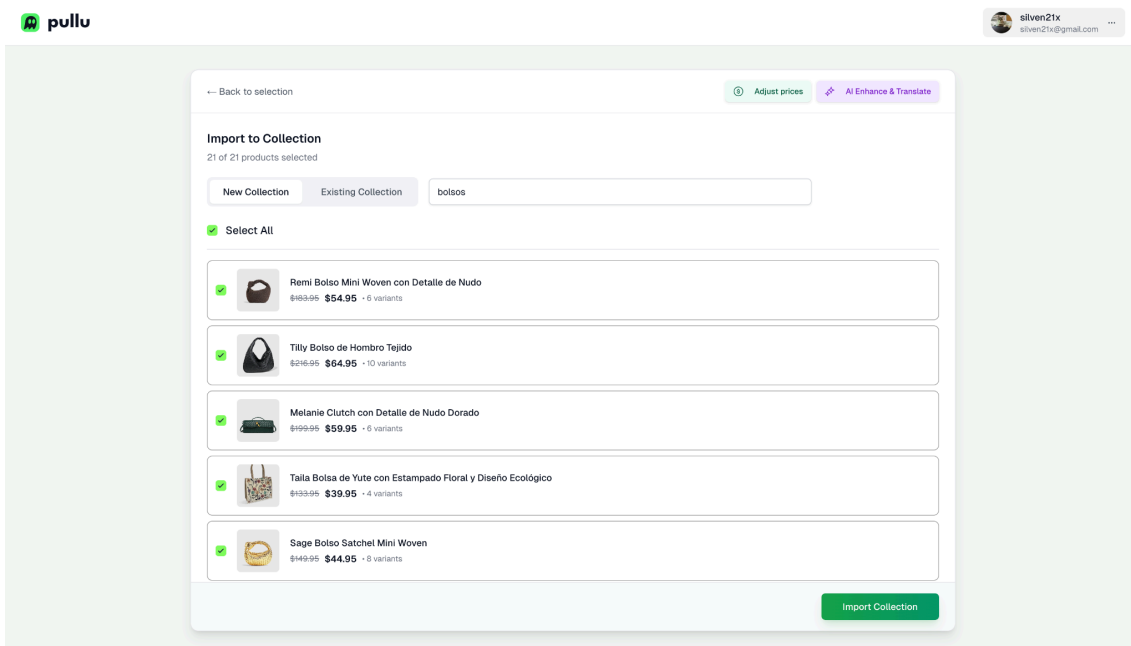


Figura 30

4.8 Riscos e Mitigação

No que diz respeito ao RGPD, a Pullu segue o princípio da minimização de dados, armazenando apenas a informação estritamente necessária para o funcionamento da plataforma — nomeadamente o email do utilizador, as credenciais de ligação à loja Shopify e a chave de API da OpenAI caso o utilizador a configure. Não são recolhidos dados desnecessários nem partilhados dados com terceiros para fins que não os estritamente operacionais. O utilizador pode, a qualquer momento, eliminar a sua conta através da página de definições da plataforma, o que resulta na remoção dos seus dados do sistema, em conformidade com o direito ao apagamento previsto no RGPD. Esta funcionalidade está implementada e acessível sem necessidade de contactar suporte.

Em termos de segurança, toda a base de dados está protegida por políticas de Row Level Security (RLS) no Supabase, garantindo que cada utilizador apenas consegue aceder e modificar os seus próprios dados, independentemente da forma como os pedidos são feitos. Toda a comunicação entre o cliente e o servidor é feita exclusivamente via HTTPS, e as credenciais sensíveis, como os tokens de acesso à Shopify e chaves de API, são armazenadas de forma encriptada. A autenticação é gerida pelo Supabase Auth com tokens JWT, renovados automaticamente, eliminando a necessidade de gestão manual de sessões e reduzindo a superfície de ataque. Nesta fase do projeto, estas medidas cobrem as principais vulnerabilidades de uma aplicação SaaS em fase MVP.

Relativamente à conformidade com os Termos de Serviço da Shopify, a Pullu utiliza exclusivamente as APIs e endpoints oficiais e públicos da plataforma, com a ligação à loja do utilizador estabelecida via OAuth com consentimento explícito e permissões mínimas necessárias. Não é contornado nenhum mecanismo de segurança nem acedido qualquer dado sem autorização. Nos Termos de Serviço da Pullu está claramente definido que o utilizador é o único responsável pelo uso que faz da ferramenta e pelo conteúdo que importa e publica. Tal como um fabricante de impressoras não é responsável se alguém as usar para contrafazer livros com direitos de autor de terceiros, a Pullu não é responsável por eventuais usos indevidos por parte do utilizador. Qualquer violação dos Termos de Serviço pode resultar no encerramento imediato da conta sem aviso prévio. No fundo, a conformidade com os ToS da Shopify é mais uma camada de boas práticas do que um risco crítico — a aplicação foi desenhada desde o início para operar dentro dos limites definidos pela plataforma.

Quanto a dependências externas, importa contextualizar o risco de forma realista. A API da Shopify utilizada pela Pullu é exatamente a mesma que alimenta o próprio painel de administração da Shopify. Isto significa que, se essa API estiver indisponível, o utilizador já não consegue sequer fazer upload de produtos manualmente através do backoffice da sua loja — o problema seria da Shopify e afetaria toda a plataforma, não sendo algo específico da Pullu. Para os restantes serviços, como Supabase e Vercel, ambos assentam em infraestrutura AWS, amplamente reconhecida pela sua estabilidade e uptime elevado. Em caso de indisponibilidade prolongada ou de necessidade de maior controlo sobre a infraestrutura, a solução de contingência passa pela migração dos serviços para um VPS próprio, o que é tecnicamente viável dado que a aplicação está construída em Next.js e utiliza PostgreSQL, tecnologias portáteis e sem dependência de vendor específico.

5 Testes e Validação

O desenvolvimento do projeto foi realizado de forma contínua com interação e testes com utilizadores, sendo a validação da solução integrada ao longo de todo o processo de construção da aplicação. Desde as primeiras versões, o sistema foi testado em contexto real por um pequeno conjunto de utilizadores (entre 1 a 3), com experiência na área do e-commerce, permitindo recolher feedback direto e iterar rapidamente sobre a solução.

Nos primeiros testes realizados, foi imediatamente identificado que a interface da aplicação apresentava fragilidades significativas ao nível da usabilidade e organização. Apesar de a ideia ter sido considerada “interessante” e com potencial, existiam vários problemas que dificultavam a utilização eficiente da plataforma.

Entre os principais pontos identificados nesta fase inicial, destacam-se:

- Existência de demasiado espaço vazio na interface, resultando numa perceção de falta de estrutura e organização da informação.
- Problemas na importação de coleções, onde as variantes dos produtos não ficavam corretamente associadas às respetivas imagens e, em alguns casos, os próprios dados das variantes eram corrompidos, levando a falhas na importação completa de coleções.
- O editor de texto rico (rich text editor) utilizado para edição das descrições não apresentava um comportamento totalmente estável, dificultando a formatação e edição eficiente do conteúdo.
- Ausência de uma funcionalidade simples que permitisse, ao importar um produto individual, ajustar diretamente o desconto através de uma percentagem, tornando o processo menos intuitivo e mais demorado.
- Falta de opção para descarregar imagens do produto (individualmente ou em lote, por exemplo em formato ZIP) durante o processo de importação, funcionalidade relevante para utilizadores que pretendem avançar diretamente para a criação de anúncios (media buying), permitindo poupança de tempo operacional.
- Inexistência de uma funcionalidade para edição de preços em massa (bulk) durante a importação de coleções, limitando a eficiência na gestão de múltiplos produtos.
- Ausência de uma opção rápida para visualizar o produto diretamente no painel de administração da Shopify ou na loja online após a importação, obrigando o utilizador a navegar manualmente até essas páginas.

Com base neste feedback inicial, a aplicação foi significativamente reformulada. A interface (dashboard) foi redesenhada praticamente na sua totalidade, incluindo um novo layout, reorganização dos elementos, melhoria da hierarquia visual e introdução de um novo esquema de cores, com o objetivo de tornar a experiência mais clara, moderna e intuitiva.

Paralelamente à reformulação da interface, os problemas identificados foram corrigidos e várias novas funcionalidades foram adicionadas, alinhadas com as necessidades reais dos utilizadores. Estas melhorias permitiram não só resolver os erros iniciais, mas também otimizar o fluxo de trabalho e reduzir a fricção no processo de importação e edição de produtos.

Para além das alterações diretamente resultantes do feedback dos utilizadores, foram também introduzidas melhorias adicionais com base na análise contínua do desenvolvimento, contribuindo para uma solução mais robusta, consistente e orientada ao uso prático.

Ao longo do desenvolvimento, foram igualmente realizados testes funcionais contínuos, com o objetivo de garantir o correto funcionamento das principais funcionalidades da aplicação. Estes testes incluíram a importação de produtos a partir de URLs, a edição de descrições com recurso a inteligência artificial e a publicação direta na loja Shopify, permitindo validar o fluxo completo da aplicação.

Adicionalmente, a aplicação foi testada em ambiente real, recorrendo às integrações com serviços externos essenciais ao seu funcionamento, nomeadamente a API da Shopify e serviços de inteligência artificial para processamento de texto e imagem. Esta validação operacional permitiu confirmar que a solução funciona de forma integrada em condições reais, cumprindo o seu objetivo principal de automatizar e simplificar o processo de criação de listagens de produtos no contexto do e-commerce.

6 Método e Planeamento

6.1 Planeamento inicial

O planeamento inicial do desenvolvimento da Pullu foi estruturado em quatro fases de dois meses cada, garantindo uma evolução progressiva das funcionalidades e permitindo ajustes ao longo do processo. A organização descrita nesta secção segue o calendário proposto e reflete tanto o trabalho realizado como o planeamento das fases seguintes. Cada fase corresponde a um conjunto de tarefas associadas aos requisitos funcionais, componentes definidos no capítulo 4.6 e às user stories descritas no capítulo 3.

Fase 1 – Desenvolvimento Inicial e Estrutura Fundamental (Semanas 1–8)

A primeira fase focou-se na criação da base técnica da aplicação e no desenvolvimento inicial das funcionalidades principais. Nesta etapa foram implementados vários elementos importantes, ainda que muitos deles tenham permanecido incompletos ou em versão inicial. Foi desenvolvida uma parte significativa do sistema de autenticação, que se encontra já bastante avançado, incluindo email e Google OAuth. A Landing Page inicial também foi construída e está praticamente concluída na sua primeira versão. O sistema de importação de produtos e coleções de lojas Shopify ficou funcional, permitindo obter dados como variantes, imagens, preços e descrições. A interface de edição do produto, incluindo alteração de textos, imagens, preços e organização de variantes, foi também desenvolvida e encontra-se funcional. Foi ainda implementado o processo de ligação da loja Shopify do utilizador através de OAuth. No entanto, a API responsável por adicionar diretamente o produto editado à loja Shopify ainda necessita de otimização, especialmente ao nível da velocidade de resposta. Durante esta fase foi também criada toda a base de dados no Supabase, com as tabelas essenciais para utilizadores, lojas, produtos importados, créditos e subscrições.

Fase 2 – Consolidação, Otimização e Implementação de AI e Pagamentos (Semanas 9–16)

A segunda fase está orientada para a conclusão e consolidação das funcionalidades já iniciadas, bem como para a introdução de novos módulos fundamentais. Nesta etapa pretende-se finalizar a Landing Page, incluindo a secção de preços (pricing) e melhorias visuais. Será concluída a implementação da importação de produtos Shopify, garantindo tempos de resposta mais rápidos e maior estabilidade. A edição com inteligência artificial terá um papel central nesta fase, integrando ferramentas tanto para otimização e reescrita de texto como para edição de imagens, permitindo ao utilizador gerar descrições completas e melhorar conteúdos visuais do produto. Será também incorporado o sistema de pagamentos, incluindo Stripe, gestão de subscrições e compra de créditos. A interface do dashboard será refinada, tornando-se completa, funcional e visualmente consistente com a experiência final pretendida.

Fase 3 – Expansão de Fontes, Novas Funcionalidades de AI e Preparação da Versão Pública (Semanas 17–24)

A terceira fase foca-se na expansão da plataforma para além do ecossistema Shopify. Está

previsto adicionar a importação de produtos provenientes do Temu e AliExpress, permitindo ao utilizador obter produtos de diferentes marketplaces. Serão ainda adicionadas funcionalidades avançadas de inteligência artificial utilizando Fal.ai, incluindo geração de imagens mais avançada do que a disponível na fase anterior e geração automática de vídeos de produto. Durante esta fase será efetuado o primeiro deploy estável da aplicação, colocando-a num ambiente de produção. A aplicação começará a ser testada por utilizadores reais, recolhendo opiniões, feedback e identificando pontos críticos a melhorar.

Fase 4 – Testes com Utilizadores, Estabilização e Correções (Semanas 25–32)

A última fase do projeto será dedicada sobretudo à validação final da solução. O principal objetivo é testar a aplicação com utilizadores reais, identificar bugs, recolher feedback e implementar todas as alterações necessárias para garantir estabilidade, qualidade e alinhamento com os requisitos definidos. Esta análise permitirá melhorar a usabilidade, corrigir problemas encontrados durante a fase de testes e ajustar funcionalidades antes da disponibilização pública final da plataforma.

6.2 Análise Crítica ao Planeamento

O desenvolvimento tem seguido o plano geral, com pequenas adaptações. As únicas dificuldades relevantes até ao momento estiveram relacionadas com o desempenho da importação de coleções, uma vez que cada operação precisa de ser rápida e existe limitação de pedidos devido aos rate limits da Shopify. Adicionalmente, o desenvolvimento da interface do dashboard revelou-se mais demorado do que o previsto, exigindo várias iterações de design e organização.

Apesar destes pontos, o progresso mantém-se estável e dentro dos objetivos principais. Existe ainda a intenção de adiantar algumas tarefas das fases seguintes para permitir mais tempo de testes com utilizadores e potenciais melhorias antes da conclusão do TFC, embora sem garantia formal de antecipação.

Formulário de declaração de uso de ferramentas de Inteligência Artificial

<https://drive.google.com/file/d/1rZxX6IbTmT12Xitj5xkQ1XD3af4Td8P7/>

Bibliografia

- [DEISI24] DEISI, Regulamento de Trabalho Final de Curso, Out. 2024.
- [DEISI24b] DEISI, www.deisi.ulusofona.pt, Out. 2024.
- [Shop25] Shopify, Shopify Admin REST API Documentation, <https://shopify.dev/docs/api/admin-rest>, acedido em Nov. 2025.
- [Shop26] Shopify Developer Docs for Apps, <https://shopify.dev/docs/apps/build>, acedido em Abr. 2026.
- [Next25] Vercel, Next.js Documentation - The React Framework for the Web, <https://nextjs.org/docs>, acedido em Nov. 2025.
- [Supa25] Supabase, Supabase Documentation - Open Source Firebase Alternative, <https://supabase.com/docs>, acedido em Nov. 2025.
- [Stripe25] Stripe, Stripe API Documentation - Payment Processing Platform, <https://stripe.com/docs/api>, acedido em Nov. 2025.
- [OpenAI25] OpenAI, OpenAI API Documentation - GPT Models, <https://platform.openai.com/docs>, acedido em Nov. 2025.
- [DeepL25] DeepL, DeepL API Documentation - Translation Service, <https://www.deepl.com/docs-api>, acedido em Nov. 2025.
- [Photo25] PhotoRoom, PhotoRoom API - Background Removal Service, <https://www.photoroom.com/api>, acedido em Nov. 2025.
- [Fal25] Fal.ai, Fal.ai Documentation - AI Image Processing, <https://fal.ai/docs>, acedido em Nov. 2025.
- [Tail25] Tailwind Labs, Tailwind CSS Documentation, <https://tailwindcss.com/docs>, acedido em Nov. 2025.
- [Type25] Microsoft, TypeScript Documentation, <https://www.typescriptlang.org/docs>, acedido em Nov. 2025.
- [Stat25] Statista, E-commerce worldwide - Statistics & Facts, <https://www.statista.com/topics/871/online-shopping/>, acedido em Nov. 2025.

[Kopy25] Kopy, Kopy - Shopify Product Importer, <https://apps.shopify.com/kopy>, acedido em Nov. 2025.

[Poky25] Poky, Poky - Product Import Tool, <https://www.pokyapp.com>, acedido em Nov. 2025.

[ULHT25] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt, acedido em Nov. 2025.

Glossário

AI	Artificial Intelligence / Inteligência Artificial.
API	Application Programming Interface / Interface de Programação de Aplicações.
CDN	Content Delivery Network / Rede de Distribuição de Conteúdo.
CSS	Cascading Style Sheets.
HTML	HyperText Markup Language / Linguagem de Marcação de Hipertexto.
JS	JavaScript.
JSON	JavaScript Object Notation.
JWT	JSON Web Token.
LEI	Licenciatura em Engenharia Informática.
LIG	Licenciatura em Informática de Gestão.
MVP	Minimum Viable Product / Produto Mínimo Viável.
OAuth	Open Authorization (protocolo de autorização).
ODS	Objetivos de Desenvolvimento Sustentável.
REST	Representational State Transfer.
RLS	Row Level Security.
SaaS	Software as a Service.
SEO	Search Engine Optimization.
SSR	Server-Side Rendering.
TFC	Trabalho Final de Curso.
UI	User Interface / Interface de Utilizador.
ULHT	Universidade Lusófona de Humanidades e Tecnologias.
URL	Uniform Resource Locator.